



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Facultat d'Informàtica de Barcelona



Grado en Ingeniería Informática

Especialidad de Computación
Trabajo Final de Grado

Portal de mantenimiento para clientes

Autor: **Victor Vidal Rojas Condori**

Director: **Luis Ferré**

Ponente: **Javier Vázquez Salceda**

Octubre 2021
Barcelona

Resumen

La situación pandémica actual ha provocado que las industrias del mundo se den cuenta de la necesidad de informatizar todo su entorno de producción. aggity es una empresa que se especializa en ofrecer este tipo de servicios. Sin embargo, la gran cantidad de productos similares dificulta la captación de clientes.

El portal de mantenimiento para clientes de aggity es una herramienta encargada de unificar el mantenimiento de los servicios aggity, así como la creación de incidencias que puedan surgir, de todo el paquete de aplicaciones de aggity, centralizando la información y facilitando la interacción de los clientes con el entorno.

Resum

La situació pandèmica actual ha provocat que les indústries del món s'adonin de la necessitat d'informatitzar tot el seu entorn de producció. aggity és una empresa que s'especialitza en oferir aquest tipus de serveis. No obstant això, la gran quantitat de productes similars dificulta la captació de clients.

El portal de manteniment per a clients de aggity és una eina encarregada d'unificar el manteniment dels serveis aggity, així com la creació d'incidències que puguin sorgir, de tot el paquet d'aplicacions de aggity, centralitzant la informació i facilitant la interacció dels clients amb l'entorn.

Abstract

The current pandemic situation has caused the world's industries to realize the need to computerise their entire production environment. aggity is a company that specializes in offering these types of services. However, the large number of similar products makes it difficult to attract customers.

The aggity maintenance portal for clients is a tool in charge of unifying the maintenance of aggity services, as well as the creation of incidents that may arise, of the entire aggity application package, centralizing information and facilitating customer interaction with the environment.

Índice de Contenido

Introducción y Contextualización	7
1.1 Introducción	7
1.2 Contexto	7
1.3 Conceptos	8
1.3.1 Industria 4.0	8
1.3.2 Manufacturing Execution System (MES)	8
1.3.3 Clientes	8
1.3.4 Productos	9
1.3.5 Gestor de Incidencias	9
1.4 Actores implicados	9
Justificación	10
Alcance del Proyecto	11
3.1 Objetivo Principal	11
3.1.1 Conexión con servicios aggity	12
3.1.2 Gestor de Incidencias	12
3.2 Objetivos Secundarios	12
3.2.1 Protección de Datos	12
3.2.2 Gestión eficiente de los datos	12
3.2.3 Gestión de usuario	12
3.3 Requerimientos no funcionales	12
3.3.1 Diseño dinámico	13
3.3.1 Usabilidad de la web	13
3.3.1 Compatibilidad con navegadores	13
3.3.1 Diseño dinámico	13
3.3.1 Accesibilidad (lenguaje)	13
3.4 Obstáculos y Riesgos	13
3.4.1 Compatibilidad	13
3.4.2 Errores de programación	13
3.4.3 Carga de trabajo	14
3.4.4 Situación actual	14
Metodología y rigor	14
4.1 Metodología de trabajo	14
4.1.1 Funcionamiento de la metodología Scrum	14
4.1.2 Motivos de la elección y funcionamiento del equipo	15
Planificación Temporal	17

5.1	Introducción	17
5.2	Recursos	17
5.2.1	Humanos	17
5.2.2	Software	17
5.2.3	Hardware	18
5.3	Descripción de tareas	18
5.3.1	Gestión del Proyecto - GP	19
5.3.2	Trabajo Previo - TP	19
5.3.3	Diseño - DI	20
5.3.4	Primera Fase Desarrollo - FD	20
5.3.5	Prototipos - PT	20
5.3.6	Gestión de Usuario - GU	21
5.3.7	Gestión de Sistema de Incidencias - GSI	21
5.3.8	Gestión de los servicios de aggity - GSA	22
5.3.9	Fase Final Desarrollo - FF	22
5.4	Dependencias de Tareas	22
5.5	Gestión de Riesgos y obstáculos	24
5.6	Cambios respecto a la planificación inicial	25
	Presupuesto	27
6.1	Identificación y estimación de los Costes	27
6.1.1	Costes de los recursos	27
6.1.2	Costes por actividad	29
6.1.3	Coste de contingencias e imprevistos	30
6.1.4	Presupuesto final	31
6.2	Control de gestión	31
6.3	Cambios en el presupuesto respecto al presupuesto inicial	32
	Informe de Sostenibilidad	35
7.1	Dimensión ambiental	35
7.2	Dimensión Económica	36
7.3	Dimensión Social	37
	Arquitectura del sistema	37
8.1	Visión global del proyecto	37
8.2	Diseño de Bases de datos	39
8.3	Diseño interfaz aplicación web	40
8.3.1	Página principal	40
8.3.2	Página registro	41
8.3.3	Página Login	42
8.3.4	Páginas informativas	42
8.3.5	Página Estado Servicios (Solo conectado)	43

8.3.6	Página Incidencias (Solo conectado)	43
8.3.7	Página configuración usuario (solo conectado)	44
8.3.8	Páginas contraseña	44
8.3.9	Página empresa	44
8.3.10	Página teléfono	44
	Implementación	44
9.1	Tecnologías y lenguajes utilizados	44
9.2	Implementación de la aplicación web	45
9.2.1	Conexión a base de datos	45
9.3	Implementación servicio	47
	Cambios a futuro	49
	Conclusión	50
	Bibliografía	51

Índice de Figuras

Figura 1. Diagrama de actores y servicios implicados en el proyecto	10
Figura 2. Esquema de la Metodología Scrum	14
Figura 3. Diagrama de Gantt	24
Figura 4. Diagrama de Gantt final	34
Figura 5. Diagrama de relación entre las soluciones del proyecto	38
Figura 6. Diagrama de entidad-relación de la base de datos utilizada	40
Figura 7. Vista de la página principal sin inicio de sesión	41
Figura 8. Vista de la página principal con inicio de sesión	41
Figura 9. Vista de la página de registro	42
Figura 10. Vista de la página de inicio de sesión	42
Figura 11. Vista de la página de estado de los servicios	43
Figura 12. Vista de la página de incidencias	43
Figura 13. Vista de la página de administración de cuenta	44
Figura 14. Vista de la página de edición de contraseña	44
Figura 15. Ejemplo de uso de la sentencia using	46
Figura 16. Ejemplo de cadena de conexión del archivo de configuración	46
Figura 17. Lista de funciones usadas en la aplicación web	47
Figura 18. Variables en archivo de configuración del servicio	48
Figura 19. Ejemplo de uso de la clase timer	48
Figura 20. Servicio en línea en Windows	49

Índice de Tablas

Tabla 1. Resumen de las tareas del proyecto y sus dependencias	23
Tabla 2. Resumen de las tareas del proyecto en etapa final y sus dependencias	26
Tabla 3. Sueldo bruto por hora en función del rol	27
Tabla 4. Desglose del coste de los recursos físicos	28
Tabla 5. Desglose del coste de los recursos digitales	29
Tabla 6. Desglose del coste de los servicios ajenos al proyecto	29
Tabla 7. Horas dedicadas a cada tarea, por rol, y el coste que implica	30
Tabla 8. Horas dedicadas a cualquier error de programación no contemplado	31
Tabla 9. Estimación total del coste del proyecto	31
Tabla 10. Desglose final del coste de los recursos digitales	32
Tabla 11. Horas finales dedicadas a cada tarea, por rol, y el coste que implica	33
Tabla 12. Estimación total final del coste del proyecto	34

1. Introducción y Contextualización

1.1 Introducción

El sector industrial representa toda actividad manufacturera que consiste en producir bienes elaborados o semielaborados a partir de materias primas. Con los avances tecnológicos, este sector se ha visto obligado a evolucionar para poder satisfacer las necesidades de la población.

Estas evoluciones abarcan diversos cambios en el funcionamiento de las fábricas, desde la automatización de procesos hasta el tratamiento de grandes cantidades de datos para mejorar el rendimiento mediante el uso de tecnologías de análisis como la Inteligencia Artificial (IA) o reestructuración del flujo de trabajo gracias a un planificador.

Existen varias empresas que ofrecen una gran variedad de herramientas para facilitar esta transición a las nuevas tecnologías. *aggity* es una empresa multinacional que dispone de diversas tecnologías especializadas en la Transformación digital con soluciones prácticas y digitales.

La pandemia que estamos viviendo en la actualidad ha provocado que empresas, que en el pasado eran reticentes al cambio, hayan visto la necesidad de dedicar recursos a estos procesos de digitalización para seguir siendo competitivas en el mercado [1].

1.2 Contexto

Este documento tiene la función de definir de manera detallada todos los componentes relacionados con este Trabajo de Fin de Grado (TFG) que consiste en la creación de un *Portal de mantenimiento para clientes* para la empresa *aggity*. *aggity* dispone de diversas soluciones tecnológicas que ayudan, a diversos negocios, a transicionar a un sistema digital. En este proyecto nos centraremos en las soluciones enfocadas al sector industrial.

La situación actual de la pandemia ha provocado que diversas empresas se planteen la digitalización de su estructura de trabajo, sin embargo, la gran cantidad de soluciones disponibles en el mercado puede suponer un problema a la hora de destacar los servicios que ofrece *aggity*.

Empresas como SAP o Mapex, disponen de más experiencia en el mercado, lo que les coloca en una mejor posición de cara a un cliente que tiene que decidir la opción más económica que encaje con su modelo de negocio.

A todo esto hay que añadirle que gran parte de la comunicación, entre aggity y los clientes, se realiza a través de mensajes de correo sin usar ningún tipo de herramientas especializada. Esto provoca que no haya un control eficiente de cualquier comentario o incidencia, lo que ralentiza cualquier solución o desarrollo y, por lo tanto, empeora la experiencia del cliente.

Con este proyecto intentamos crear un producto que conecte las diversas herramientas que posee aggity para facilitar la recepción y estudio de los datos por parte del cliente, consiguiendo aumentar el interés de posibles clientes en el mercado.

Al tener todos los datos centralizados en un mismo servicio podremos, a su vez, agilizar y automatizar la comunicación ante cualquier incidencia por parte nuestra o del cliente al incluir un sistema de gestión de incidencias.

1.3 Conceptos

En este apartado definiremos conceptos importantes que serán recurrentes a lo largo del proyecto y serán necesarios para entender el proyecto en profundidad.

1.3.1 Industria 4.0

Concepto utilizado para identificar la cuarta revolución industrial y que, al ser una etapa bastante reciente, no es posible definirla completamente. El objetivo principal consistiría en aumentar la eficiencia y adaptabilidad de las fábricas utilizando medios tecnológicos actuales, lo que se conoce como “Smart Factories”.

1.3.2 Manufacturing Execution System (MES)

Los MES [2] son sistemas diseñados para la monitorización y control de la fábrica. En la actualidad, también disponen de herramientas para conectarse con máquinas y así automatizar su funcionamiento. Opera MES by aggity es un ejemplo de este tipo de sistemas.

1.3.3 Clientes

Un cliente es un individuo o empresa que solicita servicios a otro profesional o empresa de manera regular. En este proyecto, tendremos que diferenciar entre los clientes de aggity y un cliente final.

Llamaremos cliente a toda empresa que solicite cualquiera de los servicios que ofrece aggity. Estos clientes son, en su mayoría, empresas especializadas en la producción y distribución de productos.

En cambio, los clientes finales son empresas a las que se le distribuyen los productos producidos por los clientes.

1.3.4 Productos

Son las mercancías que elaboran o distribuyen los clientes. El rango de tipología de productos es tan amplio que abarca desde la distribución de papel hasta maquinaria industrial. Los clientes finales son los receptores de dicho producto.

1.3.5 Gestor de Incidencias

Con este concepto hacemos referencia al software que se encarga de automatizar y realizar un seguimiento de las incidencias que puedan surgir en un servicio. En nuestro caso, incorporaremos un gestor de incidencias externo en nuestro servicio para facilitar y automatizar el control de errores, del software de aggity, por parte de nuestros clientes.

1.4 Actores implicados

Los actores son las partes que están interesadas en el desarrollo de esta aplicación. Esto abarca desde la empresa, que se encarga de desarrollar el servicio, hasta los clientes interesados en comprar el producto para su uso o venta. A continuación se listan dichos actores.

aggity: Es la empresa encargada de distribuir este servicio cuando esté finalizado y, por lo tanto, interesada en el desarrollo del mismo. Es el actor más beneficiado económicamente ya que se encargará de la venta del proyecto a

Departamento de Industria: Encargados de desarrollar el producto y ofrecer el mantenimiento necesario para mantener el servicio activo. En este proyecto, yo me encargaré del desarrollo del servicio y, en caso de salir al mercado, el departamento de industria será el encargado de realizar el mantenimiento.

Clientes: Son los actores principales a los que va dirigido el producto. La mayoría de funcionalidades del servicio, como el gestor de incidencias o la visualización del estado del software de aggity, tienen a los clientes como usuarios principales agilizando la consulta de datos de los servicios de aggity.

Este cliente ha de tener contratado algún software de aggity para poder aprovechar este proyecto. Respecto a la parte de gestión de incidencias, no será necesario que el cliente disponga de algún servicio contratado porque el portal de mantenimiento incorporará funcionalidades básicas de un gestor de incidencias gracias a una aplicación externa.

Clientes finales: El portal de mantenimiento permitirá, a los clientes finales, enterarse de cualquier error o retraso que haya podido surgir en el producto solicitado al cliente gracias a mensajes automatizados. Esto reducirá el tiempo de comunicación entre cliente y cliente final.

