



Universidad de Valladolid

Escuela de Ingeniería Informática

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática

Mención en Computación

**Aplicación para la gestión de las horas
extra en un entorno empresarial
utilizando Microsoft Power Apps**

Autor:

Miguel Sacristán de Frutos

Tutor:

Jesús María Vegas Hernández

Agradecimientos

A mi tutor Jesús, por su guía y apoyo a lo largo del proceso que tanto me ha aportado y ayudado a mi desarrollo tanto personal como profesional.

A mi familia, en especial a mi abuela, mis padres y mi hermano, que me han acompañado y apoyado durante toda la vida, animándome a seguir adelante a pesar de las dificultades y sobreponerme a los retos que se presentaban.

A mis compañeros, sin los que esta etapa no hubiera sido lo mismo, no podría haber tenido un grupo mejor.

Por último a todos los amigos que han estado a mi lado todos estos años, gracias a todos de corazón.

Resumen

En muchas empresas, la gestión de las horas extra de los empleados puede ser un proceso complejo. El objetivo de este proyecto es desarrollar una herramienta que permita llevar un seguimiento de estas horas, desde la creación del registro hasta su aprobación en las distintas fases. La herramienta cuenta con una interfaz intuitiva y fácil de usar que permite realizar una gestión eficiente de las horas extra.

El desarrollo se ha llevado a cabo utilizando PowerApps, Sharepoint y Power Automate, herramientas integradas en el entorno de Microsoft 365 y siguiendo la metodología de desarrollo ágil Scrum.

Palabras clave

PowerApps, SharePoint, Power Automate, Scrum, Horas extra

Summary

In many companies the management of employee overtime can be a complex process. The aim of this project is to develop a tool to keep track of these hours, from the creation of the record to its approval at different stages. The tool has an intuitive and user-friendly interface that allows for efficient management of overtime.

The development was carried out using PowerApps, Sharepoint and Power Automate, tools integrated in the Microsoft 365 environment and following the Scrum agile development methodology.

Keywords

PowerApps, SharePoint, Power Automate, Scrum, Work overtime

Índice general

1. Introducción	1
1.1. Contexto	1
1.2. Motivación	1
1.3. Objetivos	2
1.4. Aplicaciones similares	3
1.4.1. Beebole	3
1.4.2. Clockify	4
1.4.3. TramitApp	5
1.5. Formación y referencias relevantes	6
1.6. Estructura de la memoria	7
2. Planificación del proyecto	9
2.1. Gestión del proyecto	9
2.2. Metodologías ágiles	9
2.3. Scrum	10
2.4. Planificación inicial del proyecto	12
2.5. Análisis de costes	13
2.6. Análisis de riesgos	14
3. Análisis	21
3.1. Actores y roles	21
3.2. Requisitos	22
3.2.1. Requisitos funcionales	23
3.2.2. Requisitos no funcionales	25
3.2.3. Requisitos de información	25
3.2.4. Reglas de negocio	26
3.3. Casos de uso	27
3.3.1. Diagrama de flujo de una incidencia	28
3.3.2. Diagrama de casos de uso	29
3.3.3. Descripción de los casos de uso	31
3.4. Diagramas de actividad	41
3.5. Modelo de dominio	45

4. Diseño	47
4.1. Patrones de diseño en PowerApps	48
4.2. Diseño de la base de datos	49
4.3. Diseño de la interfaz de usuario	51
4.3.1. Atributos de usabilidad	51
4.3.2. Elaboración de bocetos	52
5. Implementación	59
5.1. Tecnologías empleadas	59
5.1.1. Microsoft PowerApps	59
5.1.2. Microsoft SharePoint	62
5.1.3. Microsoft Power Automate	64
5.2. Control de versiones	70
5.3. Seguridad en el entorno Microsoft	70
6. Pruebas	73
6.1. Pruebas	73
6.1.1. Pruebas en PowerApps	73
7. Despligue	79
7.1. Despligue	79
7.1.1. Permisos de SharePoint	80
7.1.2. Permisos de Power Automate	80
7.1.3. Publicación	81
7.1.4. Control de versiones	82
8. Seguimiento del proyecto	83
8.0.1. Sprint 0	83
8.0.2. Sprint 1	84
8.0.3. Sprint 2	85
8.0.4. Sprint 3	86
8.0.5. Sprint 4	87
8.0.6. Sprint 5	88
8.0.7. Sprint 6	90
9. Conclusiones y trabajo futuro	91
9.1. Conclusiones	91
9.2. Trabajo futuro	92
Bibliografía	97
Apéndices	105
A. Manual de usuario	105

ÍNDICE GENERAL

A.1. Pantalla de Carga	105
A.2. Pantalla de Inicio	106
A.3. Pantalla People	106
A.3.1. Envío registro a pago	107
A.3.2. Rechazo de registro	108
A.3.3. Generar reporte People	109
A.4. Pantalla Manager	109
A.4.1. Aprobación de incidencia	110
A.4.2. Rechazo de incidencia	111
A.4.3. Cancelación de incidencia	112
A.4.4. Cambio de pantalla de Manager a Empleado	113
A.4.5. Generar reporte Manager	113
A.5. Pantalla Empleado	114
A.5.1. Nuevo registro	115
A.5.2. Detalles de un registro	118
A.5.3. Edición de un registro	119
A.5.4. Enviar un registro	120
A.5.5. Cancelación de un registro	121
A.5.6. Borrado de un registro	122
B. Código utilizado en la aplicación	125

Índice de figuras

1.1. Introducir registro y solicitar aprobación	4
1.2. Aplicación Clockify	5
1.3. Solicitar aprobación en Clockify	5
1.4. Aplicación Tramatapp	6
2.1. Matriz de riesgos; tomada de [16].	15
3.1. Diagrama de flujo de una incidencia	29
3.2. Diagrama de casos de uso	30
3.3. Diagrama de actividad de identificarse	42
3.4. Diagrama de actividad de crear incidencia	43
3.5. Diagrama de actividad de eliminar una incidencia	44
3.6. Diagrama de actividad de generar reporte Manager	44
3.7. Diagrama de actividad de aprobar o rechazar una incidencia Manager	45
3.8. Modelo de dominio	46
4.1. Patrón Aprobación; tomada de [26]	48
4.2. Patrón Inspección/Auditoría; tomada de [27]	49
4.3. Modelo relacional	50
4.4. Atributos de usabilidad; tomada de [31].	52
4.5. Boceto de la Pantalla de Login	53
4.6. Boceto de la Pantalla de People	54
4.7. Boceto Consultar Detalles	54
4.8. Boceto Enviar Incidencias	55
4.9. Boceto Rechazar Incidencias	55
4.10. Boceto de la Pantalla de Manager	56
4.11. Boceto de la Pantalla Generar Reporte Manager	56
4.12. Boceto de la Pantalla de Empleado	57
4.13. Boceto de la Pantalla de Crear/Editar Incidencia	58
5.1. Conector SharePoint para esperar por una aprobación	66
5.2. Conector SharePoint obtención de elementos	67
5.3. Conectores SharePoint para crear el archivo y obtener su ruta	67
5.4. Conector Outlook archivo adjunto	68
5.5. Conector SharePoint obtener elementos Manager	69

5.6. Conector de Outlook en Power Automate	69
7.1. Compartir aplicación	79
7.2. Permisos SharePoint	80
7.3. Publicar aplicación	81
7.4. Historial de versiones	82
A.1. Animación Pantalla de Carga	105
A.2. Notificación de error	106
A.3. Pantalla de Inicio	106
A.4. Pantalla de People	107
A.5. Botón enviar para pago	107
A.6. Envío para pago	108
A.7. Botón rechazar registro	108
A.8. Rechazar registro	109
A.9. Botón generar reporte People	109
A.10.Pantalla Manager	110
A.11.Botón aprobar incidencia	110
A.12.Aprobación incidencia	111
A.13.Rechazo de una incidencia	112
A.14.Botón Cancelar para confirmar	112
A.15.Confirmar cancelacion	113
A.16.Rechazo de una incidencia	113
A.17.Botón generar reporte Manager	113
A.18.Formulario de reporte	114
A.19.Pantalla Empleado	115
A.20.Filtros de la Pantalla de Empleado	115
A.21.Nuevo registro	116
A.22.Formulario de Guardia	117
A.23.Formulario de Intervenciones	117
A.24.Formulario de Trabajo programado	118
A.25.Botón de detalles	118
A.26.Detalles de un registro	119
A.27.Editar registro	120
A.28.Botón Enviar para confirmar	120
A.29.Confirmar envío	121
A.30.Botón Cancelar para confirmar	121
A.31.Confirmar cancelacion	122
A.32.Botón Borrar para confirmar	122
A.33.Confirmar borrado	123

Índice de tablas

2.1. Planificación del proyecto en sprints	13
2.2. Costes finales del proyecto	14
2.3. Riesgo 1: Fallo en la planificación	15
2.4. Riesgo 2: Falta de conocimiento	16
2.5. Riesgo 3: Responsabilidades adicionales	16
2.6. Riesgo 4: Ausencia	16
2.7. Riesgo 5: Fallo de disponibilidad tecnológico	17
2.8. Riesgo 6: Fallo en el equipo informático	17
2.9. Riesgo 7: Pérdida del proyecto.	17
2.10. Riesgo 8: Incumplimiento de objetivos	18
2.11. Riesgo 9: Requisitos cambiantes o mal definidos	18
2.12. Riesgo 10: Cambio en las tecnologías	18
2.13. Riesgo 11: Falta de adaptación a la aplicación	19
2.14. Riesgo 12: Limitación de tiempo	19
2.15. Riesgo 13: Inexperiencia trabajando con SCRUM	19
3.1. Caso de Uso 01: Identificarse	31
3.2. Caso de Uso 02: Consultar detalles incidencia.	32
3.3. Caso de Uso 03: Crear Guardia	33
3.4. Caso de Uso 04: Crear trabajo programado	34
3.5. Caso de Uso 05: Editar guardia	35
3.6. Caso de Uso 06: Enviar incidencias al Manager	36
3.7. Caso de Uso 07: Eliminar incidencia	37
3.8. Caso de Uso 08: Cancelar incidencia	38
3.9. Caso de Uso 09: Aprobar incidencia Manager	38
3.10. Caso de Uso 10: Cancelar incidencia Manager	39
3.11. Caso de Uso 11: Generar reporte Manager	40
3.12. Caso de Uso 12: Generar reporte People	40
3.13. Caso de Uso 13: Enviar incidencia para pago	41
6.1. Prueba 1: Crear Trabajo programado	74
6.2. Prueba 2: Crear Guardia	75
6.3. Prueba 3: Eliminar registros	75
6.4. Prueba 4: Enviar incidencias al Manager	76

6.5. Prueba 5: Cancelar incidencias al Manager	76
6.6. Prueba 6: Rechazar incidencias	77
6.7. Prueba 7: Enviar para pago incidencias People	77
6.8. Prueba 8: Generar reporte Manager	78
6.9. Prueba 9: Inicio de sesión	78
8.1. Sprint 1 Semana 1	84
8.2. Sprint 1 Semana 2	84
8.3. Sprint 2 Semana 3	85
8.4. Sprint 2 Semana 4	85
8.5. Sprint 3 Semana 5	86
8.6. Sprint 3 Semana 6	86
8.7. Sprint 4 Semana 7	87
8.8. Sprint 4 Semana 8	87
8.9. Sprint 5 Semana 9	88
8.10. Sprint 5 Semana 10	89
8.11. Sprint 6 Semana 11	90
8.12. Sprint 6 Semana 12	90

Capítulo 1

Introducción

1.1. Contexto

En muchas empresas, en especial aquellas que ofrecen servicio en un horario muy amplio o que cuentan con operaciones continuas, es común la realización de trabajos fuera del horario habitual, a través de horas extras, guardias o trabajos programados. Este tipo de trabajos deben ser gestionados adecuadamente para asegurar que el personal esté disponible, se cumpla la normativa laboral, exista un registro y se retribuya adecuadamente a los empleados.

Tradicionalmente, la solicitud, aprobación y gestión de horas extras y trabajos programados se lleva a cabo a través de métodos manuales internos de la empresa, o usando sistemas no centralizados y menos personalizados a las necesidades individuales de cada empresa lo que puede generar ineficiencias, descontento y pérdidas de información. Esto representa un reto, especialmente para departamentos de recursos humanos y supervisores, que necesitan optimizar el flujo de información para asegurar un seguimiento adecuado.

1.2. Motivación

La necesidad de una plataforma centralizada y eficiente para la gestión de estas horas extras y guardias surge de los entornos laborales actuales cada vez mas complejos. En muchas empresas estos sistemas manuales, o no centralizados dificultan la trazabilidad, lo que lleva a errores, demoras y en ocasiones tensiones entre los empleados y la dirección debido a procesos de aprobación y compensación poco claros.

Adicionalmente, la automatización, digitalización y centralización de estos procesos, puede reducir la carga administrativa y mejorar la eficiencia, proporcionando una solución transparente que beneficie tanto a los empleados como a los gestores.

Microsoft 365, principalmente a través de PowerApps, nos ofrece un entorno en el que poder diseñar

una solución personalizada para la gestión de este tipo de tareas, mejorando la eficiencia del proceso de solicitud, aprobación y gestión.

En mi caso y a través de una idea proporcionada por mis responsables en la empresa en la que realicé las prácticas, me pareció interesante desarrollar una aplicación interna personalizada, que se pudiera implementar en el entorno de la empresa para la monitorización eficiente de estas horas extras que realizaba el equipo, como guardias o trabajos fuera de horario (normalmente despliegues de procesos evolutivos), usando la tecnología con la que trabajábamos, Microsoft PowerApps. Esto resultaba especialmente relevante en un equipo como el mío, orientado al mantenimiento de aplicaciones, en el que este tipo de tareas ocurrían con frecuencia.

1.3. Objetivos

El objetivo principal es desarrollar una aplicación para la gestión centralizada de las horas realizadas fuera de horario utilizando Microsoft Power Apps, Sharepoint y Power Automate y que mejorara aplicaciones ya existentes, de forma que la solución se ajustase mejor a unas necesidades como las del equipo con el que trabajé.

Los objetivos del sistema son:

- **Facilitar la introducción de solicitudes:** permitir que los empleados puedan introducir las horas extras realizadas, que llamaremos trabajos programados, y reportar guardias, de forma ágil y sencilla, a través de cualquier dispositivo compatible con el entorno de Microsoft 365.
- **Automatizar el proceso de aprobación:** desarrollar un sistema de aprobación que notifique automáticamente tanto a los responsables como a los empleados del estado de las solicitudes, permitiendo una validación rápida, que reduzca el tiempo de espera y los cuellos de botella. También facilitar a los responsables de aprobación el proceso de añadir comentarios basados en sus respuestas previas, haciendo esta tarea menos repetitiva.
- **Mejorar la trazabilidad y transparencia:** asegurar un registro claro y accesible de todas las solicitudes en los diferentes estados del proceso, facilitando la consulta por parte del personal de recursos humanos o managers. Esto incluye almacenar fechas de inicio y fin, número de horas trabajadas, si se han tenido que realizar intervenciones, así como guardar comentarios relacionados con las horas extras. Esto también tiene como objetivo posibilitar la obtención de informes con los que poder trabajar fuera del sistema.
- **Uso de una interfaz intuitiva y eficiente:** diseñar una interfaz de usuario simple y fácil de navegar, que permita a los empleados y supervisores utilizar la aplicación sin complicaciones, minimizando el tiempo de formación y garantizando una experiencia de usuario óptima.
- **Optimizar el cálculo de compensaciones:** incluir en el sistema el cálculo automático de las compensaciones correspondientes a los trabajos programados y a las guardias, diferenciando por

nivel de cada empleado. También parametrizar estas remuneraciones de forma que su actualización a lo largo del tiempo sea sencilla.

- **Garantizar la seguridad e integridad de los datos:** aprovechar la robusta infraestructura del entorno de Microsoft, que asegure que los datos sensibles de los empleados estén protegidos mediante cifrado de datos, autenticación multifactor y otras regulaciones de privacidad y seguridad laboral.

Como objetivo de formación, se adquirirán los siguientes conocimientos:

- Aprendizaje del desarrollo de aplicaciones utilizando funcionalidades complejas de Power Apps.
- Aprendizaje de la creación de flujos automáticos usando Power Automate.
- Aprendizaje del desarrollo y utilización de listas de Sharepoint Online como base de datos.

En definitiva, el objetivo de este proyecto es el de desarrollar una aplicación centralizada para la gestión de las horas realizadas fuera de horario, que optimice el proceso de aprobación, la gestión de la información, y que optimice la toma de decisiones y el trabajo tanto de empleados, como de responsables y del personal de recursos humanos, a través de un entorno de desarrollo de aplicaciones como el de Microsoft.

1.4. Aplicaciones similares

A través de un estudio de mercado, se encontraron y analizaron aplicaciones similares. Las aplicaciones más comunes fueron las de control de tiempo en la empresa, tanto de horarios de entrada y salida, como de tiempo dedicado a diferentes tareas y proyectos. Esto proporciona aplicaciones poco específicas, pero que se adaptan bien a diversos entornos de trabajo, permitiendo ser usadas en diferentes dispositivos. Entre ellas, las más destacadas son:

1.4.1. Beebole

Se trata de una herramienta multiplataforma de gestión de tiempo, cuyo principal enfoque está en el control de proyectos y el seguimiento de tareas. Permite a la empresa registrar el tiempo dedicado a cada actividad o cliente, facilitando de esta forma la planificación y análisis de la productividad de los empleados. Se adapta bien a entornos donde se gestionan proyectos con múltiples tareas y turnos,

dato que permite visualizar el tiempo trabajado y la obtención de informes detallados. Aunque su enfoque es más amplio, su modelo de seguimiento de tiempo por tarea puede ser una forma eficiente de registrar guardias y trabajos programados.