En la Figura 1 se ofrece un diagrama donde se observa cada actor y su relación con el software a utilizar en este proyecto. Cabe recalcar que el Portal de mantenimiento (proyecto), aunque aparezca dos veces en el diagrama, representan

el mismo servicio. Se ha decidido representarlo de esta manera para dejar claro que aggity ofrece este servicio a los clientes y, de la misma manera, los clientes pueden ofrecer algunas funcionalidades de este servicio a los clientes finales.

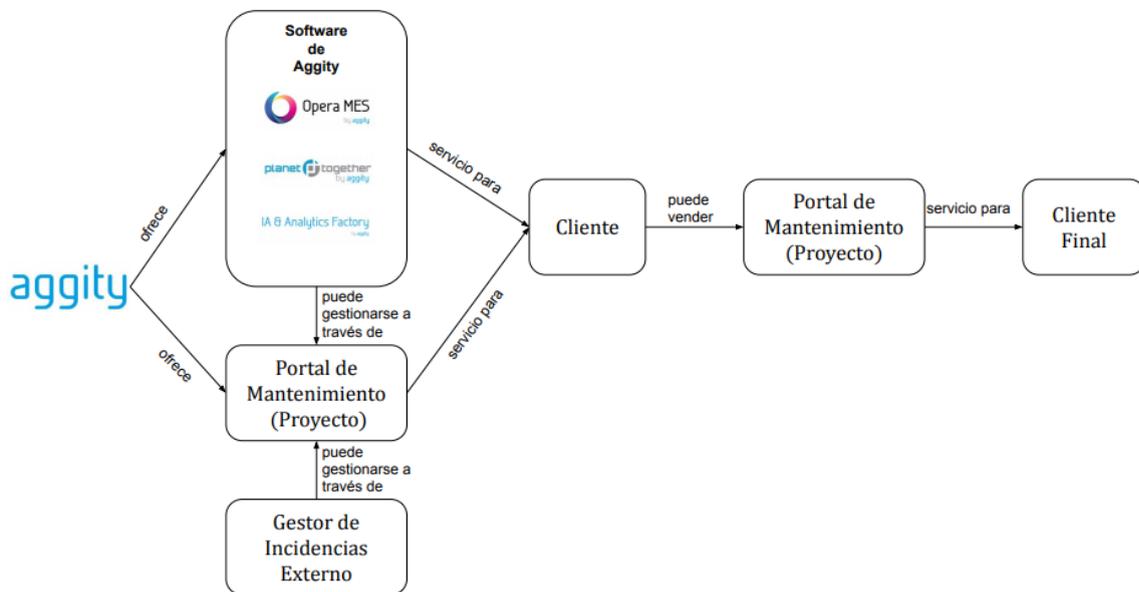


Figura 1: Diagrama de actores y servicios implicados en el proyecto. Elaboración Propia

2. Justificación

En la actualidad, aggity dispone de tres herramientas enfocadas al sector industrial:

Opera MES: Software enfocado en la gestión y control de actividades de Producción, Calidad, Materiales y Mantenimiento

Planet Together: Herramienta que permite la planificación y programación de las actividades de producción, en función de los recursos y materiales que dispone la fábrica.

IA & Analytics Factory: Solución analítica para la mejora de la producción industrial mediante Inteligencia Artificial.

Actualmente formo parte del departamento de Industria encargado de dar soporte a la herramienta Opera MES.

Para empezar, es necesario que el cliente haya comprado cualquiera de los servicios anteriormente mencionados, ya que el *portal de mantenimiento* trabaja de manera paralela a estos.

Este proyecto pretende ofrecer una plataforma donde poder gestionar el estado de los servicios de aggity. Existen empresas como SAP [3], especializadas en ofrecer soluciones de software para empresas como SAP Manufacturing Execution [4], sistema MES similar a Opera MES.

SAP dispone de un portal web, llamado SAP for me, donde los clientes pueden gestionar y revisar el estado del software de SAP en la nube, así como recibir alertas ante cualquier fallo en el sistema, similar al objetivo que intentamos conseguir con este proyecto.

En este caso, la elección del cliente irá más en función del ecosistema, aggity o SAP, con el que quiera trabajar y le encaje mejor en su metodología de trabajo, que en el sistema de gestión de servicios de cada ecosistema.

Respecto a la parte de gestión de incidencias, existen diversas herramientas [5] que cumplen con este objetivo, sin embargo, el objetivo principal de nuestro servicio es facilitar y agilizar estas acciones. Por eso hemos decidido incluir las funcionalidades de un gestor de incidencias directamente en el portal de mantenimiento. Creemos que tener todas las herramientas centralizadas en una plataforma, es motivo suficiente para vender el producto.

En resumen, existen herramientas que cumplen los mismos objetivos que nuestro portal de mantenimiento, sin embargo, son herramientas que están diseñadas para un ecosistema de aplicaciones en específico y, por lo tanto, no son viables para el ecosistema de aggity. Por ese motivo es necesario continuar con el desarrollo de este proyecto para aggity.

3. Alcance del Proyecto

Es un proyecto con un alcance bastante amplio, por ese motivo es necesario definir los objetivos genéricos principales y estudiar los riesgos que puedan reducir considerablemente su viabilidad.

3.1 Objetivo Principal

Como se ha mencionado en la contextualización, el objetivo principal del proyecto es un portal web que permita gestionar y ver el estado de los distintos servicios de aggity, así como facilitar y automatizar la comunicación de incidencias entre aggity y cliente, y entre un cliente y el cliente final.

Para esto será necesario cumplir los siguientes objetivos:

3.1.1 Conexión con servicios aggity

Conseguir una conexión en tiempo real con las herramientas de aggity permitirá detectar, por parte del cliente como por parte de aggity, cualquier incidencia que pueda surgir. Esto incluye el estado del servicio y cualquier incidencia relacionada con el funcionamiento interno de cada herramienta de aggity de manera individual.

3.1.2 Gestor de Incidencias

Incorporar a nuestro portal, un gestor de incidencias externo que permita al cliente controlar el estado de cualquier incidencia que pueda surgir con las herramientas de aggity.

Actualmente en aggity, disponemos de una herramienta para el control de incidencias llamada Jira [6], pero existen diversas herramientas Open Source, como Request Tracker [7] que pueden ser más fáciles de implementar. Será necesario hacer un análisis de las opciones disponibles y su compatibilidad con otras herramientas.

3.2 Objetivos Secundarios

El objetivo principal, al ser tan amplio, deriva en que ha de tener unos estándares mínimos para su distribución, entre los que encontramos:

3.2.1 Protección de Datos

En aggity, cabe la posibilidad de que los servicios ofrecidos trabajen con datos privados relacionados con nuestros clientes. Esto nos obliga a implementar un sistema de seguridad robusto y fiable que dé garantías a los futuros usuarios.

3.2.2 Gestión eficiente de los datos

Debido a la gran cantidad de datos con los que trabajaremos, tendremos que utilizar algoritmos que tratan de manera eficiente los datos para reducir la carga de trabajo del servidor y así conseguir un servicio funcional.

3.2.3 Gestión de usuario

Para poder realizar el objetivo principal correctamente, será necesario implementar un sistema que permita gestionar cualquier tipo de cliente. Este sistema ha de poder crear, editar y asignar permisos a usuarios en función del cliente y las herramientas que tenga contratadas.

3.3 Requerimientos no funcionales

Aparte de los objetivos principales y secundarios, será necesario que nuestro portal web cumpla unos requerimientos no funcionales que permitan una navegación sencilla y atractiva al consumidor:

3.3.1 Diseño dinámico

En función de los permisos otorgados a cada usuario, se ha de ser capaz de mostrar u ocultar las distintas secciones que conforman el portal de mantenimiento.

3.3.1 Usabilidad de la web

Uno de los aspectos importantes del servicio, es facilitar la visualización de los datos a través del portal, por lo tanto, tiene que ser un factor a tener en cuenta al diseñar la web.

3.3.1 Compatibilidad con navegadores

Nuestro sistema ha de funcionar correctamente en distintos dispositivos y distintos navegadores en función de unos requisitos mínimos que determinaremos a lo largo del proyecto.

3.3.1 Diseño dinámico

En función de los permisos otorgados a cada usuario, se ha de ser capaz de mostrar u ocultar las distintas secciones que conforman el portal de mantenimiento.

3.3.1 Accesibilidad (lenguaje)

Este producto está dirigido a empresas de distintos ámbitos y localizaciones, por ese motivo, tendremos que habilitar la web en 3 idiomas: inglés, español y catalan.

3.4 Obstáculos y Riesgos

Existen riesgos que son intrínsecos de un proyecto de software y, en nuestro caso, riesgos derivados de la empresa. A continuación se listan dichos riesgos y qué medidas se han pensado para reducir su impacto en el proyecto.

3.4.1 Compatibilidad

Para conseguir el objetivo principal, es necesario el estudio y uso de software externo al proyecto en lo referente al gestor de incidencias. Por ese motivo, dedicaremos recursos en el diseño y test de prototipos para probar las distintas conexiones y funcionalidades necesarias en el proyecto, y poder escoger la herramienta que mejor encaje con nuestra idea de proyecto.

3.4.2 Errores de programación

Es normal que en proyectos de esta envergadura puedan surgir problemas relacionados con la codificación. Se diseñarán test para comprobar el correcto funcionamiento de cada parte del proyecto y se incorporará un control de versiones para reducir la aparición de los mismo.

3.4.3 Carga de trabajo

Es posible que en ciertas etapas del proyecto, exista un exceso de trabajo por parte de la empresa que provoque una reducción en los recursos destinados a este proyecto. Por ese motivo, se ha decidido realizar el proyecto fuera del horario laboral. De esta manera podemos asegurar que la carga de trabajo empresarial no afectará a los plazos del proyecto.

3.4.4 Situación actual

Al estar en una situación excepcional debido a la pandemia, no es posible prever cualquier cambio en la situación laboral u otros problemas derivados de ello. Aunque es posible mantener cierto nivel de eficiencia laboral desde casa, es normal que los tiempos de comunicación entre los trabajadores se vean afectados. Aunque es imposible controlar esto, se tendrá en cuenta a la hora de escoger la metodología para reducir al mínimo su impacto implementando comunicación asíncrona si es necesario.

4. Metodología y rigor

A la hora de escoger la metodología más adecuada, hemos tenido en cuenta la envergadura del proyecto, los recursos disponibles (personas, tiempo) y los riesgos explicados anteriormente.

4.1 Metodología de trabajo

Se ha decidido trabajar con una metodología ágil, en concreto la metodología Scrum. ya que nos aporta una flexibilidad necesaria dadas las circunstancias.

4.1.1 Funcionamiento de la metodología Scrum

Para empezar, será necesario recoger los requerimientos, funcionalidades y tareas que abarca el proyecto a lo que llamaremos Product Backlog.

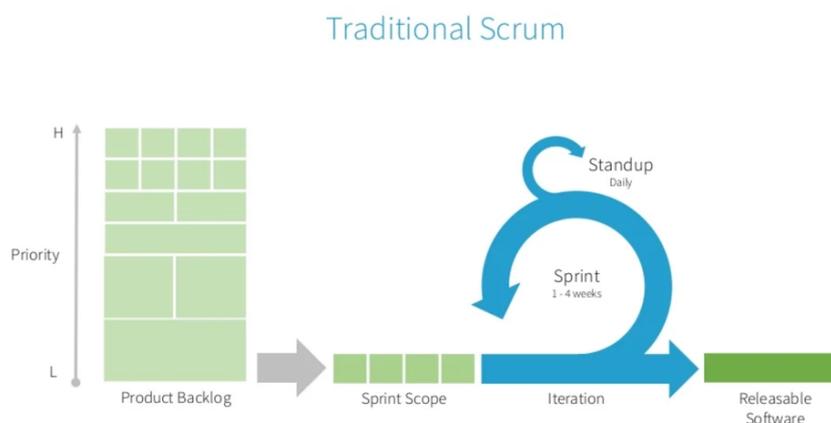


Figura 2: Esquema de la metodología Scrum. Imagen obtenida de la web OpenWebinars [8]

Una vez realizado el paso inicial, será necesario definir el coste de las tareas y a quien se las asignará. Es un paso delicado, ya que se tendrán que detectar las tareas que tengan un volumen grande y se intentará dividir en tareas de menor tamaño o determinar metas dentro de la misma tarea.

A continuación, las tareas se irán desarrollando en ciclos cortos llamados sprint[9] donde han de cumplirse ciertas condiciones:

- La duración no ha de ser superior a un mes.
- No se pueden realizar cambios drásticos que afectan al objetivo final.
- Ha de tener unos estándares de calidad mínimos.

Estos ciclos se suelen acompañar con reuniones diarias cortas con la intención de informar, a todo el equipo, del estado de las tareas.

Se finalizará con una revisión de las tareas del sprint. En esta etapa se comprobará el estado de las tareas en función de lo planeado, problemas que hayan surgido y la solución propuesta entre otras cosas. La intención de estas reuniones es la de detectar problemas a solucionar de cara al nuevo ciclo.

4.1.2 Motivos de la elección y funcionamiento del equipo

En el departamento de industria de aggity, actualmente trabajamos con la metodología Kanban, sin embargo, suele estar enfocado más al flujo de trabajo constante mezclando diversos proyectos y por eso lo he descartado.

Scrum es una metodología más flexible que permite revisar el avance del proyecto en intervalos más cortos, permitiéndonos detectar cualquier problema sin detener el flujo de trabajo.

Debido a la carga de trabajo que tenemos actualmente en aggity, se ha decidido realizar el proyecto de manera unipersonal, dedicando un tiempo diario menor pero asegurándonos que los plazos no se verán afectados por trabajo ajeno al proyecto.