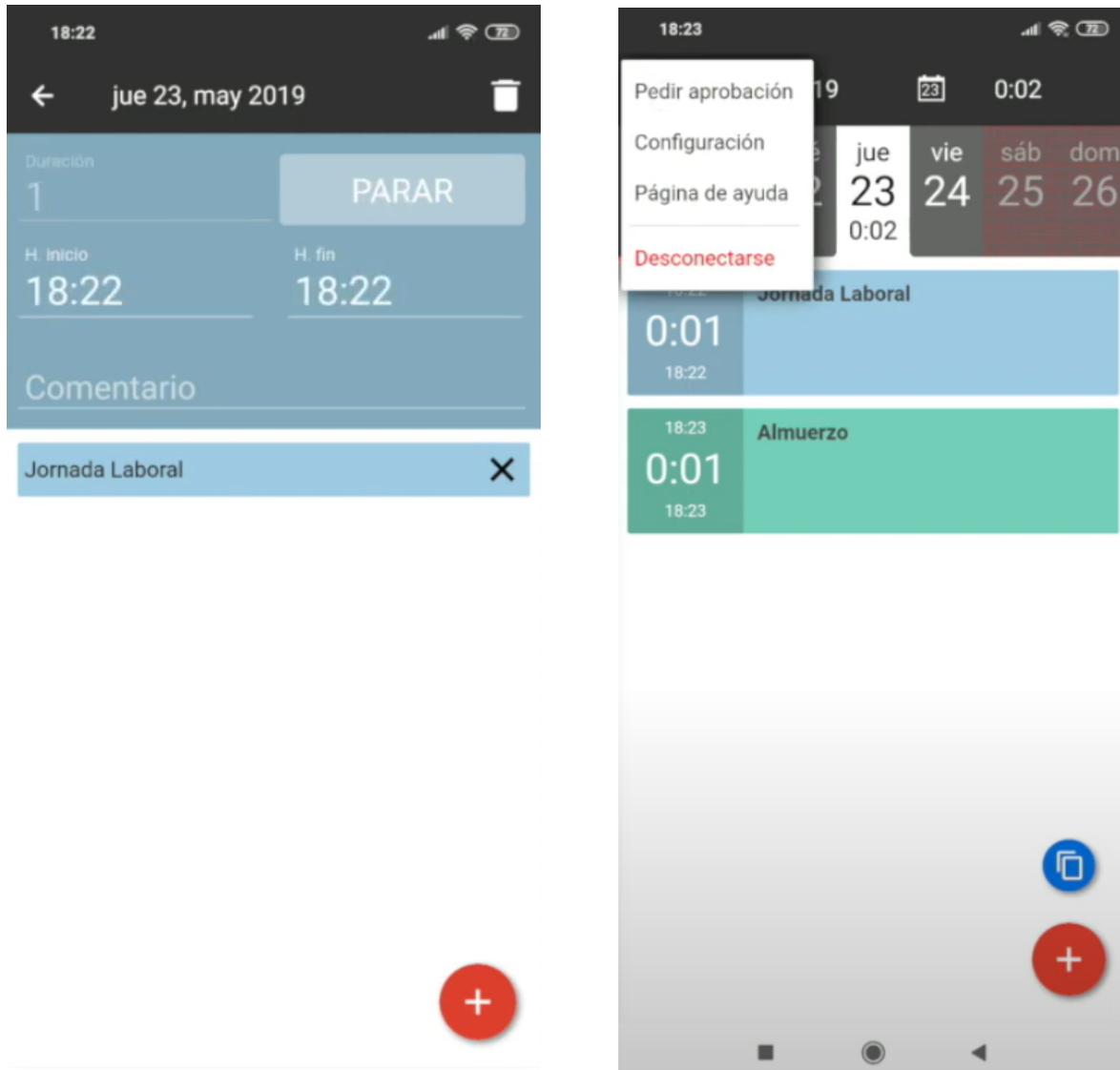


Figura 1.1: Introducir registro y solicitar aprobación

1.4.2. Clockify

Es una plataforma gratuita de seguimiento de tiempo que se utiliza para registrar horas trabajadas, gestionar proyectos y medir la productividad de equipos. Permite organizar el tiempo por tareas y proyectos. Esta herramienta permite a los usuarios fichar manualmente o mediante temporizador, y posteriormente enviar las horas a los aprobadores. Esto facilita el control horario en diferentes tipos de jornadas laborales incluyendo guardias. Esta aplicación permite también la obtención de informes detallados entre los que podemos encontrar información sobre el tiempo invertido.

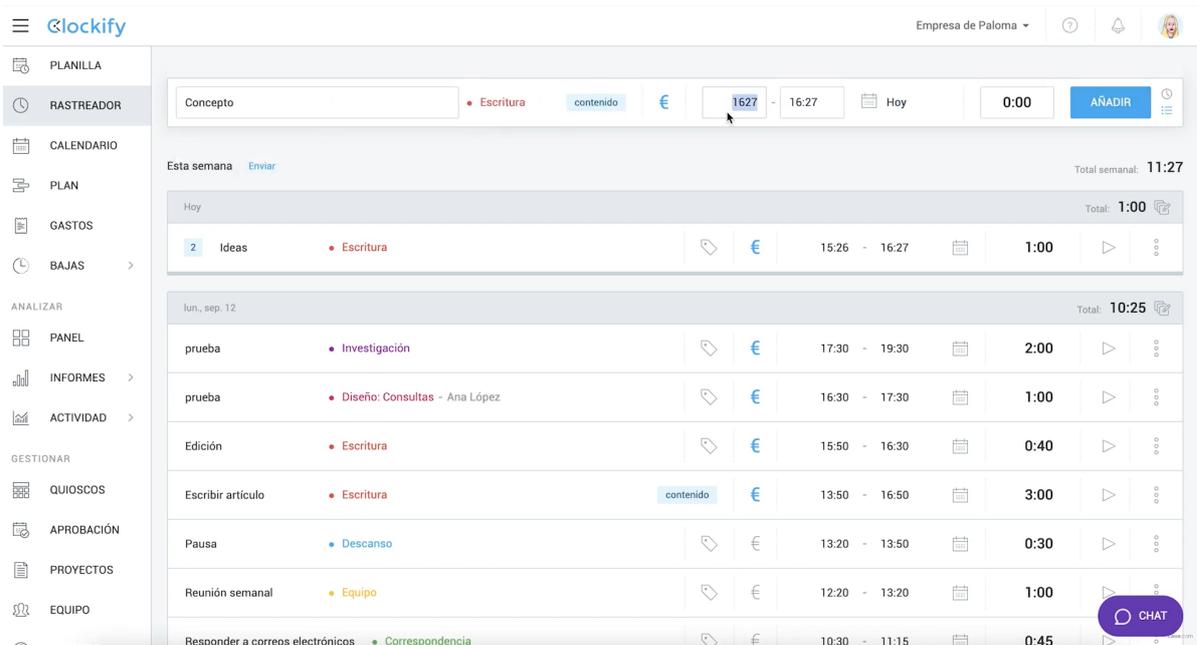


Figura 1.2: Aplicación Clockify

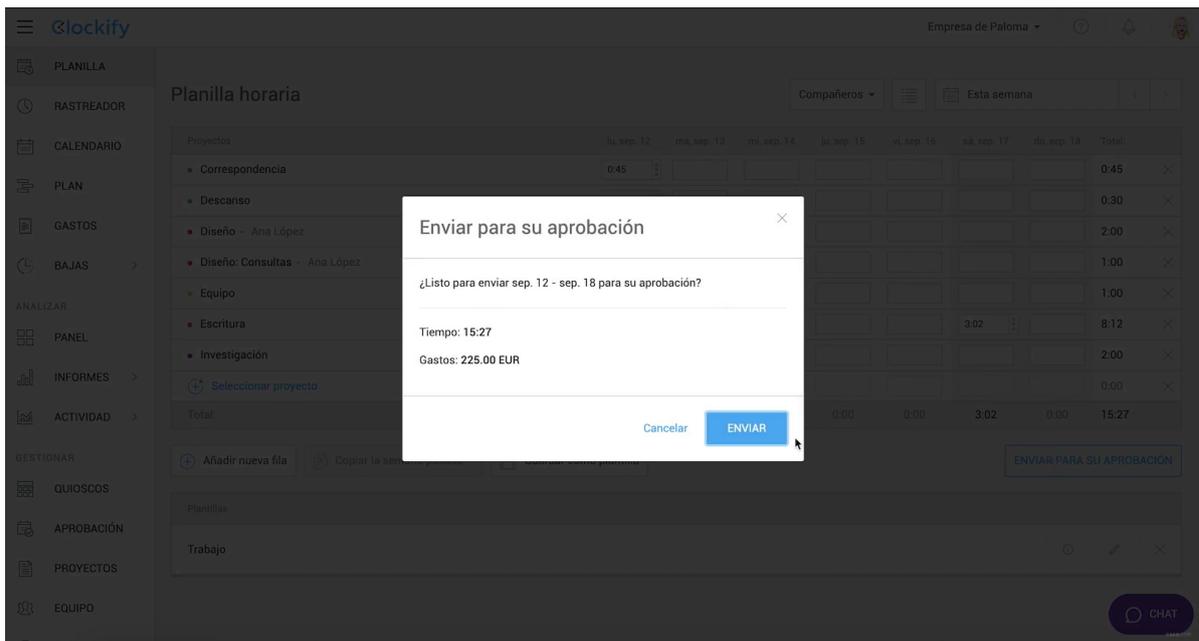


Figura 1.3: Solicitar aprobación en Clockify

1.4.3. TramitApp

TramitApp está más enfocada en la gestión de recursos humanos y en el cumplimiento de la normativa laboral. Proporciona herramientas para el fichaje, la gestión de ausencias, los permisos y las vacaciones. Al estar diseñada para cumplir con las normativas laborales de registro de jornada, se convierte en una herramienta muy potente para el control de horarios. También permite gestionar documentación relacionada con recursos humanos, como contratos y nóminas.

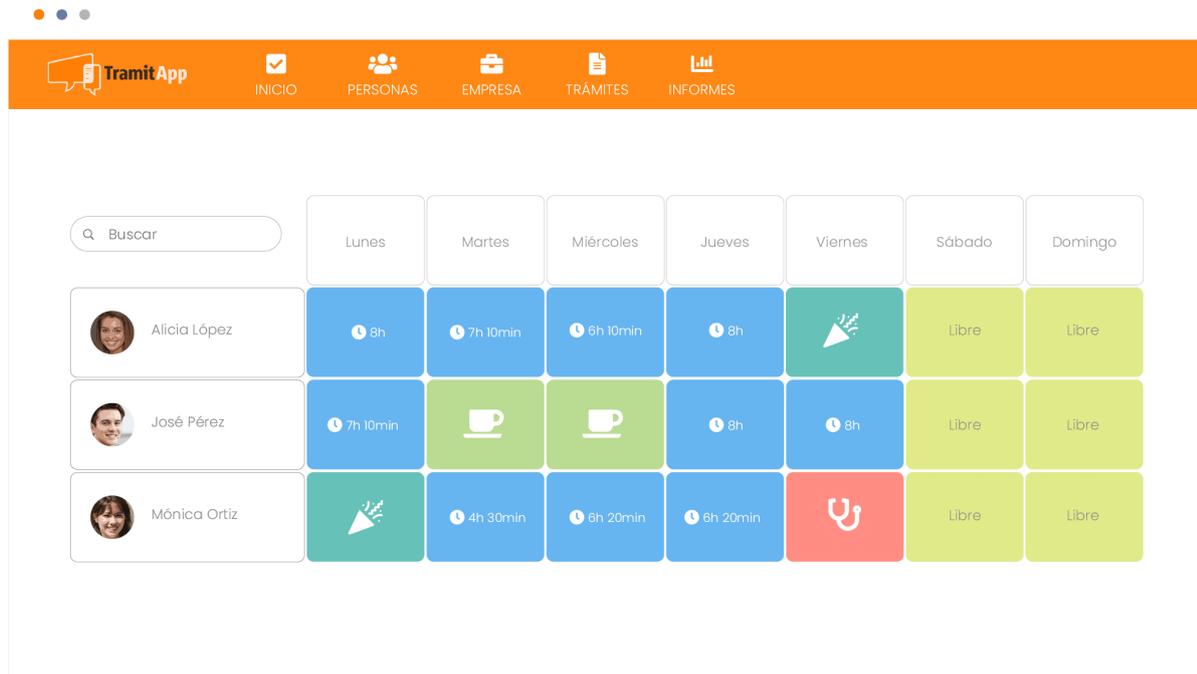


Figura 1.4: Aplicación Tramitapp

Por lo tanto y tras analizar los pros y los contras de estas tres aplicaciones, y con el fin de conseguir una herramienta más adaptada a la empresa que contara con un plus de utilidades, se decidió que nuestra aplicación iba a tener que cumplir con lo siguiente:

- Ofrecer una solución personalizada a las necesidades del cliente, garantizando la opción de introducir las horas extras distinguiendo entre si son guardias o son trabajos programados fuera del horario habitual.
- Ofrecer una solución multiplataforma, que garantice que puede ser usada en dispositivos diferentes, desde ordenadores hasta tablets o móviles, proporcionando la misma experiencia de usuario.
- Incluir en nuestra solución la posibilidad de generar informes sobre las horas registradas, costos y demás datos de interés.
- Añadir un sistema de mensajería power correo electrónico que permita un seguimiento de las incidencias.
- Facilitar el proceso de aprobación permitiendo recuperar las respuestas más frecuentes.

1.5. Formación y referencias relevantes

Durante el desarrollo de este TFG, algunas asignaturas han sido particularmente útiles de cara a proporcionar las bases conceptuales y técnicas necesarias para llevar a cabo tanto el desarrollo de la aplicación como para elaborar el proyecto.

1. **Programación Orientada a Objetos:** aunque PowerApps es una plataforma low code, los conceptos adquiridos en esta asignatura facilitaron la comprensión del modelado de datos y la lógica necesaria para gestionar incidencias. Fue especialmente relevante para el diseño de la estructura de la base de datos y las relaciones entre las distintas entidades de la aplicación.
2. **Análisis y Diseño de Bases de Datos:** esta asignatura fue esencial para definir y estructurar las listas de SharePoint que funcionan como base de datos. Los métodos de diseño y los diagramas de dominio y relacional aprendidos aquí fueron cruciales en la planificación del almacenamiento y la manipulación de datos.
3. **Fundamentos de Ingeniería de Software:** todo lo aprendido en esta asignatura fue clave para la elaboración del capítulo de análisis, en el que se definieron actores, casos de uso y modelo de dominio. En esta asignatura obtuve también valiosos conocimientos sobre Visual Paradigm, la herramienta utilizada para elaborar los diagramas.
4. **Planificación y Diseño de Sistemas Computacionales:** en esta asignatura aprendí a trabajar con la metodología ágil Scrum, algo que resultó fundamental para la organización de las tareas y el enfoque incremental basado en sprints. También aprendí el concepto de patrones de diseño que se aplicaron en el capítulo 4 4.1.
5. **Interacción Persona Computadora:** los conocimientos obtenidos sobre el diseño de interfaces gráficas atractivas visualmente y centradas en la usabilidad resultaron esenciales para el diseño de la interfaz de la aplicación.

Finalmente, todas las asignaturas relacionadas con la programación aportaron una base valiosa para implementar la lógica subyacente en la aplicación, integrando los conocimientos técnicos y prácticos necesarios para cumplir con los objetivos de manera efectiva.

Para la elaboración de la estructura y el desarrollo del TFG, se tomaron como referencia varios trabajos previos, en particular aquellos que abordaban proyectos similares con herramientas low code y automatización de procesos empresariales. Estos TFG sirvieron como guía en la definición de las secciones clave y ayudaron a estructurar la documentación de forma organizada y comprensible. Los principales son los de Héctor M. Hilanderas [1], Víctor Montequi [2] y Tomás Castaño [3].

1.6. Estructura de la memoria

- **Capítulo 1. Introducción:** es el capítulo en que nos encontramos. Aquí se muestra una introducción del proyecto, indicando su contexto, motivación y objetivos. También se hace referencia a las asignaturas más relevantes para la elaboración de este trabajo y la estructura de la memoria.
- **Capítulo 2. Planificación del proyecto:** a lo largo de este capítulo se describe la metodología empleada, la planificación, costes y riesgos del proyecto, así como los riesgos a tener en cuenta durante su desarrollo.

- **Capítulo 3. Análisis:** en este capítulo se definen los actores que usan la aplicación, los requisitos que esta debe tener, y se analizarán los diferentes casos de uso y el modelo de dominio del proyecto.
- **Capítulo 4. Diseño:** durante este capítulo se muestran los patrones de diseño tenidos en cuenta para el desarrollo de una aplicación a través de PowerApps, el diseño de la base de datos y el proceso de elaboración de la interfaz de usuario.
- **Capítulo 5. Implementación:** a lo largo de este capítulo se describen en profundidad las tecnologías utilizadas para desarrollar la plataforma final. También se muestran partes de la implementación de la solución final, el control de versiones y la seguridad.
- **Capítulo 6. Pruebas:** en este capítulo se muestran una serie de pruebas unitarias realizadas a las funcionalidades de la aplicación final. El conjunto de estas resulta en el plan de pruebas integral.
- **Capítulo 7. Despliegue:** durante este capítulo se explica el proceso de puesta en marcha en un entorno real de la aplicación con varios usuarios.
- **Capítulo 8. Seguimiento del proyecto:** en el transcurso de este capítulo se muestra el progreso real del proyecto y las horas dedicadas a las tareas frente a las estimadas. Para ello se usa la planificación inicial planteada en el capítulo 2.2.4.
- **Capítulo 9. Conclusiones y trabajo futuro:** en este último capítulo se presentan las conclusiones obtenidas tanto con el resultado final como con todo el proceso de desarrollo del proyecto. También se presentan una serie de propuestas para expandir el proyecto y aportar funcionalidad adicional en el futuro.
- **Bibliografía:** aquí se referencian todas las fuentes de información que han sido relevantes y utilizadas durante el proyecto.
- **Anexos:** se trata de material complementario que proporciona información adicional a lo presentado en el documento. Aquí podemos encontrar un manual de usuario para la aplicación, así como código relevante usado para algunas funcionalidades complejas.

Capítulo 2

Planificación del proyecto

2.1. Gestión del proyecto

La planificación de este proyecto ha presentado diversos desafíos, considerando el contexto en el que se ha desarrollado. Puesto que lo realiza un equipo con una experiencia limitada, y los objetivos iniciales del proyecto no estaban completamente definidos, se decidió que una planificación tradicional en cascada no era la opción más adecuada. Este tipo de planificación resulta más efectiva cuando contamos con unos requisitos y producto final claramente establecidos desde un primer momento, cosa que aquí no ocurría [4]. Por esta razón, la metodología a utilizar fue una metodología ágil, que se ajusta mejor a la naturaleza dinámica y cambiante de este proyecto [5]. Tras analizar varias opciones se decidió aplicar SCRUM, dado que era una metodología con la que ya había trabajado previamente, y que nos ofrecía una mayor flexibilidad para adaptarnos a las necesidades y cambios durante el desarrollo.

2.2. Metodologías ágiles

Las metodologías ágiles son un conjunto de enfoques para la gestión de proyectos que se caracterizan por su flexibilidad y capacidad de adaptación a los cambios. Surgieron como respuesta a las imitaciones de los métodos tradicionales, como el modelo en cascada [4], que tienden a ser más rígidos y lineales. Las metodologías ágiles permiten ciclos cortos y repetidos, y se centran en la entrega de valor continuo al cliente, facilitando la adaptación a requisitos que cambian durante el ciclo de vida del proyecto [5].

Las metodologías ágiles cuentan con las siguientes características principales:

- **Iterativas e incrementales:** el desarrollo se realiza en ciclos cortos, conocidos como iteraciones o "sprints". Cada iteración produce una versión funcional del producto, que permite ajustes y mejoras continuas [6].
- **Flexibilidad ante cambios:** a diferencia de los enfoques tradicionales, las metodologías ágiles

son tolerantes y se pueden adaptar a los cambios en los requisitos, incluso en fases avanzadas del desarrollo.

- **Enfoque en el cliente:** estas metodologías ponen al cliente en el centro del proceso, participando activamente para garantizar que el producto final cumple sus necesidades. El feedback y las entregas parciales tras los sprints resultan esenciales.
- **Colaboración y equipos autogestionados:** se fomenta la comunicación continua y la colaboración entre los miembros del equipo, así como con los clientes, de forma que todos los miembros del equipo se organicen para lograr los objetivos.
- **Entrega continua de valor:** en cada sprint se busca entregar al cliente partes funcionales del producto, de esta forma puede obtener un producto funcional desde las primeras etapas del proyecto, y sugerir cambios en caso de ser requeridos.
- **Prioridad al software funcional:** se toma como el principal indicador de progreso que el software funcione correctamente, dejando en un segundo plano la documentación exhaustiva.
- **Mejora continua:** las metodologías ágiles fomentan la reflexión y mejora después de cada iteración, a través de reuniones como se sugiere en SCRUM, para optimizar tanto los procesos como el producto.