Al ser jornadas de trabajo tan cortas, se ha decidido prescindir de las reuniones diarias a cambio de una reunión semanal, que es lo que durará un sprint. Esto nos servirá para detectar cualquier problema que surja y poder solucionarlo, reduciendo al máximo el desvío que pueda provocar tanto a nivel económico como temporal.

Para gestionar las diferentes etapas del proyecto, se ha decidido utilizar una herramienta gratuita llamada ProjectWork. ProjectWork es un complemento de Google Drive que nos permite crear y gestionar las distintas tareas que se realizan. Podremos observar el estado y la dedicación de cada tarea, lo que nos permitirá detectar cualquier desvío en la planificación.

También usaremos una herramienta muy conocida de control de versiones llamada GitHub. Esta herramienta nos permitirá guardar las distintas fases por las que pasa el código del proyecto y poder recuperarlo ante cualquier incidencia. Se creará un repositorio privado con una rama principal (master) y una rama de desarrollo donde se realizarán y probarán los cambios.

5. Planificación Temporal

5.1 Introducción

El proyecto supondría una dedicación aproximada de 520 h. que se realizará a lo largo de 27 semanas en jornadas (de lunes a viernes) de 4 horas diarias. La poca dedicación diaria al proyecto es debido a las actividades laborales que desempeño actualmente en aggity. Al tener diversos proyectos en desarrollo, se ha decidido realizar el proyecto fuera del horario laboral asegurando unas horas mínimas de dedicación.

El proyecto va iniciarse, con la sesión informativa de GEP, el día 23 de febrero de 2021 y tiene previsto finalizar el 20 de agosto de 2021. Entre el fin previsto y el próximo turno de lectura, que se realizará en el mes de septiembre de 2021 [[10](#)], hay 1 semana (20 h.) de margen para gestionar cualquier retraso que pueda surgir en medio del desarrollo.

5.2 Recursos

Tras un estudio del proyecto, se ha determinado que son necesarios una serie de recursos, listados en la Tabla 1, para el correcto desarrollo del mismo. A continuación, podremos encontrar cuales son y una breve explicación.

5.2.1 Humanos

- **[H-JP] Jefe de Proyecto:** Se encarga de estudiar el alcance del proyecto, diseccionar las tareas a realizar y el coste de estas. Suele interactuar con el cliente con la intención de estudiar y plasmar la idea del proyecto en papel. En este caso, tanto yo como mi director y ponente tendremos el rol de jefe de proyecto.
- **[H-P] Programador:** Persona o grupo de personas encargada de estudiar el software que se va a utilizar en el proyecto y desarrollar todas las funcionalidades planificadas por el jefe de proyecto. Debido a la carga de trabajo y para realizar una estimación fiable, seré el único programador asignado a este proyecto.
- **[H-T] Tester:** La función del Tester es la de comprobar que las funcionalidades del proyecto funcionan de acorde al estudio realizado por el jefe de proyecto. En este caso, también realizaré la función de Tester.

5.2.2 Software

- **[S-G] Ganttter:** Software online utilizado para crear el diagrama de Gantt del proyecto.

- **[S-SSMS] SQL Server Management Studio:** Aplicación que utilizaremos para crear y gestionar las bases de datos (DB) del proyecto.
- **[S-VS] Visual Studio 2019:** IDE principal que usaremos para desarrollar la aplicación web. Utilizaremos ASP.NET [11] que es un framework comercializado por Microsoft.
- **[S-GH] Github:** Software donde alojaremos nuestro proyecto para tener un control de versiones.
- **[S-GI] Gestores de incidencia (Jira y Request Tracker):** Software destinados a la creación y mantenimiento de tickets de incidencia.
- **[S-T] SourceTree:** Software gratuito para interactuar con el control de versiones. [12]
- **[S-GD] Google Drive:** Servicio para crear, editar y alojar archivos.
- **[S-AG] Opera MES, PlanetTogether, Analytics Factory:** Software de aggity diseñado para la transformación digital de los negocios.
- **[S-CH] Google Chrome:** Este será nuestro navegador predeterminado para las distintas versiones del proyecto.

5.2.3 Hardware

- **[HW-S] Servidor:** En este caso, aggity tiene preparado un servidor con las siguientes características: Intel Xeon E5-2650 v4 2.20GHz de procesador con 12GB de RAM y el sistema operativo Windows Server 2016 Standard de 64 bits.
- **[HW-O] Ordenador:** No se necesita un ordenador portátil con gran potencia. Usaremos uno con las siguientes especificaciones: Intel Core i7-6600U 2.60GHz de procesador con 8GB de RAM y el sistema operativo Windows 10 Pro de 64bits.

5.3 Descripción de tareas

A continuación, describiremos las tareas a realizar agrupadas en bloques. En la Tabla 1 está definido la cantidad de horas teóricas que se dedicará a cada tarea junto a los recursos necesarios para realizarla. También se dispone de la Figura 1 y la Figura 2 donde se representa el tiempo dedicado a cada tarea mediante un diagrama de Gantt.

5.3.1 Gestión del Proyecto - GP

Aquí se agrupan todas las tareas relacionadas con el estudio previo del proyecto necesario para el éxito del mismo. En este caso, también incluimos todo lo relacionado con la presentación del TFG, como las reuniones o los tiempos dedicados a documentar.

- **[GP-1] Alcance:** Para una correcta realización de un proyecto, es necesario definir los límites del proyecto . Se dedicarán 25 horas para documentarte y planificar correctamente, para sentar las bases del proyecto.
- **[GP-2] Planificación:** Una vez realizado el alcance, es necesario una planificación, tanto de tiempo como de recursos, de las distintas tareas que componen el proyecto, siempre teniendo en cuenta los obstáculos detectados en el alcance. Se le ha dedicado 20 horas. Depende del Alcance (GP-1).
- **[GP-3] Presupuesto y Sostenibilidad:** El coste del proyecto y personal son necesarios para que el proyecto llegue hasta la fase final, siempre pensando en el impacto ambiental derivado del desarrollo. Usaremos varios software que requieren la compra de licencias, por este motivo le dedicaremos 20 horas. Depende de la Planificación (GP-2).
- **[GP-4] Documentación:** Para realizar correctamente el TFG, es necesario documentar los distintos ciclos por los que pasa el proyecto. Se ha estimado unas 5 h. por cada semana de trabajo (18), lo que da un total de 90 h.
- **[GP-5] Presentación:** Para determinar la duración de esta tarea, se ha tenido en cuenta presentaciones previas realizadas en la universidad. Se dedicarán 22 horas para la preparación de diapositivas y prácticas de la presentación. Depende de la Documentación (GP-4) y la Fase de Testeo (FF-1).
- **[GP-6] Reuniones:** Al desglosar el proyecto en tareas de poca duración, hemos decidido que la duración del sprint sea de 1 semana con una reunión al finalizar cada sprint. Eso nos da una duración de 24 h. más 10 h. extra para cualquier reunión con el director o ponente. En total 34 h.

5.3.2 Trabajo Previo - TP

Todas las tareas necesarias, para el correcto funcionamiento del software necesario para el proyecto, están agrupadas aquí

- **[TP-1] Aprendizaje Software:** Como programador, no he trabajado con gran parte del software que utilizaremos en este proyecto. Por ese motivo, dedicaremos 40 horas para aprender el funcionamiento de las siguientes herramientas: ASP.NET, Analytics Factory, PlanetTogether, Jira, Request Tracker.

- **[TP-2] Estudio Requisitos Mínimos:** Una vez realizado el aprendizaje, podremos determinar los requisitos mínimos, tanto del sistema como del hardware, para el correcto funcionamiento de todas las aplicaciones involucradas en el proyecto. Dedicaremos 10 h. . Depende del Aprendizaje del Software (TP-1).
- **[TP-3] Preparación Entorno:** Prepararemos el entorno de prueba, en el servidor y en el ordenador, instalando todo el software necesario para empezar a desarrollar el proyecto. Con 16 h. tendremos el tiempo suficiente para instalar todas las aplicaciones y sus dependencias, si las tuvieran. Depende del Estudio de Requisitos Mínimos (TP-2)

5.3.3 Diseño - DI

Aquí se agrupan las tareas relacionadas con el diseño tanto del front-end como del back-end.

- **[DI-1] Estudio necesidades mínimas Usuario:** Junto al jefe de proyecto, se determinará las condiciones de diseño que ha de cumplir una aplicación web como, por ejemplo, qué tipo de cifrado tendrá las contraseñas guardadas en la BD. Se dedicarán 5 h. para aclarar estos temas.
- **[DI-2] Diseño Web:** En función de lo hablado entre el cliente y el jefe de proyecto, se define el diseño a nivel de front-end. Se ha de definir por cada estado por la que pasa la aplicación web. Se le dedicará 20 h. . Depende de las necesidades mínimas del Usuario (DI-1)
- **[DI-3] Diseño Base de Datos:** Se establecerán las tablas necesarias y sus relaciones para guardar los datos sin caer en la redundancia. Es una tarea importante porque tiene un gran impacto en el rendimiento. Se le dedicarán 24 h. por su importancia. Depende de las necesidades mínimas del Usuario (DI-1)

5.3.4 Primera Fase Desarrollo - FD

- **[FD-1] Creación Base de Datos:** Una vez aclarado el diseño, creamos la BD principal donde se guardarán los datos recogidos de la aplicación web. Será una tarea de 12 h. Depende del Diseño de la Base de Datos (DI-3)
- **[FD-2] Preparar Proyecto ASP.NET:** Empezaremos creando la base de la aplicación que será el core del portal. Comprobaremos que se conecta a la BD sin problemas. Serán 12 h.de trabajo. Depende del Diseño Web (DI-2)

5.3.5 Prototipos - PT

Debido a los distintos software que usamos, se ha decidido crear pequeños prototipos donde comprobaremos las distintas conexiones con la aplicación web para detectar, desde el principio, posible incompatibilidades que puedan retrasarnos

- **[PT-1] Preparar Base de Datos de Prueba:** Prepararemos una DB de prueba por cada software de aggity simulando su funcionamiento. Un total de 8 h. .
- **[PT-2] Funcionamiento API Software aggity:** Probamos que el software funcione correctamente y creamos una aplicación web para probar las funciones que utilizaremos en el proyecto final. Se le dedicará 12 h. Depende de preparar el proyecto ASP.NET (FD-2) y Preparar la Base de Datos de Prueba (PT-1)
- **[PT-3] Compatibilidad Sistema de Incidencias:** En este caso tenemos dos posibilidades en mente. Comprobamos el funcionamiento de ambos y, en el caso de que ambos sean compatibles con la aplicación web, estudiaremos las ventajas y desventajas de cada software para decantarnos por uno. 7 h. por cada software, 14 h. en total. Depende de preparar el proyecto ASP.NET (FD-2)

5.3.6 Gestión de Usuario - GU

Todo el desarrollo de la interfaz con el usuario y sus funcionalidades se encuentra en este grupo.

- **[GU-1] Login de Usuario:** Desarrollaremos las funciones e interfaz necesaria para que el cliente pueda conectarse o recuperar su contraseña de manera segura. Es una tarea de 12 h. Depende de preparar el proyecto ASP.NET (FD-2)
- **[GU-2] Editar Usuario:** Añadimos la funcionalidad para editar datos del cliente. 8 h. de trabajo. Depende de tener desarrollado el Login del Usuario (GU-1)
- **[GU-3] Editar Clientes Finales:** Los clientes tendrán la opción de añadir algún cliente final o eliminarlo. Similar a la tarea de editar usuario, esta tendrá una duración de 8 h. Depende de tener desarrollado el Login del Usuario (GU-1)
- **[GU-4] Control de Permisos de Clientes Finales:** Aquí se configurarán, qué clientes finales podrán recibir avisos desde la web, y de qué servicios de aggity . Se necesitarán 12 h. ya que es necesario implementar un envío de mensaje a través de un servidor SMTP. Depende de poder Editar Clientes Finales (GU-4)

5.3.7 Gestión de Sistema de Incidencias - GSI

Todo el desarrollo de la interfaz con el software de incidencia se encuentra aquí.

- **[GSI-1] Integración Sistema de Incidencia:** Incluiremos el sistema escogido en PT-3 en el servidor y comprobamos que las funcionalidades se mantienen. Le dedicaremos 8 h. Depende de preparar el proyecto ASP.NET (FD-2)
- **[GSI-2] Formulario Creación de Incidencia:** Se crea un formulario web que comunique, a aggity, cualquier incidencia y que quede registrado en la BD junto a cualquier comentario del error. Se le ha estimado 10 h. . Depende de haber integrado el Sistema de Incidencias (GSI-1).

- **[GSI-3] Listado Estado de Incidencia:** Recuperar, de la BD, la lista de incidencias por cliente y permitirle añadir cualquier comentario o archivo extra a la incidencia. Se han previsto 10 h. para esta tarea. Depende de haber integrado el Sistema de Incidencias (GSI-1).

5.3.8 Gestión de los servicios de aggity - GSA

En este grupo se encuentran las tareas relacionadas con las herramientas de aggity. Consisten en implementar la lectura de datos, de los servicios de aggity, por la aplicación web.

- **[GSA-1] Implementación API Software aggity:** Implementamos la lectura de datos por parte de la aplicación web. Para esta tarea, requerimos menos tiempo ya que cualquier inconveniente será encontrado y solucionado en el prototipo. En total 8 h. . Depende de preparar el proyecto ASP.NET (FD-2)
- **[GSA-2] Listado Estado Servicios aggity:** Con los datos leídos, facilitamos la lectura de los datos por parte del cliente en función del diseño definido anteriormente. Le dedicaremos 12 h. . Depende de haber implementado el API de Software de aggity (GSA-1)
- **[GSA-3] Envío Incidencias Clientes Finales:** Una vez hayamos incorporado los clientes finales a la gestión de usuario, podemos incorporar los avisos en tiempo real de cualquier fallo en la producción de alguno de sus productos. Se le dedicará 12 h. .Depende de haber implementado el API de Software de aggity (GSA-1)

5.3.9 Fase Final Desarrollo - FF

- **[FF-1] Fase de Testeo:** Cada tarea que implique el desarrollo de software, no se dará por finalizada hasta que pase por una fase de testeo para no arrastrar errores entre tareas. Se le dedicará 3 h. por cada tarea con necesidad de testeo (12) dando un total de 36 h.