Algunos ejemplos de metodologías ágiles populares son:

- **SCRUM:** esta metodología divide el trabajo en sprints, con roles definidos como el Scrum Master y el Product Owner, y eventos clave como las reuniones diarias, revisiones y retrospectivas.
- **Kanban:** se trata de un método visual que utiliza un tablero para gestionar el flujo de trabajo. De esta forma, se optimiza la eficiencia y se reduce el trabajo en proceso.
- **Extreme Programming (XP):** hace énfasis en prácticas como el desarrollo iterativo, la programación en parejas y la entrega frecuente de software. Es útil para equipos con un gran conocimiento en la tecnología,

En resumen, las metodologías ágiles presentan una serie de beneficios, como una mayor flexibilidad, un incremento en la satisfacción del cliente, una mayor eficiencia en el trabajo y una mejor capacidad para adaptarse a los cambios. Gracias a su capacidad para responder a las necesidades cambiantes del entorno empresarial actual, estas metodologías se han establecido como un enfoque eficaz y muy utilizado en el desarrollo y gestión de proyectos.

2.3. Scrum

Como se indicó en el apartado 2.1, tras analizar una serie de metodologías ágiles, se optó por trabajar con SCRUM, debido a que era un marco de trabajo con el que ya estaba familiarizado, y que a

Planificación del proyecto

parte cuenta con una gran adopción en el ámbito de la ingeniería de software, que encajaba con el tipo de aplicación que se iba a desarrollar.

SCRUM se trata de un marco ágil utilizado en la gestión de proyectos complejos que requieren de un enfoque iterativo e incremental. Su objetivo gira en torno a la entrega de valor a través de sprints, que son ciclos de trabajo cortos y predefinidos, que normalmente van de dos a cuatro semanas. SCRUM hace mucho hincapié en la colaboración entre los miembros del equipo y en la adaptación continua a los cambios de entorno y de requisitos del cliente.

Dentro de SCRUM podemos encontrar los siguientes roles:

- **Product Owner:** es el responsable de maximizar el valor del producto. Es quien define y prioriza el *Product Backlog* 2.3. Es quien actúa como el principal contacto con los stakeholders (grupos interesados) o clientes, para asegurar que el producto final cumpla con sus expectativas.
- **Scrum Master:** es quien garantiza que SCRUM se implemente de manera efectiva. Es importante destacar que no es quien gestiona al equipo, sino que actúa como facilitador y guía, asegurándose de esta forma que se sigan los principios y reglas de SCRUM.
- **Equipo de desarrollo:** con esto nos referimos al grupo de profesionales cuya función es la de entregar incrementos funcionales del producto en cada sprint. Una de las principales características de SCRUM es que este equipo es autogestionado y sus miembros deben colaborar estrechamente para lograr los objetivos del sprint.

SCRUM cuenta con las siguientes herramientas para gestionar el trabajo:

- **Product Backlog:** se trata de una lista priorizada de las funcionalidades, mejoras y correcciones a realizar en el producto. El Product Owner es el responsable de mantener actualizada esta lista.
- **Sprint Backlog:** es un subconjunto del anterior, que contiene los elementos con los que se trabajará en el sprint. Se gestiona y actualiza por el equipo de desarrollo.
- **Incremento:** es la suma de todos los elementos completados del Sprint Backlog, unido a todos los incrementos anteriores. Un incremento debe ser funcional y estar preparado para poderse entregar al cliente.

SCRUM se guía por una serie de eventos que se realizan con regularidad a fin de inspeccionar el proceso del producto y adaptarse a los cambios, estos eventos son:

1. **Sprint Planning:** es una reunión en la que se seleccionan los elementos del *Product Backlog* 2.3 que se trabajarán durante el sprint. El equipo es el encargado de discutir y estimar el trabajo necesario para completar los elementos elegidos.
2. **Daily Scrum:** se trata de una breve reunión que se lleva a cabo todos los días, en la que el equipo revisa el progreso que se ha llevado a cabo. Los miembros del equipo ponen en común que hicieron, las tareas a realizar y las dificultades que enfrentan.

3. **Sprint Review:** al finalizar un sprint, el equipo muestra el incremento del producto a los stakeholders, quienes proporcionan feedback. Esto es muy importante para asegurar que el desarrollo cumple las expectativas del cliente.
4. **Sprint Retrospective:** es una reunión posterior al sprint donde el equipo reflexiona sobre cómo mejorar el proceso de trabajo. Se identifican áreas de mejora para los próximos sprints lo que fomenta la mejora continua.

Finalmente, y por todo lo citado anteriormente, SCRUM ofrece numerosas ventajas sobre otras metodologías para la gestión de proyectos, entre las que podemos encontrar:

- **Flexibilidad:** permite una rápida adaptación a los cambios de requisitos, incluso en fases avanzadas del desarrollo.
- **Transparencia:** los eventos y herramientas de SCRUM proporcionan visibilidad sobre el progreso del equipo y el estado del proyecto en todo momento.
- **Entrega continua de valor:** a través de los sprints, se entrega valor de manera constante al cliente, permitiendo ajustes basados en feedback real.
- **Mejora continua:** las retrospectivas al final de cada sprint promueven la reflexión, mejora y optimización constante tanto en los procesos como en el trabajo en equipo.
- **Alta productividad:** los equipos autogestionados y multifuncionales permiten un trabajo más eficiente, al eliminar jerarquías innecesarias y fomentar la colaboración.
- **Compromiso y responsabilidad:** al ser un enfoque basado en equipos autogestionados, se fomenta un alto nivel de responsabilidad y compromiso con los objetivos.

2.4. Planificación inicial del proyecto

Una vez que se ha decidido que la metodología va a ser SCRUM, el trabajo se va a dividir en sprints, de forma que la funcionalidad se vaya consiguiendo de forma iterativa, aportando al final de cada sprint un avance tangible que permita la evaluación y adecuación a los requisitos iniciales.

Dado que la estimación inicial del trabajo es de 300 horas para cumplir con los 12 créditos requeridos, y el inicio del proyecto, que tuvo lugar el 30 de septiembre de 2024, se calculó un tiempo aproximado de trabajo de 25 horas semanales, o de media 5 horas diarias en un horario de lunes a viernes. Con esto, el proyecto se desarrollará en un plazo de doce semanas, que dividiremos en 6 sprints de 2 semanas cada uno, con los que se espera poder conseguir un producto funcional. Esto nos deja una fecha estimada de finalización en torno al 20 de diciembre. También nos reservaremos un sprint adicional para posibles retrasos, dado que durante el desarrollo del proyecto existe la posibilidad de que empiece a trabajar y tenga que reducir la cantidad de horas semanales que le pueda dedicar a realizar el proyecto.

Planificación del proyecto

Finalmente, al ser una estimación inicial, puede verse alterada por eventos ajenos a mi voluntad que serán contemplados en el posterior análisis de riesgos.

Cada sprint de dos semanas de duración, empezará con un *Sprint Planning*, en los que definiré los objetivos del sprint y las tareas para lograrlo. Una vez que se termine el sprint se hará una review, conocida como *Sprint Review*, en la que analizaré si se han logrado los objetivos. También se contactará con regularidad, tanto si surgen dudas a cerca de la planificación o la estructura del proyecto, como para ser supervisado en cuestiones como la realización de la memoria, o en cualquier aspecto en el que pueda aportarme tanto ideas como conocimiento, a mi tutor Jesus María Vegas Hernández.

Sprint	Fecha Inicio	Fecha Fin
Sprint 1	Lunes 30 Septiembre, 2024	Viernes 11 Octubre, 2024
Sprint 2	Lunes 14 Octubre, 2024	Viernes 25 Octubre, 2024
Sprint 3	Lunes 28 Octubre, 2024	Viernes 8 Noviembre, 2024
Sprint 4	Lunes 11 Noviembre, 2024	Viernes 22 Noviembre, 2024
Sprint 5	Lunes 25 Noviembre, 2024	Viernes 6 Diciembre, 2024
Sprint 6	Lunes 9 Diciembre, 2024	Viernes 20 Diciembre, 2024

Tabla 2.1: Planificación del proyecto en sprints

2.5. Análisis de costes

El análisis de costes es una parte fundamental en el ciclo de vida de cualquier proyecto de desarrollo de software.

El proceso de desarrollo de software, desde la fase de planificación hasta su implementación, implica una serie de costos directos e indirectos que deben ser considerados para asegurar la viabilidad del proyecto.

Cabe destacar que este proyecto se ha realizado en un entorno académico, siendo un Trabajo de Fin de Grado, por lo que el coste del equipo de desarrollo es nulo, sin embargo y a fin de ser extrapolable a un entorno empresarial, se tendrá en cuenta el precio por hora de un desarrollador junior de PowerPlatform. Para ello se ha obtenido el sueldo medio a través del portal *Glassdoor* [7], que es de

22.231 €/año de sueldo base si trabaja a jornada completa. Si tenemos en cuenta que trabaja alrededor de 2080 horas al año, esto nos deja un precio por hora de 10.68€/hora. El tiempo de trabajo total estimado asciende a 300 horas, que a al precio anteriormente citado por hora, dejaría un costo total de 3206.39€.

En cuanto a la suscripción de Microsoft 365, para nuestro caso está proporcionada gratuitamente por la UVa, sin embargo, en el caso de no estar bajo el amparo de esta institución, para una empresa el precio sería de 11.70€/mes [8]. Con esto obtendremos SharePoint y las funcionalidades de Power Automate que realizará nuestra aplicación. En cuanto a la licencia de Power Apps, podemos utilizar el plan gratuito para desarrolladores que tiene coste 0 [9]. Esto nos deja un coste total en 3 meses de desarrollo del proyecto de 35.10€.

Finalmente, en cuanto al coste de hardware, se ha usado un equipo Lenovo Ideapad 3 con un precio de 499€, teniendo en cuenta un tiempo de vida útil de 5 años, y que el proyecto ha durado 3 meses, lo que nos da un 5% su vida útil. Por lo tanto su precio amortizado es de 24.95€. También se ha usado un monitor Philips [10] con un precio de 100€, cuyo precio amortizado es de 5€ aproximadamente.