5.4 Dependencias de Tareas

Las dependencias de cada tarea pueden encontrarse en la Tabla 1 junto a los recursos que utilizan. La mayoría de tareas son dependientes de otras tareas del mismo grupo salvo la presentación (GP-5), donde se necesita acabar la documentación y la fase de test del proyecto, y la preparación del proyecto ASP.NET (FD2) que, al ser el núcleo del proyecto, es necesario finalizarlo para proseguir con el mismo.

En la Figura 2 se observa el esquema de Gantt con todas las tareas del proyecto junto a los recursos humanos necesarios.

Tarea	Cód.	Duración (h)	Dependencias	Recursos (Humano Software Hardware)
Gestión de Proyecto	GP			
Alcance	GP-1	25		H-JP S-GD HW-O
Planificación	GP-2	20	GP-1	H-JP S-G, S-GD HW-O
Presupuesto y sostenibilidad	GP-3	20	GP-2	H-JP S-GD HW-O
Documentación	GP-4	90		H-JP S-GD HW-O
Presentación	GP-5	22	GP-4, FF-1	H-JP S-GD HW-O
Reuniones	GP-6	34		H-JP, H-P, H-T S-CH HW-O
Trabajo Previo	TP			
Aprendizaje Software	TP-1	40		H-P S-CH, S-T, S-AG HW-O
Estudio requisitos mínimos	TP-2	10	TP-1	H-P S-CH, S-AG, S-GI HW-O
Preparación Entorno	TP-3	16	TP-2	H-P S-CH, S-GH, S-T, S-AG, S-GI HW-O, HW-S
Diseño	DI			
Estudio necesidades mínimas usuario	DI-1	5		H-P S-SSMS, S-AG -
Diseño Web	DI-2	20	DI-1	H-P - HW-O
Diseño Base de Datos	DI-3	24	DI-1	H-P - HW-O
Primera Fase Desarrollo	FD			
Creación Base de Datos	FD-1	12	DI-3	H-P S-SSMS HW-O, HW-S
Preparar Proyecto ASP.NET	FD-2	12	DI-2	H-P S-VS, S-GH, S-T HW-O, HW-S
Prototipos	PT			
Preparar Base de Datos de prueba	PT-1	8		H-P S-SSMS HW-O, HW-S
Funcionamiento API software aggity	PT-2	12	PT-1, FD-2	H-P S-VS, S-AG HW-O, HW-S
Compatibilidad de Sistema de Incidencias	PT-3	14	FD-2	H-P S-VS, S-GI HW-O, HW-S
Gestion de Usuario	GU			
Login de Usuario	GU-1	12	FD-2	H-P S-VS, S-GH, S-T HW-O, HW-S
Editar Usuario	GU-2	8	GU-1	H-P S-VS, S-GH, S-T HW-O, HW-S
Editar Clientes Finales	GU-3	8	GU-1	H-P S-VS, S-GH, S-T HW-O, HW-S
Control de Permisos de Clientes Finales	GU-4	12	GU-3	H-P S-VS, S-GH, S-T HW-O, HW-S
Gestión del Sistema de Incidencias	GSI			
Integración Sistema de Incidencia	GSI-1	8	FD-2	H-P S-VS, S-GH, S-T, S-GI HW-O, HW-S
Formulario de Creación de Incidencia	GSI-2	10	GSI-1	H-P S-VS, S-GH, S-T, S-GI HW-O, HW-S
Listado de Estado de Incidencias	GSI-3	10	GSI-2	H-P S-VS, S-GH, S-T, S-GI HW-O, HW-S
Gestión de los servicios de aggity	GSA			
Implementación API software aggity	GSA-1	8	FD-2	H-P S-VS, S-GH, S-T HW-O, HW-S
Listado de Estado Servicios aggity	GSA-2	12	GSA-1	H-P S-VS, S-GH, S-T HW-O, HW-S
Envío Incidencias Clientes Finales	GSA-3	12	GSA-1	H-P S-VS, S-GH, S-T HW-O, HW-S
Fase Final Desarrollo	FF			
Fase de Testeo	FF-1	36		H-T S-CH, S-AG HW-O, HW-S
Total Horas		520		

Tabla 1. Resumen de las tareas del proyecto y sus dependencias. Elaboración Propia.

solucionar cualquier retraso fuera de la planificación original. Esta tarea extra tiene una duración de 1 semana o 20 h.

5.6 Cambios respecto a la planificación inicial

Se ha producido un cambio de nombre de proyecto y de algunas tareas. Hemos eliminado todo lo relacionado con cliente final, respecto a la planificación inicial, debido a que hemos detectado que la mayoría de clientes de Aggity, no utilizan una metodología de trabajo fabrica - cliente que encaje con nuestra idea de cliente final.

A la hora de empezar el proyecto, se han detectado algunas tareas y dependencias que no eran funcionales con la metodología de trabajo que se estaba llevando a cabo. Esto era debido a la dependencia entre algunos pares de tareas

Al trabajar con diversos tipos de herramientas (software), al principio se intentó agrupar subtareas similares, aplicadas a distintas herramientas, en una misma tarea. Esto provoca que existan tareas que sean dependientes debido a la naturaleza de la tarea en sí, pero si las tratáramos de manera individual, en función de la herramienta, podemos paralelizar subtareas que se enfoquen en distintas herramientas.

Por este motivo, se ha decidido eliminar algunas dependencias para permitir el trabajo paralelo.

También se han añadido algunas funcionalidades que no se detectaron en las fases iniciales de la planificación pero, debido a la poca dedicación que se necesita para implementarlo, se ha decidido añadirlo a la tarea GSA-2. Esta tarea nueva implica diversas acciones que se podrán realizar, desde la web, sobre los servicios tales como iniciar o reiniciarlos.

Se ha reducido la duración de las tareas PT-2 y GSA-1 debido a que una de las herramientas de Aggity aún está en desarrollo, por lo tanto, no se han podido realizar pruebas suficientes para implementarla en el sistema.

Como último, en la planificación inicial del proyecto, se planteó un tramo de 2 semanas extras ante cualquier incidencia que pudiera haber surgido en el desarrollo. Se ha decidido añadir estas semanas como tarea para poder incluirlas en el Gantt ya que se ha hecho uso de ellas.

Estas incidencias se han producido en las tareas GSI-3 y GSA-2, debido a problemas con el proyecto ASP.NET a la hora de visualizar los datos correctos en el portal.

En la tabla 2 se ven reflejados los distintos cambios en la planificación .

Tarea	Cód.	Duración (h)	Dependencias	Recursos (Humano Software Hardware)
Gestión de Proyecto	GP			
Alcance	GP-1	25		H-JP S-GD HW-O
Planificación	GP-2	20	GP-1	H-JP S-G, S-GD HW-O
Presupuesto y sostenibilidad	GP-3	20	GP-2	H-JP S-GD HW-O
Documentación	GP-4	90		H-JP S-GD HW-O
Presentación	GP-5	22	GP-4, FF-1	H-JP S-GD HW-O
Reuniones	GP-6	34		H-JP, H-P, H-T S-CH HW-O
Trabajo Previo	TP			
Aprendizaje Software	TP-1	40		H-P S-CH, S-T, S-AG HW-O
Estudio requisitos mínimos	TP-2	10	TP-1	H-P S-CH, S-AG, S-GI HW-O
Preparación Entorno	TP-3	16	TP-2	H-P S-CH, S-GH, S-T, S-AG, S-GI HW-O, HW-S
Diseño	DI			
Estudio necesidades mínimas usuario	DI-1	5		H-P S-SSMS, S-AG -
Diseño Web	DI-2	20	DI-1	H-P - HW-O
Diseño Base de Datos	DI-3	24	DI-1	H-P - HW-O
Primera Fase Desarrollo	FD			
Creación Base de Datos	FD-1	12	DI-3	H-P S-SSMS HW-O, HW-S
Preparar Proyecto ASP.NET	FD-2	12	DI-2	H-P S-VS, S-GH, S-T HW-O, HW-S
Prototipos	PT			
Preparar Base de Datos de prueba	PT-1	8		H-P S-SSMS HW-O, HW-S
Funcionamiento API software Aggity	PT-2	12 10	PT-1, FD-2	H-P S-VS, S-AG HW-O, HW-S
Compatibilidad de Sistema de Incidencias	PT-3	14 22	FD-2	H-P S-VS, S-GI HW-O, HW-S
Gestión de Usuario	GU			
Login de Usuario	GU-1	12	FD-2	H-P S-VS, S-GH, S-T HW-O, HW-S
Editar Usuario	GU-2	8	GU-1	H-P S-VS, S-GH, S-T HW-O, HW-S
Editar Clientes Finales	GU-3	8	GU-1	H-P S-VS, S-GH, S-T HW-O, HW-S
Control de Permisos de Clientes Finales	GU-4	12	GU-3	H-P S-VS, S-GH, S-T HW-O, HW-S
Gestión del Sistema de Incidencias	GSI			
Integración Sistema de Incidencia	GSI-1	8	FD-2	H-P S-VS, S-GH, S-T, S-GI HW-O, HW-S
Formulario de Creación de Incidencia	GSI-2	10	GSI-1	H-P S-VS, S-GH, S-T, S-GI HW-O, HW-S
Listado de Estado de Incidencias	GSI-3	10	GSI-2	H-P S-VS, S-GH, S-T, S-GI HW-O, HW-S
Gestión de los servicios de Aggity	GSA			
Implementación API software Aggity	GSA-1	8 6	FD-2	H-P S-VS, S-GH, S-T HW-O, HW-S
Listado de Estado Servicios Aggity e implementar acción de reinicio.	GSA-2	12 26	GSA-1	H-P S-VS, S-GH, S-T HW-O, HW-S
Envío Incidencias Clientes Finales	GSA-3	12	GSA-1	H-P S-VS, S-GH, S-T HW-O, HW-S
Fase Final Desarrollo	FF			
Fase de Testeo	FF-1	36		H-T S-CH, S-AG HW-O, HW-S
Tiempo dedicado a las incidencias	FF-2	40		H-T S-CH, S-AG HW-O, HW-S
Total Horas		570		

Tabla 1: Resumen de las tareas del proyecto en la etapa final y sus dependencias. Elaboración Propia

6. Presupuesto

6.1 Identificación y estimación de los Costes

Ya habiendo identificado los distintos recursos necesarios para desarrollar correctamente el proyecto, procederemos a calcular los costes en función del tipo de recurso. También tendremos que añadir, a la estimación de costes, los gastos producidos en servicios (Internet, Luz, Workspace, etc) y posibles imprevistos.

Debido a que el proyecto se realizará fuera del horario laboral, se ha decidido plantear la estimación de los costes desde un punto ajeno a la empresa aggity. Por ese motivo, se incluirán los costes de software propiedad de aggity. Del mismo modo, calcularemos todos los costes en base a datos recopilados por internet, sin tener en cuenta el coste real proporcionado por la empresa.

6.1.1 Costes de los recursos

- Humanos

Para los recursos humanos, será necesario calcular el sueldo bruto por hora (salario anual / (273 días laborables * 8 horas laborables por día)). Los salarios anuales se consultarán en la web LinkedIn [13], una de las mayores redes profesionales del mundo. Se tendrá en cuenta tanto la posición como la ubicación del puesto, que en nuestro caso será Barcelona, para el cálculo de coste listado en la Tabla 3.

En lo que respecta a la seguridad social, se ha cogido un porcentaje aproximado de lo expuesto en el Boletín Oficial del Estado del año 2019 [14].

Rol	Código	Salario bruto anual	Salario bruto/hora	Salario bruto + Seguridad Social (30% del salario bruto)
Jefe de Proyecto	H-JP	41.000€	18,77€	24,40€
Programador	H-P	31.000€	14,19€	18,45€
Tester	H-T	27.000€	12,36€	16,07€

Tabla 3. Sueldo bruto por hora en función del rol. Elaboración propia a partir de datos de la web LinkedIn Salary

Una vez calculado el salario bruto por hora medio de los distintos roles, será necesario calcular cuántas horas dedican cada rol a este proyecto. Para ello tendremos en cuenta la planificación realizada anteriormente y la distribución de recursos de la Tabla 1.

Algo importante a tener en cuenta, a la hora de revisar los costes por actividad, son las personas que desempeñan cada rol en el proyecto, explicado anteriormente en el apartado de Recursos Humanos.

Por este motivo, las 25 horas de la tarea Alcance (**GP-1**) están divididas entre Jefe de proyecto y Programador ya que esta tarea ha sido realizada de manera individual por mi. En cambio se ha decidido duplicar el coste en recursos humanos de la tarea Reuniones (**GP-6**) ya que, en este caso, yo realizaré las funciones de Programador y Tester, y el director junto al ponente del proyecto realizará las funciones de Jefe de Proyecto debido a que no tiene sentido reuniones de un solo rol en este proyecto.

Teniendo en cuenta todo lo mencionado anteriormente, obtenemos un coste de recursos humanos de 11.312,60€ desglosado por tareas en la Tabla 6.

- **Hardware**

El desarrollo de este proyecto requiere el hardware especificado en el apartado de planificación. El uso, tanto del ordenador portátil como del servidor, será continuo a lo del proyecto. Al coste de material, será necesario añadirle la amortización que se calculará con la siguiente fórmula:

$$\frac{\text{Coste del material (€)}}{\text{Vida útil (años)} * \text{Días laborables al año} * \text{Horas de uso diaria}} * \text{Duración del proyecto (horas)}$$

En nuestro caso, la vida útil será de 4 años con un uso diario de 4 horas por día. En la tabla 4 se observa el coste que implica el uso de dicho hardware en nuestro proyecto.