Los costos, que ascienden a un total de 3271.44€, pueden verse desglosados en la tabla a continuación 2.2:

Categoría	Concepto	Costo (€)
Costes de Personal	Ingeniero informático Junior	3206.39€
Costes de Software	Subscripción Microsoft 365	35.10€
	Subscripción Power Apps	0€
Costes de Hardware	Lenovo Ideapad 3	24.95€
	Monitor Philips	5€
Total		3271.44€

Tabla 2.2: Costes finales del proyecto

2.6. Análisis de riesgos

En la ingeniería de software, el análisis de riesgos es una práctica esencial para anticipar y mitigar problemas potenciales que puedan afectar el éxito del proyecto. Este proceso implica identificar, evaluar y planificar estrategias para minimizar el impacto de riesgos específicos [11]. La correcta gestión de riesgos permite a integrantes del proyecto adaptarse a eventos inesperados, mejorando de esta forma la probabilidad de éxito [12].

El análisis de riesgos se realiza generalmente en tres etapas: *identificación*, *evaluación* y *mitigación*. Durante la identificación, se hace una lista de los posibles riesgos, como fallos técnicos o cambios en los requisitos. En la evaluación, se analiza la probabilidad y el impacto de cada riesgo identificado. Finalmente, en la mitigación, se desarrollan estrategias para reducir la probabilidad de ocurrencia o

Planificación del proyecto

minimizar su impacto [13, 14].

Una herramienta fundamental en esta fase es la **matriz de análisis de riesgos**, que permite organizar y priorizar los riesgos según su probabilidad y severidad. Esta matriz facilita la visualización de los riesgos más críticos y ayuda a definir estrategias adecuadas de mitigación [15]. Cada riesgo en esta matriz tiene, tanto una *probabilidad*, como un *impacto*, que se medirán en una escala del 1 al 5, siendo el 1 el valor más bajo y el 5 el más alto. Esto nos da para las probabilidades: “Raro”, “Improbable”, “Moderado”, “Probable”, y “Casi Seguro”. Para el impacto tenemos los siguientes niveles: “Insignificante”, “Menor”, “Significativo”, “Mayor” y “Grave”.

A continuación, encontramos la matriz en la figura 2.1

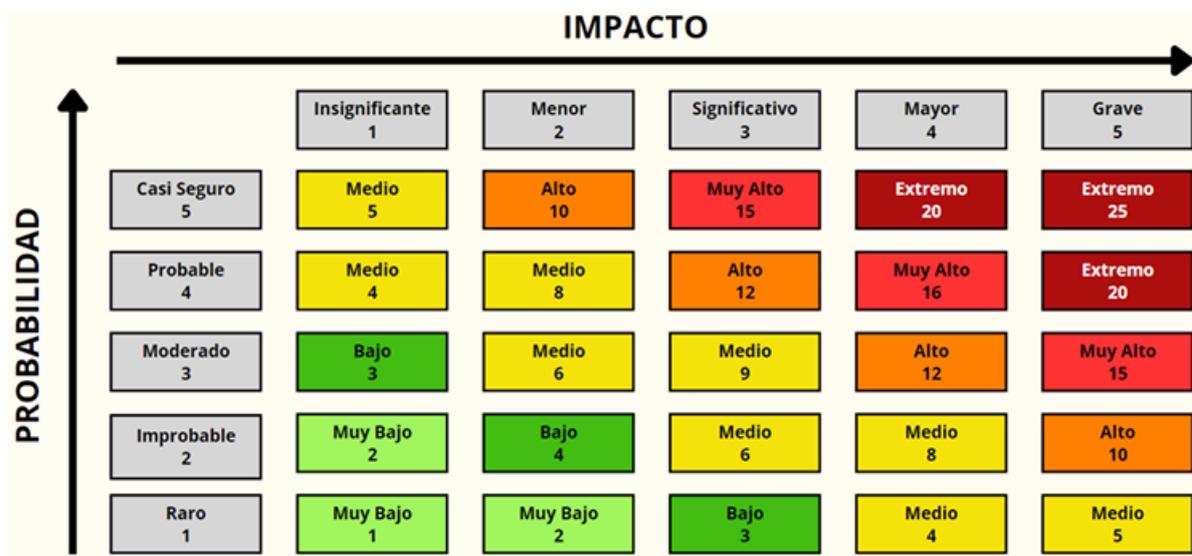


Figura 2.1: Matriz de riesgos; tomada de [16]

Es importante que la gestión de riesgos sea un proceso continuo y dinámico, ajustándose conforme avanza el proyecto y se obtiene más información sobre posibles riesgos [17]. Mantener la evaluación y mitigación de riesgos actualizadas reduce la incertidumbre y mejora la toma de decisiones [18].

A continuación, se presenta una lista de los posibles riesgos tenidos en cuenta, con la probabilidad de ocurrencia, su impacto, el nivel de riesgo, un plan de mitigación y uno de contingencia.

RIESGO 1	Organización y/o planificación incorrecta
DESCRIPCIÓN	Las tareas pueden diferir de la planificación inicial estimada
PROBABILIDAD	3-Moderada
IMPACTO	3-Significativo
NIVEL DE RIESGO	9-Medio
PLAN DE MITIGACIÓN	Realización de revisiones periódicas en las que se ajuste la planificación al progreso real de las tareas.
PLAN DE CONTINGENCIA	Utilizar una distribución de tareas flexible, y usar el sprint adicional para absorber retrasos imprevistos. Reasignación de recursos y prioridades para completar las tareas críticas

Tabla 2.3: Riesgo 1: Fallo en la planificación

RIESGO 2	Falta de conocimiento en las tecnologías
DESCRIPCIÓN	Se trata de un proyecto realizado por un equipo de desarrollo poco experimentado, por lo que la falta de conocimiento puede dificultar el desarrollo del producto.
PROBABILIDAD	4-Probable
IMPACTO	3-Significativo
NIVEL DE RIESGO	12-Alto
PLAN DE MITIGACIÓN	Las tareas con tecnologías desconocidas serán estimadas con más horas, dedicando el tiempo extra a la formación.
PLAN DE CONTINGENCIA	Emplear tiempo en la formación sobre las tecnologías a utilizar, lo que garantice un buen resultado. Contactar con personas con más experiencia que nos puedan resolver problemas puntuales.

Tabla 2.4: Riesgo 2: Falta de conocimiento

RIESGO 3	Responsabilidades adicionales al proyecto
DESCRIPCIÓN	Durante el tiempo de desarrollo de este proyecto, estoy buscando trabajo, con un horario que puede limitar el tiempo de desarrollo dedicado, especialmente en los últimos sprints, debido a una jornada laboral completa, limitando el desarrollo a los fines de semana.
PROBABILIDAD	4-Probable
IMPACTO	3-Significativo
NIVEL DE RIESGO	12-Alto
PLAN DE MITIGACIÓN	Ajustar las tareas de los sprints a la carga de trabajo total, priorizando las tareas que lleven más horas en los sprints en los que tengamos más tiempo.
PLAN DE CONTINGENCIA	Organizar el tiempo de forma eficiente y que el cronograma tenga en cuenta la carga de trabajo total.

Tabla 2.5: Riesgo 3: Responsabilidades adicionales

RIESGO 4	Ausencia o enfermedad del desarrollador
DESCRIPCIÓN	Dado que el proyecto cuenta con un único desarrollador, las ausencias por vacaciones, enfermedad o motivos personales causarán una demora significativa en los tiempos de desarrollo y entrega.
PROBABILIDAD	2-Improbable
IMPACTO	4-Mayor
NIVEL DE RIESGO	8-Medio
PLAN DE MITIGACIÓN	Priorizar la salud y planificar de tal manera que se puedan suplir estos imprevistos. Disponer de un proyecto bien documentado para que se pueda retomar con facilidad.
PLAN DE CONTINGENCIA	Reprogramación de las tareas que no hayan sido realizadas durante la ausencia, si esta fuera muy prolongada, hacer uso del <i>sprint adicional</i> . Disponer de una forma de trabajar en este proyecto sin importar la ubicación.

Tabla 2.6: Riesgo 4: Ausencia

Planificación del proyecto

RIESGO 5	Fallo de disponibilidad de las tecnologías y herramientas
DESCRIPCIÓN	Todas las herramientas usadas son en red, por lo que si se encuentran en mantenimiento o no disponemos de conexión, no se puede trabajar en el proyecto
PROBABILIDAD	2-Improbable
IMPACTO	5-Grave
NIVEL DE RIESGO	10-Alto
PLAN DE MITIGACIÓN	Tratar de disponer siempre un acceso robusto a la red, y evitar acciones que puedan interrumpir nuestra conexión
PLAN DE CONTINGENCIA	Disponer de copias locales actualizadas para poder restaurar tanto de la aplicación como de la documentación en caso de que perdiéramos la información.

Tabla 2.7: Riesgo 5: Fallo de disponibilidad tecnológico

RIESGO 6	Fallo en el equipo informático
DESCRIPCIÓN	El ordenador portátil podría fallar y perder la información que contiene en local
PROBABILIDAD	1-Raro
IMPACTO	2-Menor
NIVEL DE RIESGO	2-Muy bajo
PLAN DE MITIGACIÓN	Disponer de copias de seguridad del proyecto en dispositivos externos y en la nube, de forma que se puedan restaurar. Tener guardadas las cuentas y contraseñas usadas para la elaboración tanto del proyecto como de la documentación.
PLAN DE CONTINGENCIA	Adquirir otro equipo con el que poder seguir con el desarrollo del proyecto.

Tabla 2.8: Riesgo 6: Fallo en el equipo informático

RIESGO 7	Pérdida del proyecto
DESCRIPCIÓN	El perder el proyecto o la documentación significaría tener que repetir el trabajo realizado. Dado que la mayor parte del trabajo se realiza en la nube es extraño que esto ocurra, pero también dependemos de información que no está guardada en local.
PROBABILIDAD	1-Raro
IMPACTO	5-Grave
NIVEL DE RIESGO	10-Medio
PLAN DE MITIGACIÓN	Realizar copias de seguridad recurrentes que guardemos en diferentes lugares, tanto del proyecto como de la documentación que nos permita reestablecer una versión reciente.
PLAN DE CONTINGENCIA	Realizar una documentación clara y actualizada que permita, en caso de pérdida recuperar el trabajo realizado hasta el momento.

Tabla 2.9: Riesgo 7: Pérdida del proyecto.

RIESGO 8	El proyecto no cumple su cometido inicial
DESCRIPCIÓN	El proyecto podría no estar a la altura de los objetivos definidos en el punto 1.3, lo que provocaría insatisfacción en el producto o una demora en los plazos.
PROBABILIDAD	1-Raro
IMPACTO	5-Grave
NIVEL DE RIESGO	5-Medio
PLAN DE MITIGACIÓN	Cada vez que se complete un sprint, evaluar si el resultado está a la altura de las expectativas, y si se ha cumplido con los plazos establecidos. Adicionalmente es recomendable tener reuniones frecuentes con el tutor para evaluar el progreso obtenido hasta el momento.
PLAN DE CONTINGENCIA	Marcar unos objetivos alcanzables, con hitos al final de cada sprint que sean medibles para evaluar el progreso y efectuar correcciones.

Tabla 2.10: Riesgo 8: Incumplimiento de objetivos

RIESGO 9	Requisitos mal definidos y/o cambiantes
DESCRIPCIÓN	Los requisitos podrían cambiar o estar mal definidos de forma que no fueran viables o no se adecuaran a los objetivos definidos en 1.3. Esto podría llevar a descartar requisitos anteriores.
PROBABILIDAD	2-Improbable
IMPACTO	3-Significativo
NIVEL DE RIESGO	6-Medio
PLAN DE MITIGACIÓN	Revisar el profundidad los requisitos, y realizar evaluaciones frecuentes de los cambios en los requisitos para modificar el producto.
PLAN DE CONTINGENCIA	Reevaluar los requisitos y comprobar si el producto realizado hasta el momento se ciñe a ellos.

Tabla 2.11: Riesgo 9: Requisitos cambiantes o mal definidos

RIESGO 10	Cambios en las tecnologías a utilizar
DESCRIPCIÓN	Al realizarse con tecnologías que no están totalmente asentadas, los cambios y actualizaciones pueden afectar a la compatibilidad y funcionamiento futuro del producto.
PROBABILIDAD	3-Moderado
IMPACTO	3-Significativo
NIVEL DE RIESGO	9-Medio
PLAN DE MITIGACIÓN	Realizar un seguimiento de los cambios y actualizaciones de las tecnologías que usa nuestra aplicación y procurar mantenerla actualizada en la última versión del software.
PLAN DE CONTINGENCIA	Definir un proceso de actualización, acompañado de una documentación exhaustiva que permita acceder a los puntos en los que la aplicación falla.

Tabla 2.12: Riesgo 10: Cambio en las tecnologías

Planificación del proyecto

RIESGO 11	Los usuarios finales no se adaptan a la aplicación
DESCRIPCIÓN	Puede ocurrir que los usuarios finales no estén familiarizados con aplicaciones de este tipo y expresen dificultades para adaptarse a ella.
PROBABILIDAD	2-Improbable
IMPACTO	3-Significativo
NIVEL DE RIESGO	6-Medio
PLAN DE MITIGACIÓN	Realizar pruebas de usabilidad en la aplicación, así como un manual de usuario detallado que recoja la funcionalidad de la aplicación.
PLAN DE CONTINGENCIA	Ofrecer formación a los usuarios para que se adapten de mejor manera a la nueva aplicación

Tabla 2.13: Riesgo 11: Falta de adaptación a la aplicación

RIESGO 12	Tiempo de desarrollo limitado
DESCRIPCIÓN	El trabajar con <i>sprints</i> que nos obligan a dar funcionalidad en un tiempo de desarrollo limitado, puede repercutir en la calidad del desarrollo y en la implementación adecuada
PROBABILIDAD	2-Improbable
IMPACTO	4-Mayor
NIVEL DE RIESGO	8-Medio
PLAN DE MITIGACIÓN	Realizar una estimación del tiempo consecuente con los objetivos, tiempo disponible y capacitación en las tecnologías. Realizar un seguimiento que permita reajustes en la planificación y objetivos de cada <i>sprint</i>
PLAN DE CONTINGENCIA	Reajustar los objetivos del <i>sprint</i>

Tabla 2.14: Riesgo 12: Limitación de tiempo

RIESGO 13	Inexperiencia trabajando con SCRUM
DESCRIPCIÓN	El equipo no dispone de mucha experiencia trabajando con SCRUM, lo que puede afectar a la estimación y priorización de tareas.
PROBABILIDAD	4-Probable
IMPACTO	3-Significativo
NIVEL DE RIESGO	12-Alto
PLAN DE MITIGACIÓN	Seguir la metodología de SCRUM, evaluando frecuentemente si nos estamos desviando de sus bases.
PLAN DE CONTINGENCIA	Reevaluar donde se ha fallado del proceso definido en SCRUM, y reajuste de las estimaciones y las tareas prioritarias.

Tabla 2.15: Riesgo 13: Inexperiencia trabajando con SCRUM

Capítulo 3

Análisis

En esta sección se abordará el análisis de la aplicación. Se trata de un proceso fundamental que sienta las bases de un desarrollo exitoso. En esta fase identificaremos los actores y sus roles, también definiremos los requisitos del sistema.

Esta fase resulta crucial para comprender en profundidad las necesidades de los usuarios y los objetivos del proyecto. Identificar a los usuarios de la aplicación y las funcionalidades necesarias nos permite garantizar que el producto final esté a la altura de las expectativas y requisitos del cliente final. Esta etapa consigue minimizar riesgos a la vez que optimiza los recursos, haciendo del desarrollo algo más ágil y eficiente.

3.1. Actores y roles

En el contexto de nuestra aplicación, existen 3 actores principales que interactúan con la aplicación, el personal de recursos humanos (*People*), los Managers, y los Empleados. Todos estos usuarios accederán a la aplicación a través de una cuenta designada de Microsoft, lo que garantiza un nivel de seguridad alto. Los usuarios solo podrán acceder a las pantallas específicas de su rol.

En la aplicación existe también una pantalla login, que permite que adicionalmente a la funcionalidad de autenticación de Microsoft, podamos disponer de usuarios en la aplicación y a través de una misma cuenta de Microsoft podamos acceder a diferentes perfiles. Esto garantiza una capa más de seguridad y privacidad, pues no solo vale con el usuario de Microsoft, si no que es necesario el personal de la empresa.

Los usuarios y sus roles, serán introducidos en las listas de Sharepoint por responsables de la organización.

- **Personal de People:** Se trata del departamento de personal, que de ahora en adelante llamaremos *People*. Este actor es el encargado de enviar para pago las solicitudes de los Empleados que hayan sido aprobadas por su Manager, también puede rechazarlas, añadiendo un comentario al

respecto. Tiene acceso a la información de todas las solicitudes que le llegan. Además tiene la capacidad de obtener informes que abarquen todas las solicitudes que le aparecen en el panel.

- **Manager:** el Manager es el encargado de aprobar o rechazar las incidencias que le llegan de los Empleados a su cargo. Si las aprueba las enviará a que el Personal de People realice sus gestiones correspondientes. También puede cancelar incidencias que haya aprobado, por lo que el Personal de People no podrá enviarlas para pago. Si el Manager rechaza las incidencias, volverán al Empleado para que realice las modificaciones pertinentes antes de volver a enviarla.

El Manager además puede acceder a detalles de las incidencias que le aparecen en el panel y obtener informes personalizados por correo electrónico sobre sus incidencias asociadas, filtrando por fecha.

Finalmente, un Manager puede ser también Empleado.

- **Empleado:** el Empleado es quien introduce las nuevas incidencias, tanto guardias, como trabajos programados, rellenando los formularios en su pantalla, la *Pantalla Empleado*. También puede editar incidencias que todavía no haya enviado a su Manager. Una vez que tiene la incidencia creada, puede acceder a sus detalles, editarla, enviarla para ser aprobada por su Manager, o borrarla. Si la incidencia ha sido enviada, puede cancelarla para editarla.

Cada empleado pertenece a un nivel determinado, 1, 2 o 3 en función de su puesto y responsabilidad en la empresa, permitiendo de esta forma calcular una remuneración acorde a su nivel.

3.2. Requisitos

La especificación de requisitos constituye una fase fundamental del desarrollo software, pues es aquí donde se sientan las bases de todo lo que va a hacer nuestro proyecto, de forma que satisfaga a todas las partes interesadas.

En nuestro caso, el proyecto está desarrollado con una metodología ágil [5], en la que los requisitos se gestionan de forma flexible y evolutiva, definiéndose a lo largo de los sprints. Con esto se consigue el objetivo de entregar valor al cliente, adaptarse a las necesidades del momento, ser flexible a cambios conforme avanza el proyecto, priorizar nuevas funcionalidades y permitir una mejora incremental, mejorando el resultado final.

Para este proyecto se tendrán en cuenta los requisitos funcionales, referentes a las funcionalidades específicas que debe ofrecer el sistema y los no funcionales, que definen características relacionadas con el rendimiento, seguridad o usabilidad. También se mostrarán los requisitos de información, que especifican los datos que el sistema debe gestionar y su estructura y finalmente las reglas de negocio, que establecen las condiciones y restricciones bajo las cuales opera el sistema, asegurando que cumpla con las políticas y normas de la organización.