Hardware	Código	Coste	Amortización	Total
Servidor	HW-S	2100€	259,62€	2359,62€
Dell -Latitude E5470	HW-O	800€	98,90€	898,90€
Total Proyecto	-	-	358,52€	-

Tabla 4. Desglose del coste de los recursos físicos. Elaboración propia

- **Software**

Respecto a los costes por los recursos digitales (Software), todo el software utilizado es de uso gratuito o necesita la compra de una licencia anual o mensual que no necesita un coste extra por actualización. En este caso también utilizaremos la fórmula de amortización explicada anteriormente teniendo en cuenta que, al ser software, tiene una obsolescencia más rápida. Por ese motivo, lo amortizamos durante 2 años.

El software de aggity es bastante dinámico y puede variar su coste en función de los módulos contratados o el tipo de empresa interesada, por este motivo hemos decidido calcular el precio del servicio con las prestaciones mínimas para desarrollar el proyecto.

En la Tabla 5 se puede observar cuál será el coste total destinado al Software.

Software	Código	Tipo Licencia	Coste	Amortización	Total
Ganttter	S-G		0€	0€	0€
SQL Server Management Studio 2019	S-SSMS		0€	0€	0€
Visual Studio 2019	S-VS		0€	0€	0€
Github	S-GH		0€	0€	0€
SourceTree	S-T		0€	0€	0€
Google Drive	S-GD		0€	0€	0€
Software aggyty	S-AG		37.500€	4.464,29€	41.964,29€
Google Chrome	S-CH		0€	0€	0€
RequestTracker [15]	S-RT	Free	0€	0€	0€
Jira [16]	S-J	10 usuarios/año	0€	0€	0€
Total Coste	-	-	-	4.464,29€	-

Tabla 5. Desglose del coste de los recursos digitales. Elaboración propia con los precios recogidos de la web oficial de cada recurso.

- Servicios y entorno

En este grupo encontraremos los costes de servicios ajenos al proyecto y que son necesarios para el correcto funcionamiento del equipo. Estos gastos son: la electricidad, internet y el entorno de trabajo entre otros.

Al ser un proyecto de corta duración, se ha decidido optar por usar un servicio que nos ahorra la compra de inmobiliario y gastos de gestión. El servicio se llama MGCoworking [17] y nos dota de todos los servicios anteriormente nombrados.

Pese a que el proyecto está pensado para tener 3 roles de desarrollo, en nuestro caso, una persona realizará todo el desarrollo, por lo tanto, calcularemos los costes en función de una persona.

Servicio	Tipo de Contrato	Coste por mes	Nº de Meses	Total
Alquiler Coworking	105€/persona al mes	105€	6 meses	630€
Total Coste				630€

Tabla 6. Desglose del coste de los servicios ajenos al proyecto. Elaboración propia con los precios recogidos en diversas páginas.

6.1.2 Costes por actividad

Teniendo en cuenta la subdivisión de tareas y el tiempo estimado en la planificación, en la Tabla 7 se ha calculado el coste de cada tarea en función del tiempo empleado por cada integrante del equipo del proyecto.

Tarea	Código	Duración Total (h)	Horas por cada rol			Coste Total
			H-JP	H-P	H-T	
Gestión de Proyecto	GP					
Alcance	GP-1	25	20	5		580,25€
Planificación	GP-2	20	20			488€
Presupuesto y sostenibilidad	GP-3	20	20			488€
Documentación	GP-4	90	90			2196€
Presentación	GP-5	22	22			536,8€
Reuniones	GP-6	34	34	26	8	1.408,18€
Trabajo Previo	TP					
Aprendizaje Software	TP-1	40		40		738€
Estudio requisitos mínimos	TP-2	10		10		184,5€
Preparación Entorno	TP-3	16		16		295,2€
Diseño	DI					
Estudio necesidades mínimas usuario	DI-1	5		5		92,25€
Diseño Web	DI-2	20		20		369€
Diseño Base de Datos	DI-3	24		24		442,8€
Primera Fase Desarrollo	FD					
Creación Base de Datos	FD-1	12		12		221,4€
Preparar Proyecto <u>ASP.NET</u>	FD-2	12		12		221,4€
Prototipos	PT					
Preparar Base de Datos de prueba	PT-1	8		8		147,6€
Funcionamiento API software aggity	PT-2	12		12		221,4€
Compatibilidad de Sistema de Incidencias	PT-3	14		14		258,3€
Gestion de Usuario	GU					
Login de Usuario	GU-1	12		12		221,4€
Editar Usuario	GU-2	8		8		147,6€
Editar Clientes Finales	GU-3	8		8		147,6€
Control de Permisos de Clientes Finales	GU-4	12		12		221,4€
Gestión del Sistema de Incidencias	GSI					
Integración Sistema de Incidencia	GSI-1	8		8		147,6€
Formulario de Creación de Incidencia	GSI-2	10		10		184,5€
Listado de Estado de Incidencias	GSI-3	10		10		184,5€
Gestión de los servicios de aggity	GSA					
Implementación API software aggity	GSA-1	8		8		147,6€
Listado de Estado Servicios aggity	GSA-2	12		12		221,4€
Envío Incidencias Clientes Finales	GSA-3	12		12		221,4€
Fase Final Desarrollo	FF					
Fase de Testeo	FF-1	36			36	578,52€
Total Coste		520				11.312,60€

Tabla 7. Horas dedicadas a cada tarea, por rol, y el coste que implica. Elaboración propia.

6.1.3 Coste de contingencias e imprevistos

El coste de contingencias son los costes que es probable que se produzcan pero desconocemos la cantidad precisa. En proyectos informáticos suele rondar el 15% y está incluido en la Tabla 8 del presupuesto final.

Respecto a los riesgos detectados, las medidas tomadas para solucionarlos suponen un coste añadido al proyecto, como la fase de prototipo que ya está incluida en los costes por actividad (Tabla 6) con el código PT.

También se decidió reservar un margen de tiempo ante cualquier imprevisto que pueda surgir.

La tabla 8 refleja el coste que implicaría el uso de este margen.

Tarea	Código	Duración Total (h)	Horas por cada rol			Coste Total
			H-JP	H-P	H-T	
Margen de tiempo ante imprevistos		20		20		369€
Total Coste		20				369€

Tabla 8. Horas dedicadas a cualquier error de programación no contemplado. Elaboración propia.

6.1.4 Presupuesto final

Para finalizar, será necesario sumar los costes calculados anteriormente para saber el coste estimado del proyecto.

Tipo de Coste	Coste
Coste Recurso Hardware	358,52€
Coste Recurso Software	4.464,29€
Coste Recurso Servicios	360€
Costes Recursos Humanos	11.312,60€
Coste de Imprevistos	369€
Coste de Contingencias (15%)	2.529,66€
Estimación Coste Proyecto	19.394,07€

Tabla 9. Estimación total del coste del proyecto. Elaboración propia

6.2 Control de gestión

Es tan importante una buena planificación y cálculo del presupuesto, como de tener un buen control de las horas dedicadas a cada tarea y el coste que esto supone. Por ese motivo, es necesario establecer una manera de controlar el tiempo utilizado en cada tarea y detectar cualquier imprevisto no calculado en la planificación.

Utilizaremos una hoja de cálculo de Google Drive para tener registrado el tiempo empleado en cada tarea, con el fin de obtener los desvíos haciendo uso de los siguientes cálculos:

- **Desvío de mano de obra en precio:**
(consumo horas estimado – consumo horas real) * coste estimado
- **Desvío de mano de obra en consumos**
(consumo horas estimado – consumo horas real) * coste estimado
- **Desvío total Costes Fijos**

total coste fijo presupuestado – total coste fijo real

El registro se realizará al finalizar cada sprint para poder detectar cualquier desvío lo antes posible y saber de qué margen disponemos teniendo en cuenta el coste de contingencia. El desvío total de coste fijo se realizará una vez finalizada la fase de testeo.

6.3 Cambios en el presupuesto respecto al presupuesto inicial

Como comentamos anteriormente, una herramienta de desarrollo no ha sido posible incluirla y , por lo tanto, deberemos excluirla del coste en el desarrollo [Tabla 10] y actualizar la tabla de las horas dedicadas a cada tarea [Tabla 11]

Software	Código	Tipo Licencia	Coste	Amortización	Total
Ganttter	S-G		0€	0€	0€
SQL Server Management Studio 2019	S-SSMS		0€	0€	0€
Visual Studio 2019	S-VS		0€	0€	0€
Github	S-GH		0€	0€	0€
SourceTree	S-T		0€	0€	0€
Google Drive	S-GD		0€	0€	0€
Software Aggity	S-AG		37.500€ 20.000€	4.464,29€ 2.380,95€	41.964,29€ 22.380,95€
Google Chrome	S-CH		0€	0€	0€
RequestTracker [15]	S-RT	Free	0€	0€	0€
Jira [16]	S-J	10 usuarios/año	0€	0€	0€
Total Coste	-	-	-	4.464,29€ 2.380,95€	-

Tabla 10. Desglose del coste final de los recursos digitales. Elaboración propia con los precios recogidos de la web oficial de cada recurso.

Tarea	Código	Duración Total (h)	Horas por cada rol			Coste Total
			H-JP	H-P	H-T	
Gestión de Proyecto	GP					
Alcance	GP-1	25	20	5		580,25€
Planificación	GP-2	20	20			488€
Presupuesto y sostenibilidad	GP-3	20	20			488€
Documentación	GP-4	90	90			2196€
Presentación	GP-5	22	22			536,8€
Reuniones	GP-6	34	34	26	8	1.408,18€
Trabajo Previo	TP					
Aprendizaje Software	TP-1	40		40		738€
Estudio requisitos mínimos	TP-2	10		10		184,5€
Preparación Entorno	TP-3	16		16		295,2€
Diseño	DI					
Estudio necesidades mínimas usuario	DI-1	5		5		92,25€
Diseño Web	DI-2	20		20		369€
Diseño Base de Datos	DI-3	24		24		442,8€
Primera Fase Desarrollo	FD					
Creación Base de Datos	FD-1	12		12		221,4€
Preparar Proyecto ASP.NET	FD-2	12		12		221,4€
Prototipos	PT					
Preparar Base de Datos de prueba	PT-1	8		8		147,6€
Funcionamiento API software Aggity	PT-2	12 10		12 10		221,4€ 184,5€
Compatibilidad de Sistema de Incidencias	PT-3	14 22		14 22		258,3€ 405,9€
Gestion de Usuario	GU					
Login de Usuario	GU-1	12		12		221,4€
Editar Usuario	GU-2	8		8		147,6€
Editar Clientes Finales	GU-3	8		8		147,6€
Control de Permisos de Clientes Finales	GU-4	12		12		221,4€
Gestión del Sistema de Incidencias	GSI					
Integración Sistema de Incidencia	GSI-1	8		8		147,6€
Formulario de Creación de Incidencia	GSI-2	10		10		184,5€
Listado de Estado de Incidencias	GSI-3	10		10		184,5€
Gestión de los servicios de Aggity	GSA					
Implementación API software Aggity	GSA-1	8 6		8 6		147,6€ 110,7€
Listado de Estado Servicios Aggity	GSA-2	12 26		12 26		221,4€ 479,7€
Envío Incidencias Clientes Finales	GSA-3	12		12		221,4€
Fase Final Desarrollo	FF					
Fase de Testeo	FF-1	36			36	578,52€
Tiempo dedicado a las incidencias	FF-2	40		40		738€
Total Coste		570				12.382,70€

Tabla 11. Horas finales dedicadas a cada tarea, por rol, y el coste que implica. Elaboración propia.

Obteniendo un presupuesto final de 18.228,85€.

Tipo de Coste	Coste
Coste Recurso Hardware	358,52€
Coste Recurso Software	4.464,29€ 2.380,95€
Coste Recurso Servicios	360€
Costes Recursos Humanos	11.312,60€ 12.382,70€
Coste de Imprevistos	369€
Coste de Contingencias (15%)	2.529,66€ 2.377,68€
Estimación Coste Proyecto	18.228,85€

Tabla 12. Estimación total del coste del proyecto. Elaboración propia

Respecto a la planificación de horas, se ha producido un retraso de la fecha de finalización en 2 meses debido a 3 factores principales: problemas médicos, cálculo erróneo en el tiempo disponible para el trabajo y algunos cambios en las funcionalidades de la aplicación.

También se ha reflejado el tiempo dedicado a la resolución de incidencias como una nueva tarea: FF-2.

Como resultado, se ha calculado que la entrega de la memoria se hará la semana del 15 de octubre de 2021, como se puede observar en el siguiente diagrama de Gantt.

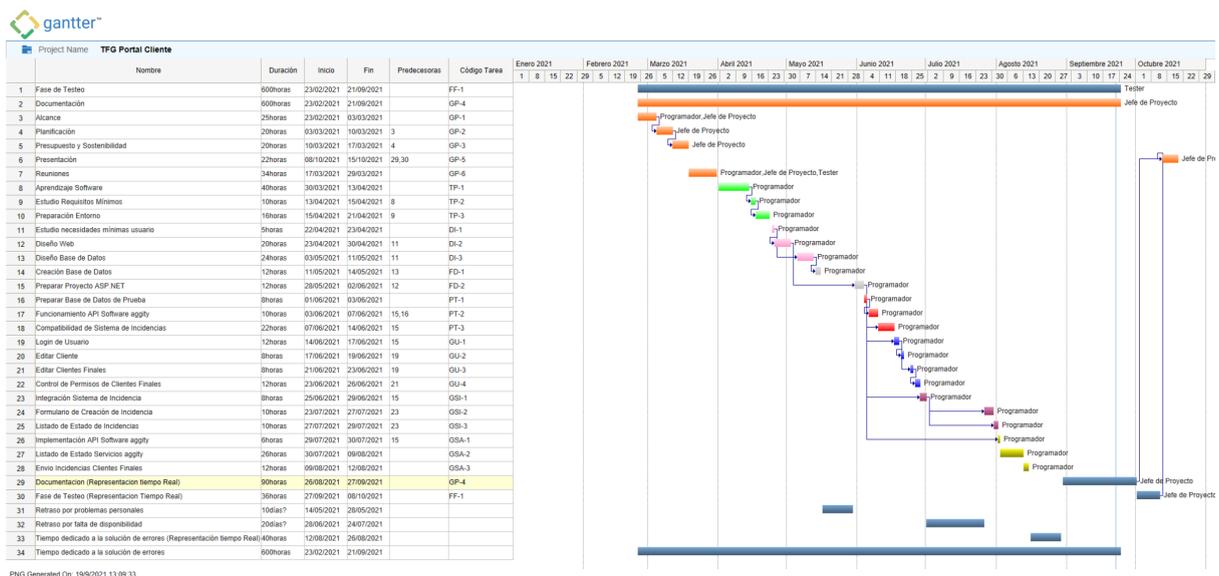


Figura 4. Diagrama de Gantt final. Elaboración Propia.

7. Informe de Sostenibilidad

La encuesta realizada por EDINSOST me ha servido para valorar el conocimiento actual en relación a la importancia de las TIC en la sostenibilidad.

He detectado una disparidad de conocimientos, a nivel teórico, entre la dimensión ambiental y el resto de dimensiones del proyecto. A lo largo de mis estudios y en la sociedad, he recibido bastante información respecto al impacto que tiene el desarrollo de un producto físico en el medio ambiente. Esto me ha ayudado a ser consciente de la importancia de una buena planificación y estudio sobre la sostenibilidad del mismo, sin embargo, también me ha ayudado a detectar el poco valor que se le da a la dimensión económica y, sobre todo la social por parte de la sociedad.

Lamentablemente, considero que no dispongo del conocimiento práctico suficiente para realizar un estudio de sostenibilidad, con garantías, debido a la poca experiencia laboral de la que dispongo y el poco conocimiento que tengo de los indicadores necesarios para medir el impacto del proyecto. Soy consciente de esta deficiencia, por eso considero que tendré que dedicarle especial dedicación a esta sección para compensar la falta de conocimientos.

En resumen, dispongo de un conocimiento mínimo en lo que respecta a la sostenibilidad, sin embargo, soy consciente de la importancia del mismo y considero que este proyecto puede servirme de guía para profundizar en este campo, sobre todo a nivel social y económico.

7.1 Dimensión ambiental

- **¿Has estimado el impacto ambiental que tendrá la realización del proyecto?**
Aunque no se haya realizado un estudio del impacto ambiental del proyecto, algunas decisiones han sido tomadas pensando en ello.
Para ahorrar en costes y reducir el impacto ambiental que puede suponer el uso de una oficina al completo, se ha decidido optar por el alquiler de un espacio compartido reduciendo el impacto producido por el consumo eléctrico.
Respecto a los demás recursos, se ha intentado minimizar el número de recursos físicos para reducir cualquier impacto ambiental derivado.
- **¿Te has planteado minimizar el impacto, por ejemplo, reutilizando recursos?**
Al ser un proyecto informático, la mayoría de los recursos utilizados son digitales que tienen un impacto reducido y, en nuestro caso, solo trabajaremos con un ordenador y un servidor.
Se ha decidido no reciclar el portátil como servidor, para el desarrollo, ya que aggitly dispone de este hardware y no veo necesario utilizar recursos para preparar el ordenador portátil con todo lo necesario para actuar como servidor.
- **¿Cómo se resuelve actualmente el problema que quieres abordar (estado del arte)?**

Al ser un proyecto desarrollado a medida para aggity, no existe ningún servicio que abarque todo lo que abarca esta solución, sin embargo, los clientes disponen de diversos servicios que simulan el mismo funcionamiento.

Los servicios actuales de aggity, por sí solos, son funcionales y no dependen entre ellos. Respecto a la comunicación de cualquier incidencia en el producto, el servicio que utilizan varía en función de cada cliente pero, en mi experiencia laboral, he detectado que la mayoría usa un proceso manual mediante el uso de correo electrónico.

- **¿En que mejorará ambientalmente tu solución a las existentes?**

Esta solución pretende unificar diversas herramientas reduciendo el tiempo que se les dedica por separado, sin embargo, esto no implica una mejora en el impacto ambiental ya que, al ser una actividad laboral, el tiempo ahorrado será empleado en otro proyecto manteniendo cualquier consumo que tenga un impacto negativo.

Para el correcto funcionamiento de la solución, será necesario seguir disponiendo de las herramientas actuales, por ello, este proyecto no supone ninguna mejora ambiental respecto a las soluciones actuales.

- **Si hubieras realizado el proyecto de nuevo, ¿Podrías haber usado menos recursos?**

Los recursos que se han usado han sido los mínimos necesarios para sacar adelante el proyecto. Así que, no creo que hubiera cambiado los recursos usados.

- **¿Qué recursos estimas que se usarán durante la vida útil del proyecto? ¿Cuál será el impacto de ese recurso?**

Complicado calcular los recursos que dedicar aggity en la vida útil del proyecto. El coste de mantener un software en funcionamiento es mínimo en cuanto a recursos.

7.2 Dimensión Económica

- **¿Has estimado el coste de la realización del proyecto (recursos humanos y materiales)?**

Antes de empezar el desarrollo, se ha realizado una planificación para reconocer los recursos necesarios para la realización del proyecto. Todo esto, junto a datos recopilados en Internet respecto a salarios y costes, ha ayudado a realizar una estimación de costes completa.

- **¿En que mejorará económicamente tu solución a la existente?**

A nivel económico, este producto supone un ahorro en el tiempo empleado por los clientes a la hora de comunicarse con los clientes finales, ya que automatiza cualquier comunicación referente a incidencias ocurridas en el desarrollo del producto.

Actualmente no disponen de ningún servicio similar que ofrecer a los clientes finales, por lo tanto, este software sería una inversión a largo plazo.

- **¿Es el coste planificado similar al coste final?, ¿Has justificado las diferencias?**

El coste final es menor al coste planificado debido a la decisión de prescindir de algunas herramientas debido a factores externos. En caso de no haberlo hecho, el coste final habría sido muy similar al planificado. Por esa parte, el cálculo se hizo correctamente.

- **¿Has considerado el coste de adaptaciones/actualizaciones/reparaciones durante la vida útil del proyecto?**
No se ha considerado ningún coste fuera del desarrollo del proyecto.

7.3 Dimensión Social

- **¿Qué crees que te va a aportar a nivel personal la realización de este proyecto?**
Este proyecto me ayudará a ganar experiencia en la gestión de proyectos y afianzar los conocimientos adquiridos a lo largo de la carrera. También me ha ayudado a detectar los conocimientos, de los cuales carezco, referente a la sostenibilidad.
A nivel práctico, me ayudará a comprender las fases por las que pasa un proyecto desde su planteamiento hasta su puesta en producción.
- **¿En que mejorará socialmente (calidad de vida) tu solución a las existentes?**
Como he explicado anteriormente, uno de los objetivos de este proyecto es automatizar una serie de procesos para facilitar la actividad laboral del cliente. Gracias a esta solución, se automatizará la detección y comunicación de incidencias por parte del cliente, reduciendo cualquier error humano a la hora de comunicarlo y reduciendo la carga laboral del mismo.
También dispondrá de datos en tiempo real de los distintos servicios de aggity. lo que facilitará su visualización y el estudio de los mismo por parte del cliente.
- **¿Existe una necesidad real del proyecto?**
Aunque actualmente considere que no existe una necesidad real del proyecto, el mismo se desarrolla con la intención de incrementar el valor de la marca aggity en un mercado que se está empezando a saturar de soluciones para Industria 4.0
- **¿El producto puede crear algún tipo de dependencia ponga a los usuarios en una posición difícil?**
El producto no realiza ninguna acción que actualmente no sea posible. La función principal de la solución es centralizar todas esas acciones en un único lugar.

8. Arquitectura del sistema

8.1 Visión global del proyecto

La solicitud propuesta consiste de dos aplicaciones apoyadas por una base de datos. En la figura 5 se puede observar la relación entre ambos aplicativos

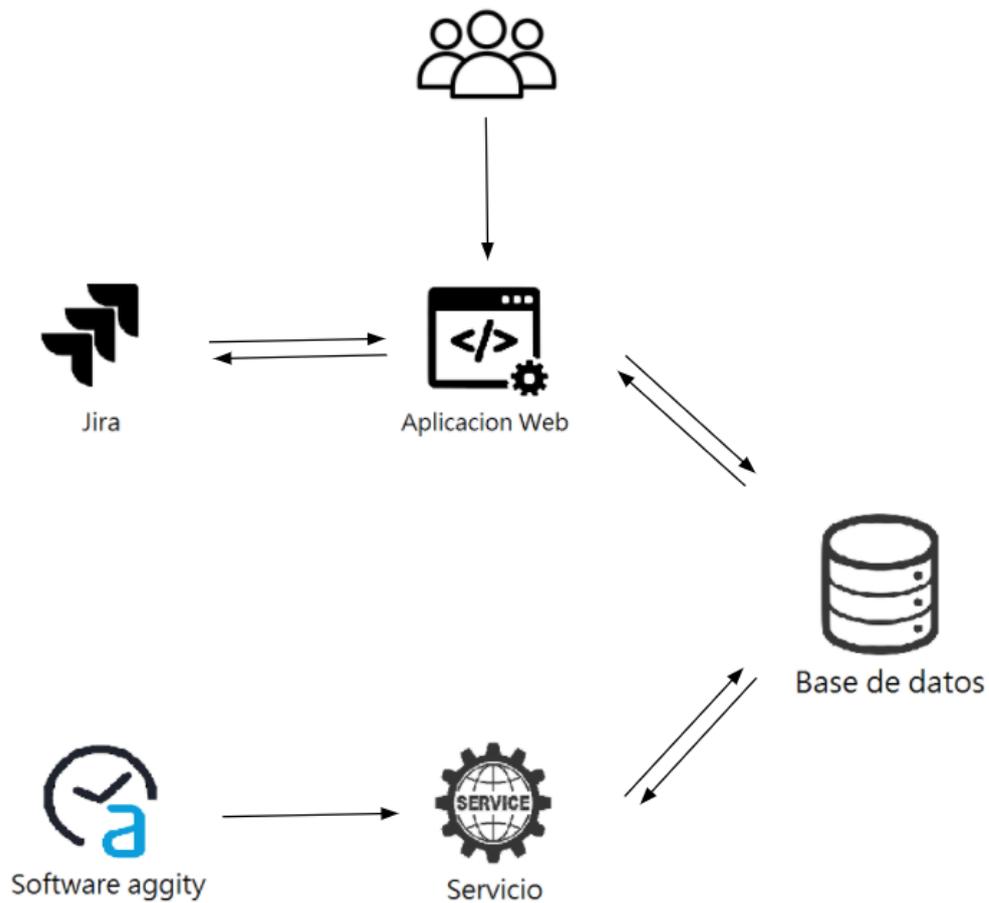


Figura 5. Diagrama de relación entre las soluciones del proyecto. Elaboración propia

La primera solución consiste en un portal web donde los usuarios pueden observar el estado de los servicios del software de aggity, iniciarlos si no estuvieran activos e incluso reiniciarlos, si han detectado alguna incidencia. También se ha incluido la función de comunicar cualquier incidencia encontrada, en el software, a través de Jira que es una herramienta de seguimiento de errores en incidencia.

Hemos utilizado el framework ASP.NET que nos permite crear una interfaz de usuario interactiva. ASP.NET es un framework desarrollado por Microsoft, compatible con diversos lenguajes de programación, en nuestro caso usaremos C# para todo el backend de la solución.

También se ha desarrollado un servicio de windows encargado de controlar el estado del software de aggity y comunicárselo a la base de datos. Se ha desarrollado en C# [19] junto al framework .NET [20].

Por último, se ha utilizado el sistema de gestión de bases de datos Microsoft SQL Server, al cual llamaremos MSSQL a partir de este punto, para manipular los datos utilizados en ambas aplicaciones.

Todas estas herramientas son muy conocidas y se están actualizando constantemente por lo que hemos podido encontrar diversas páginas con documentación.

Más adelante profundizaremos en cada una de ellas y su implementación.

8.2 Diseño de Bases de datos

Al principio, la base de datos solo estaba pensada para almacenar las credenciales de los distintos usuarios, así como la relación entre una empresa y su proyecto en la herramienta de Jira, sin embargo, también se ha utilizado como medio de comunicación entre la aplicación web y el servicio desarrollado.

A continuación listamos las tablas utilizadas y su utilidad dentro de la solución:

- **AspNetCompany**: En esta tabla se registra el dato de cada empresa y la URL del proyecto en Jira.
- **AspNetUser**: Tabla encargada de guardar los datos de los usuarios registrados por la aplicación web. El password se encripta con un algoritmo de hash seguro de 256 bits antes de guardarse.
- **AspNetCompanyUser**: Tabla que relaciona cada usuario del portal con la empresa.
- **AspNetServiceStatus**: Tabla utilizada por ambas aplicaciones para informar del estado de cada servicio de aggity y reiniciarlos si fuera necesario.

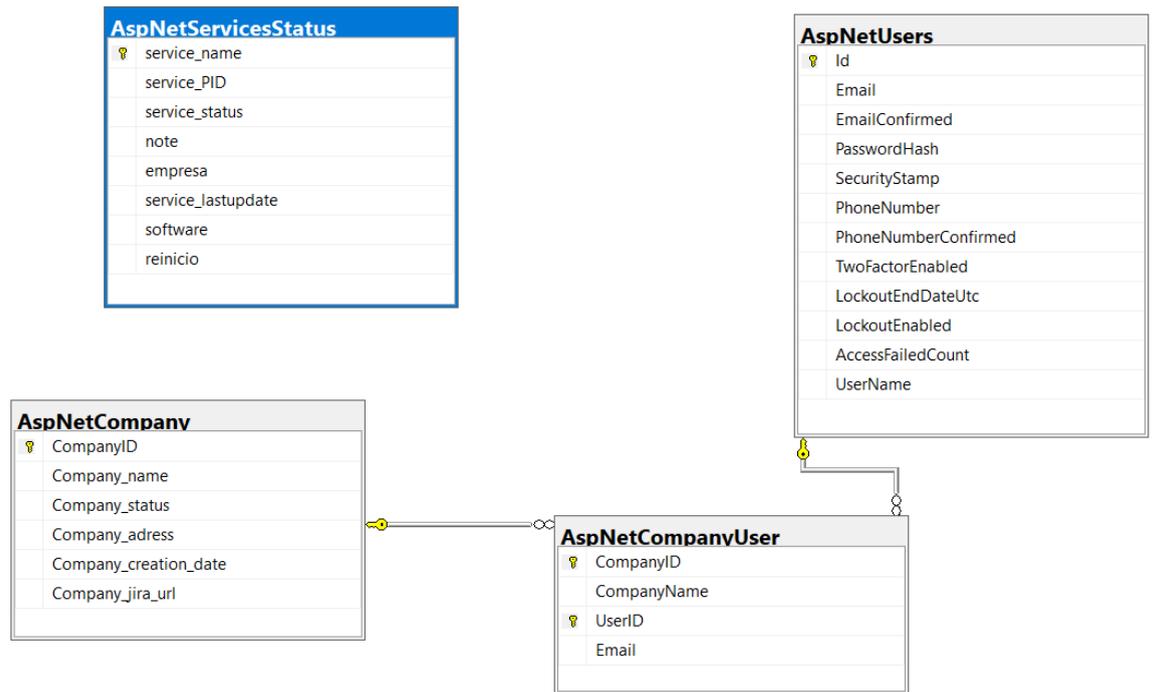


Figura 6. Diagrama de entidad-relación de la base de datos utilizada. Elaboración Propia

8.3 Diseño interfaz aplicación web

Para poder navegar por el portal, el usuario dispone de una barra de navegación superior desde la cual podrá acceder a las diversas páginas de la aplicación. Esta barra siempre estará disponible, independientemente de la página en la que nos encontremos, lo que facilita la navegación al ser bastante intuitiva

8.3.1 Página principal

La página principal dispone de dos modos de visualización:

- En la figura 7 podemos observar una página principal donde ningún usuario ha iniciado sesión. Esta vista tiene enlaces a páginas que no requieren credenciales y son informativas.
- En cambio, si hemos realizado alguna conexión, refrescamos la vista para añadir todas las funcionalidades que necesitas alguna validación

En ambas se puede observar información de aggity, software que ofrece y un enlace a la web oficial para poder obtener más información.



Figura 7. Vista de la página principal sin inicio de sesión.

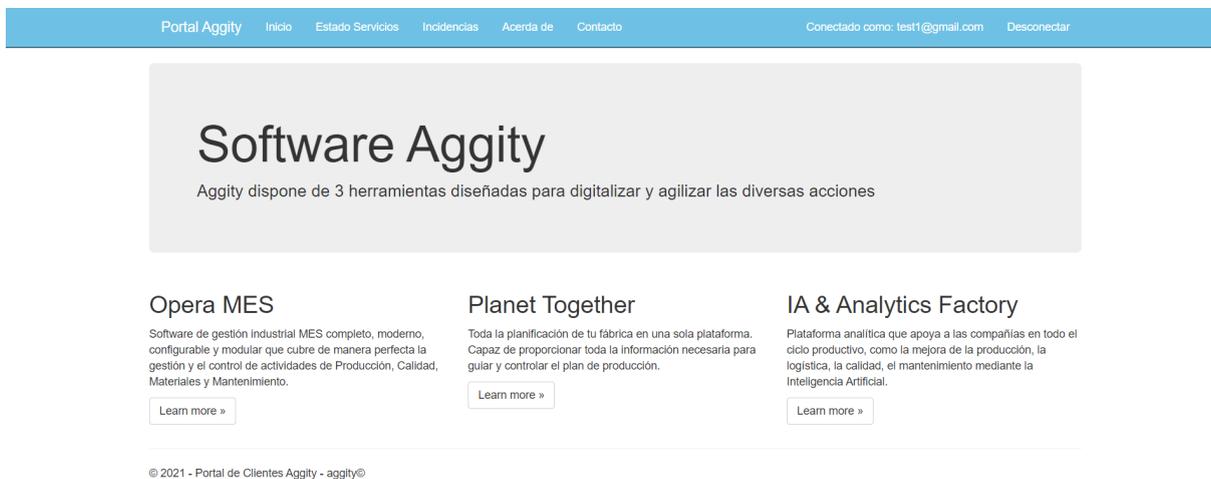


Figura 8. Vista de la página principal con inicio de sesión.

8.3.2 Página registro

En esta página, un usuario podrá darse de alta en el sistema, siempre que cumpla las restricciones de complejidad en la contraseña. En caso de realizarse correctamente, se le redirigirá a la página principal con el usuario iniciado.

Portal Aggity Inicio Acerca de Contacto Registrarse Iniciar sesión

Registrarse.

Crear un nuevo usuario.
Los campos obligatorios estan marcados con un asterisco (*)

Usuario*

Contraseña*

Confirmar contraseña*

Telefono

© 2021 - Portal de Clientes Aggity - aggity®

Figura 9. Vista de la página de registro..

8.3.3 Página Login

Página encargada de validar las credenciales en la base de datos. Al igual que la página de registro, se redirigirá a la página principal si la validación tienen éxito.

Portal Aggity Inicio Acerca de Contacto Registrarse Iniciar sesión

Iniciar sesión.

Utilice una cuenta local para iniciar sesión.

Correo electrónico

Contraseña

¿Recordar cuenta?

[Registrarse como usuario nuevo](#)

© 2021 - Portal de Clientes Aggity - aggity®

Figura 10. Vista de la página de inicio de sesión.

8.3.4 Páginas informativas

Las páginas Acerca de y Contacto son páginas de carácter informativo y no contienen ninguna funcionalidad relacionada con el proyecto.

8.3.5 Página Estado Servicios (Solo conectado)

Página encargada de mostrar la lista de soluciones que ofrece aggity y el estado de los servicios que tengas contratados en función de la empresa del usuario conectado.

Cada elemento de la lista dispone de un botón para iniciar o reiniciar, en caso de que ya estuviera iniciado, el servicio de aggity correspondiente. Detallaremos el funcionamiento más adelante.

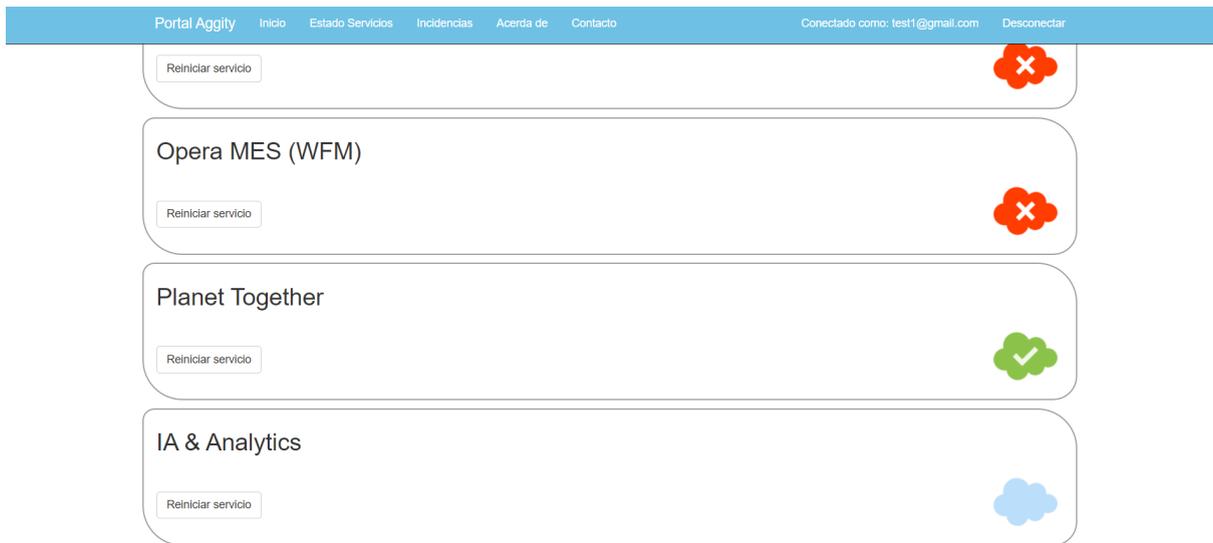


Figura 11. Vista de la página de estado de los servicios..

8.3.6 Página Incidencias (Solo conectado)

Página desde la cual se puede reportar una incidencia. Dicha incidencia será recogida en Jira y asignada al proyecto de la empresa del usuario. El nombre y correo electrónico son autocompletados con los datos del usuario conectado.

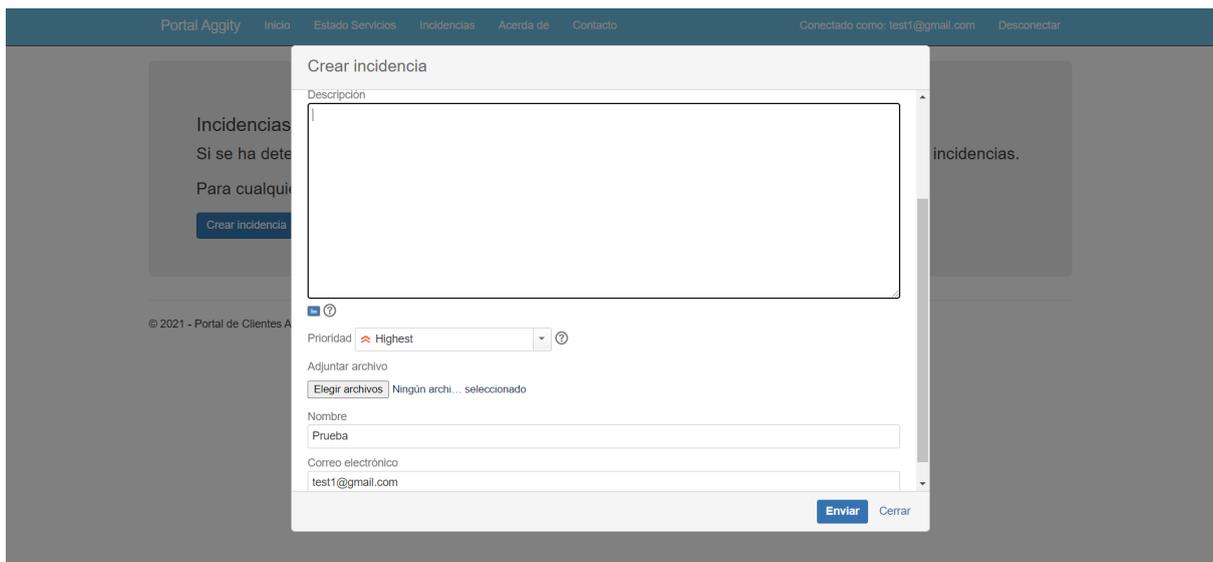


Figura 12. Vista de la página de incidencias..

8.3.7 Pagina configuracion usuario (solo conectado)

Página donde se pueden modificar datos del usuario como el teléfono móvil, el password o la empresa. Cada uno de los enlaces, te redirige a páginas con formularios para cambiar cada dato.



Figura 13. Vista de la página de administración de cuenta.

8.3.8 Páginas contraseña

Página que contiene un formulario simple para cambiar la contraseña. Antes de cambiarlo, será necesario validar la contraseña y el usuario actual.



Figura 14. Vista de la página de edición de contraseña.

8.3.9 Página empresa

Página que permite escoger empresas, las cuales están listadas en una caja de selección.

8.3.10 Página teléfono

Página simple para modificar el número de teléfono en la base de datos.

9. Implementación

9.1 Tecnologías y lenguajes utilizados

ASP.NET es un framework de desarrollo para aplicaciones web dinámicas desarrollado por Microsoft. Este framework es compatible con diversos lenguajes

como Visual Basic y F#, en nuestro casos, se ha decidido utilizar C# por la experiencia obtenida en aggity.

Para el desarrollo, se ha decidido utilizar Visual Studio debido a la compatibilidad con todas las herramientas incluidas en el proyecto así como la facilidad, a la hora de depurar la aplicación, gracias a IIS Express.

IIS Express (Internet Information Server Express) es una herramienta que permite convertir una máquina en un servidor web donde poder publicar la web desarrollada. Este servicio puede ser instalado en los sistemas operativos de Windows []

Respecto a la base de datos, se ha optado por utilizar MSSQL (MicroSoft SQL). Como se deduce del nombre, MSSQL es una sistema de gestión de bases de datos relacional desarrollado por Windows. Todas las aplicaciones de aggity hacen uso de este sistema lo que facilita su implementación en el entorno del cliente.

Destacar que, en todo el proyecto, se han ejecutado consultas parametrizadas para evitar la inyección de SQL junto a ejecuciones de procedimientos almacenados en la base de datos.

9.2 Implementación de la aplicación web

En esta aplicación hacemos uso del patron MVC (Modelo-Vista-Controlador) que es una arquitectura de software que consiste en separar la lógica de negocio de la parte visual de la aplicación, utilizando un componente controlador, que actúa de intermediario.

9.2.1 Conexión a base de datos

Se ha creado una clase llamada **CAccesoBD** donde declaramos todas las funciones encargadas de realizar las consultas a la base de datos. Aparte de los parámetros necesarios para su funcionamiento, se ha incluido una variable de tipo string, la cual pasamos por referencia y registra cualquier error que pueda producirse.

Otro dato importante es el uso de la sentencia using a la hora de declarar un objeto de tipo SqlConnection como, por ejemplo, en la Figura 15 . La sentencia using provoca que el objeto llame a la función **Dispose** una vez finalizado el bloque, esto provoca que el objeto sea liberado de la memoria de manera explícita y no provoque ningún bloqueo al realizar otra conexión desde la misma instancia ya que diversas funcionalidades trabajan de forma asíncrona.

```

using (SqlConnection con = new SqlConnection(ConfigurationManager.ConnectionStrings["DefaultConnection"].ConnectionString))
{
    con.Open();
    SqlCommand comando = con.CreateCommand();
    comando.CommandText = Aggity_GuardarEmpresa;
    comando.CommandType = CommandType.StoredProcedure;

    comando.Parameters.AddWithValue("@userId", userID);
    comando.Parameters.AddWithValue("@empresa", empresa);

    comando.ExecuteNonQuery();
    comando.Parameters.Clear();
    result = true;
}

```

Figura 15. Ejemplo de uso de la sentencia using.

Para poder conectarnos, la cadena de conexión la recuperamos del archivo de configuración del proyecto. Este archivo es editable desde fuera para facilitar cambios posteriores sin necesidad de recompilar la aplicación.

```

<configuration>
  <configSections>...</configSections>
  <connectionStrings>
    <!--Cadena de conexión-->
    <add name="DefaultConnection" connectionString="Data Source=GTTLAP003;Initial Catalog=PortalAggity;Integrated Security=True;"
        providerName="System.Data.SqlClient" />
  </connectionStrings>
  <system.web>...</system.web>
  <system.webServer>...</system.webServer>
  <runtime>...</runtime>
  <entityFramework>...</entityFramework>
  <system.codedom>...</system.codedom>
</configuration>

```

Figura 16. Ejemplo de cadena de conexión del archivo de configuración.

A continuación están listadas las funciones que incorporan consultas a la base de datos.

- **ObtenerEstado**(*string userID*): Devuelve una lista de tuplas con el nombre de un servicio de aggity y su estado.
- **MarcarReinicioServicio**(*string software, string empresa*): Marca, en la base de datos, el servicio del software de la empresa para su reinicio
- **ExisteEmpresa**(*string userID*): Devuelve una tupla con la empresa y un booleano true, o un string vacío junto a un boolean false en caso de que no tenga asignada ninguna empresa.
- **ListaEmpresas**(): Devuelve una lista de string con los nombres de las empresas.
- **GuardarEmpresa**(*string userID, string empresa*): Asigna la empresa al usuario con userID.
- **EliminarEmpresa**(*string userID*): Elimina la empresa del usuario userID.
- **ObtenerURL**(*userID*): Devuelve la URL utilizada para imputar incidencias en función de la empresa del usuario userID.

```

internal List<Tuple<string, string>> ObtenerEstados(string userId, ref string msgError)...
internal bool MarcarReinicioServicio(string userId, string software, ref string msgError)...
internal Tuple<bool, string> ExisteEmpresa(string userID, ref string msgError)...
internal List<string> ListaEmpresas(ref string msgError)...
internal bool GuardarEmpresa(string userID, string empresa, ref string msgError)...
internal bool EliminarEmpresa(string userID, ref string msgError)...
internal string ObtenerURL(string userID, ref string msgError)...

```

Figura 17. Lista de funciones usadas en la aplicación web.

Como se puede observar, ninguna de las funciones anteriores hacen referencia a la creación o actualización de un usuario. Esto es debido a que reutilizamos las herramientas que nos ofrece ASP.NET para administrar los usuarios. ASP.NET utiliza la clase Identity que ofrece, tanto herramientas de validación en línea como sistemas para controlar sesiones activas.

Para que todo funcione correctamente, se tuvo que modificar el archivo de configuración app.config con una cadena de string con los datos de conexión a la base de datos correcta.

9.3 Implementación servicio

Para esta solución, se creó un servicio de Windows con el framework .NET programado en C#. Se ha reutilizado la clase **CAccesoBD** de la aplicación anterior con las siguientes funciones:

- **ActualizarEstadosServicios**(*string empresa, string nombre, string software, int estado, string nota, int PID*): Inserta o actualiza los datos y el estado del servicio del software de aggity en empresa.
- **ComprobarReinicioServicio**(*string empresa*): Comprueba los servicios de una empresa que han sido marcados para reiniciar por la aplicación web y los añade a una lista para devolverlos.

La cadena de conexión se guarda de la misma manera que la aplicación anterior, en un documento de configuración [Figura]. En este caso también hemos incluido variables que representan los nombres de los servicios, cada cuanto tiempo se realiza la comprobación de servicios entre otras cosas. Des esta manera, el servicio es más susceptible a cambios ante cualquier incidencia.

```

<appSettings>
  <!-- Base de datos -->
  <add key="stringConectConfig" value="Data Source=GTTLAP003;Initial Catalog=PortalAggity;Persist Security Info=True;User ID=operasa;
  <!--Nombre de la empresa del cliente que tiene contratado el portal-->
  <add key="Empresa" value="test" />

  <!--Nombre del servicio de OperaMes Workflow Manager-->
  <add key="ServicioOperaMESWFM" value="SQLSERVERAGENT" />

  <!--Nombre del servicio de OperaMes Plant Manager-->
  <add key="ServicioOperaMESPM" value="ALG" />

  <!--Nombre del servicio de Planet Together-->
  <add key="ServicioPlanetTogether" value="Appinfo" />

  <!--Nombre del servicio de IA-->
  <add key="ServicioIA" value="test" />
  <!-- tiempo consulta timer, cada 5 segundos, esta expresado en milisegundos -->
  <add key="TiempoComprobacion" value="5000" />

  <add key="ClientSettingsProvider.ServiceUri" value="" />
</appSettings>

```

Figura 18. Variables en archivo de configuración del servicio.

A la hora de programar un servicio, heredamos las funciones OnStart y OnStop de la clase ServiceBase. Como su nombre indica, la función OnStart se ejecuta al iniciar el servicio y OnStop al finalizar. Hay diversas maneras de incorporar funcionalidad al servicio, sin embargo, hay que tener en cuenta que es recomendable que el funcionamiento sea asíncrono para evitar cualquier bloqueo relacionado al rendimiento, en este caso se ha escogido el uso de los Timer.

Un Timer es una clase que genera un evento pasado un tiempo determinado. Estos eventos pueden ser recurrentes y no se bloquean entre ellos ya que trabajan en distintos hilos de ejecución.

El primer paso es configurar las propiedades del timer [18] al iniciar el servicio. Configuramos el tiempo que viene determinado en el archivo de configuración app.config, Después declaramos el evento que se ha de ejecutar junto a las propiedades de AutoResert, para que el Timer vuelva a ejecutarse, y la propiedad Enabled, que lo activa.

```

TiempoComprobacion = Convert.ToInt32(System.Configuration.ConfigurationManager.AppSettings["TiempoComprobacion"]);
statusTimer = new System.Timers.Timer(TiempoComprobacion);
statusTimer.Elapsed += OnTimedEventStatus;
statusTimer.AutoReset = true;
statusTimer.Enabled = true;

```

Figura 19. Ejemplo de uso de la clase timer.

En nuestro caso activaremos 2 timer:

- Uno encargado de leer el estado de los servicios de la máquina para actualizarlos en la base de datos.
- Otro encargado de buscar los servicios que han sido marcados para reiniciar en la base de datos y reiniciarlos.

Para poder interactuar con los servicios, hacemos uso de la clase ServiceController que, dado el nombre de un servicio, crea una instancia para

poder manipularlo. Esta clase dispone de las funcionalidades para saber el estado de un servicio y reiniciarlo.

Para instalar el servicio en el sistema de windows, hemos hecho uso de la aplicación installutil.exe que es instalado con el framework .NET. Es una herramienta que nos permite registrar servicios de ensamblados específicos. En este caso, al haber utilizado Visual Studio y frameworks de Microsoft, no hemos tenido ningún problema en este punto como se puede observar en la figura 20.

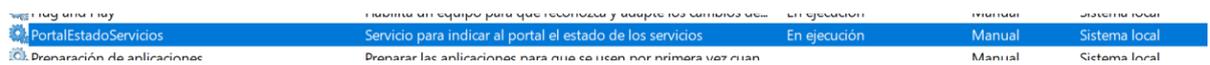


Figura 20. Servicio en línea en Windows.

10. Cambios a futuro

Como se comentó al principio del proyecto, existen diversas empresas que disponen de una solución similar con funcionalidades muy superiores a las realizadas en este proyecto. Todo esto es comprensible debido a los pocos recursos de los que ha dispuesto el proyecto y a la poca experiencia con las herramientas.

A medida que avanzaba el proyecto, se comenzaban a apreciar necesidades inherentes de las funcionalidades que estábamos implementando. Un ejemplo de ello sería implementar funcionalidades de creación y modificación de empresas en el mismo portal. También sería interesante añadir roles de usuario dentro de una empresa y permisos para editar los trabajadores de una empresa.

La mayoría de aspectos visuales de la aplicación web son placeholders, es decir, no acabarán en el producto final ya que otro departamento se encargará de maquetarlo para que represente, de manera correcta, a la marca aggity.

11. Conclusión

Este proyecto tenía 2 objetivos principales a desarrollar junto a uno secundario pero necesario para el correcto funcionamiento de la web: Mostrar el estado de los servicios de las herramientas de aggity y poder reiniciarlos, implementar la herramienta Jira en nuestro portal y un sistema de gestión de usuarios para darle sentido al funcionamiento.

A pesar de los cambios realizados, respecto a la planificación inicial debido a errores realizados en la fase de análisis, se ha realizado correctamente los objetivos menos la parte de Jira. Al principio, se tenía pensado poder ingresar incidencias desde el portal y mostrar una lista de incidencias abiertas al cliente, Esto último no se ha llegado a completar por complicaciones al configurar el proyecto en Jira

Era un proyecto ambicioso donde la falta de experiencia ha generado diferencias entre el producto real y el planificado. Aunque funcionalmente, se hayan conseguido la mayoría de propuestas que se hicieron al principio, en el estado actual dista de ser un producto funcional que se pueda vender a los clientes.

Este margen tan amplio de mejora me permitirá estudiar mejor las herramientas utilizadas y pulir el sistema más allá de los límites planteados en este proyecto.

12. Bibliografía

1. Transformación Digital. aggity, 2020:
<https://aggity.com/transformacion-digital-la-importancia-de-un-partner-tecnologico/>
2. La industria 4.0. aggity, 2020:
<https://aggity.com/evolucion-sistemas-mes/>
3. Acerca de SAP España. SAP, 2021:
<https://www.sap.com/spain/about.html>
4. SAP Manufacturing Execution. SAP, 2021:
<https://www.sap.com/spain/industries/mill-products.html?pdf-asset=d498fa1e-427d-0010-87a3-c30de2ffd8ff&page=1>
5. Sistema de Tickets. Carlos Villalobos, 1 de Octubre de 2020:
<https://blog.hubspot.es/service/sistema-tickets>
6. Sistema de Ticket Jira. Atlassian, 2021:
<https://www.atlassian.com/es/software/jira>
7. Sistema de Ticket Request Tracker. Best Practical, 2020:
<https://bestpractical.com/request-tracker>
8. ¿Que es un sprint?. Abraham Requena Mesa, 19 de Diciembre 2018:
<https://openwebinars.net/blog/que-es-un-sprint-scrum/>
9. ¿Qué son las metodologías ágiles?. IEBS, 2019:
<https://www.iebschool.com/blog/que-son-metodologias-agiles-agile-scrum/>
10. Treball de Fí de Grau. FIB, 2021:
<https://www.fib.upc.edu/ca/estudis/graus/grau-en-enginyeria-informatica/treball-de-fi-de-grau>
11. APS.NET . Microsoft, 2021:
<https://dotnet.microsoft.com/apps/aspnet>

12. SourceTree. Atlassian, 2021:
<https://www.sourcetreeapp.com/>
13. LinkedIn Salary. LinkedIn, 2021:
<https://es.linkedin.com/salary/>
14. Boletín oficial del estado, Ministerio de Trabajo. Migraciones y Seguridad Social, 2 de Febrero de 2019:
<https://www.boe.es/buscar/doc.php?id=BOE-A-2019-1366>
15. Plans for Request Tracking. Best Practical, 2020:
<https://bestpractical.com/get-started>
16. Licencias de Jira Software. Atlassian, 2021:
<https://www.atlassian.com/es/licensing/jira-software>
17. Tarifas MGCoworking. MGCoworking, 2021:
<https://www.mgcoworking.com/tarifas>
18. Agregar un Temporizador, 2021
<https://docs.microsoft.com/es-es/visualstudio/ide/step-6-add-a-timer?view=vs-2019>
19. Documentación de C#, 2021
<https://docs.microsoft.com/es-es/dotnet/csharp/>
20. Documentación de .NET, 2021
<https://docs.microsoft.com/es-es/dotnet/>
21. IIS Home, 2021
<https://www.iis.net/>
22. Stack Overflow, 2021
<https://es.stackoverflow.com/>