3.2.1. Requisitos funcionales

- **RF01:** El sistema permitirá al usuario acceder a la aplicación, si está dado de alta.
- **RF02:** El sistema notificará al usuario si su usuario no está en el sistema o rechaza el inicio de sesión.
- **RF03:** El sistema dispondrá de distintas pantallas para los roles de *People*, *Manager* y *Empleado*.
- **RF04:** El sistema permitirá crear nuevas incidencias que podrán ser de tipo guardia o trabajo programado.
- **RF05:** Para las guardias, el sistema permitirá registrar la fecha de inicio, los festivos, las intervenciones realizadas, y los comentarios.
- **RF06:** Para los trabajos programados el sistema permitirá registrar la fecha, la hora de inicio y la de fin. Será el sistema el que calcule las horas totales.
- **RF07:** El sistema permitirá a los Empleados editar sus incidencias.
- **RF08:** El sistema permitirá a los Empleados filtrar sus incidencias por tipo, estado, identificador y fecha de creación.
- **RF09:** El sistema permitirá a los Empleados enviar a su Manager asignado la incidencia para que la tramite.
- **RF10:** El sistema permitirá a los Empleados cancelar una incidencia enviada.
- **RF11:** El sistema permitirá a los Empleados borrar una incidencia creada y que no haya sido enviada.
- **RF12:** El sistema permitirá realizar en más de una incidencia, de forma simultánea las acciones de enviar, cancelar o borrar.
- **RF13:** El sistema permitirá a los Managers consultar los detalles de las incidencias que les lleguen.
- **RF14:** El sistema permitirá a los Managers filtrar las incidencias por el nombre del Empleado a su cargo, el tipo, el estado, identificador y fecha en la que fue emitida.
- **RF15:** El sistema permitirá a los Managers aprobar, cancelar aprobaciones y rechazar incidencias que le aparezcan en su pantalla.
- **RF16:** El sistema permitirá a los Managers realizar las aprobaciones, cancelaciones y rechazos en varias incidencias simultáneamente.
- **RF17:** El sistema permitirá a los Managers generar reportes de las incidencias de su panel, filtrando fechas. Estos reportes serán entregados por correo.
- **RF18:** El sistema mostrará al personal de People todas las incidencias que estén aprobadas por Managers y las que estén enviadas para pago.

- **RF19:** El sistema permitirá al Personal de People consultar los detalles de las incidencias recibidas.
- **RF20:** El sistema permitirá al personal de People, tanto enviar para pago las incidencias recibidas como rechazarlas.
- **RF21:** El sistema permitirá al personal de People filtrar las incidencias por el nombre tanto del Empleado como del Manager, tipo, estado, identificador o fecha en la que fue aprobada.
- **RF22:** El sistema permitirá al personal de People recibir por correo reportes con todas las incidencias de su panel.
- **RF23:** El sistema deberá notificar por correo al Empleado si su incidencia ha sido aprobada o rechazada por su Manager.
- **RF24:** El sistema deberá notificar por correo tanto al Empleado como al Manager si la solicitud ha sido enviada para pago por el personal de People.
- **RF25:** El sistema deberá notificar por correo al Manager que le ha llegado una incidencia para aprobar o rechazar.
- **RF26:** El sistema deberá notificar por correo tanto al Empleado como al Manager si la solicitud se rechaza por el Personal de People.
- **RF27:** El sistema deberá notificar por correo al Empleado si la solicitud ha sido cancelada por su Manager.
- **RF28:** El sistema deberá notificar al Manager si el Empleado cancela una solicitud que ha sido enviada.
- **RF29:** El sistema permitirá desde una misma cuenta de Microsoft entrar con diferentes usuarios, cada uno con unas credenciales.
- **RF30:** Cuando el usuario interactúe con una pantalla superpuesta, como un menú o pop up, podrá cerrarse automáticamente si se hace click fuera de esa pantalla, emulando el comportamiento de aplicaciones móviles táctiles.
- **RF31:** El sistema deberá calcular automáticamente la remuneración de los trabajos programados en base a las horas trabajadas y el nivel del empleado.
- **RF32:** El sistema deberá calcular la remuneración de las guardias con una base fija semanal, más un extra por las horas de intervención y festivos trabajados, teniendo en cuenta las diferentes tarifas para cada nivel de empleado.
- **RF33:** La aplicación incluirá un botón en todas las interfaces que permita acceder al manual de usuario. Este botón abrirá un archivo PDF almacenado en una ubicación accesible para los usuarios con permisos adecuados.
- **RF34:** El sistema permitirá tanto a los Managers como al Personal de People, usar sus respuestas más frecuentes para aprobar o rechazar incidencias.

3.2.2. Requisitos no funcionales

- **RNF01:** La interfaz de usuario tiene que ser intuitiva para que usuarios con poca experiencia en aplicaciones realizadas con Power Apps puedan usarla sin dificultad. Las principales funciones deben ser accesibles en menos de 3 *clicks*.
- **RNF02:** La aplicación se integrará en el entorno de SharePoint de la empresa.
- **RNF03:** El sistema utilizará Outlook para tramitar el acceso a la aplicación y para enviar las notificaciones por correo electrónico.
- **RNF04:** El sistema enviará los informes solicitados a través de Outlook.
- **RNF05:** La aplicación deberá estar disponible durante el total de la jornada laboral.
- **RNF06:** El sistema debe poder manejar grandes volúmenes de datos sin reducir su rendimiento.
- **RNF07:** El sistema debe ser capaz de soportar varios usuarios simultáneamente sin empeorar sus prestaciones y debe ser escalable a más usuarios.
- **RNF08:** La aplicación debe garantizar la integridad de los datos, evitando la pérdida de información o los duplicados.
- **RNF09:** El sistema debe ser accesible desde diferentes dispositivos, como ordenadores, tablets y teléfonos móviles.
- **RNF10:** El sistema deberá ser fácilmente adaptable a otros idiomas en el caso de que se quiera distribuir en el extranjero.
- **RNF11:** El sistema deberá ser adaptable a la estética corporativa de la compañía en que se implante.
- **RNF12:** El código deberá estar exhaustivamente documentado de forma que facilite futuras modificaciones o mantenimientos.
- **RNF13:** El archivo del manual de usuario deberá estar almacenado en un entorno SharePoint seguro y accesible, garantizando su disponibilidad para los usuarios autorizados.

3.2.3. Requisitos de información

- **RFI01:** Para las guardias, el sistema almacenará el identificador de la incidencia, el tipo, los comentarios, el estado, los festivos, las fechas de inicio y fin, las totales y el identificador nombre y email tanto del aprobante como del emisor. También almacenará las fechas de creación, emisión, aprobación y envío a pago. Finalmente almacenará el importe.
- **RFI02:** El sistema almacenará las intervenciones asociadas a cada guardia, indicando la guardia a la que pertenecen, la fecha de la intervención, el número de horas, y los comentarios.

- **RFI03:** Para los trabajos programados, el sistema almacenará el identificador de la incidencia, el tipo, los comentarios, el estado, la fecha en que tuvo lugar, el número de horas, así como la hora de inicio y de fin y el importe. Asimismo almacenará el identificador, nombre y email tanto del aprobante como del emisor. Finalmente almacenará las fechas de creación, emisión, aprobación y envío a pago de la incidencia.
- **RFI04:** El sistema almacenará la relación entre Empleados y Managers.
- **RFI05:** El sistema almacenará para cada integrante de la compañía su email, su identificador único, su nombre y su rol.
- **RFI06:** El sistema debe almacenar los comentarios introducidos por los usuarios al aprobar o rechazar una incidencia, diferenciándolos entre aprobación o rechazo.
- **RFI07:** El sistema debe almacenar los honorarios para cada nivel de empleado, incluyendo la tarea a la que se refieren (base fija, hora de guardia, hora de trabajo programado y festivo) y la fecha en la que estos han sido modificados, manteniendo un histórico de cambios.

3.2.4. Reglas de negocio

- **RN01:** Todas las incidencias deben ser aprobadas por un Manager antes de considerarse válidas.
- **RN02:** Un Manager puede ser Empleado también, pero tendrá otro Manager que le pueda aprobar las incidencias.
- **RN03:** Los Empleados son los únicos que pueden registrar intervenciones.
- **RN04:** Cualquier cambio en la aprobación, rechazo o cancelación debe ser comunicado por email al Empleado o Manager según corresponda.
- **RN05:** Todo usuario de la aplicación dispondrá de un identificador único, llamado matrícula.
- **RN06:** Toda incidencia tendrá un identificador único.
- **RN07:** Un Empleado no podrá realizar más de 80 horas de trabajos programados en un año natural [19].
- **RN08:** Los festivos de una guardia deberán quedar registrados a fin de poder llevar un seguimiento.
- **RN09:** El acceso a las funcionalidades de la aplicación debe basarse en roles de usuario.
- **RN10:** Las guardias y los tiempos de trabajo deben cumplir con las normativas laborales vigentes.
- **RN11:** La remuneración de los trabajos programados estará determinada según el nivel del empleado, aplicando tarifas diferenciadas para cada nivel.
- **RN12:** Las guardias tendrán una duración de una semana e incluirán una remuneración base fija, a la que se sumará un importe adicional por cada hora trabajada. Además, los días festivos durante la guardia se remunerarán con un incremento específico, todo lo anterior ajustado al nivel del empleado.

3.3. Casos de uso

Los casos de uso son ampliamente utilizados en el diseño y desarrollo de software para describir la interacción entre el usuario y el sistema a fin de conseguir un objetivo específico. Estos actores pueden ser personas, sistemas externos o dispositivos, aunque para el caso de nuestra aplicación serán personas. A lo largo de esta sección identificaremos los casos de uso, y describiremos su funcionamiento.

Los casos de uso son fundamentales por varias razones:

- **Claridad en los requisitos:** Proporcionan una descripción precisa de las funcionalidades que el sistema debe ofrecer, ayudando a los desarrolladores a entender las necesidades de los usuarios [20].
- **Comunicación efectiva:** Facilitan la comunicación entre los equipos de desarrollo y pruebas y los clientes, al establecer un lenguaje común sobre cómo se espera que el sistema se comporte en situaciones reales [21].
- **Diseño centrado en el usuario:** Al basarse en las interacciones del usuario con el sistema, aseguran que el software sea intuitivo y esté alineado con las necesidades de los usuarios finales [22].

Los casos de uso cuentan con los siguientes elementos clave:

- **Actores:** Son los agentes externos que interactúan con el sistema, como personas, otros sistemas o dispositivos. Los actores pueden ser primarios, quienes inician la interacción, o secundarios, que participan o se ven afectados por el caso de uso.
- **Descripción:** Proporciona un resumen del caso de uso, explicando el objetivo general que se busca alcanzar y describiendo brevemente cómo los actores interactúan con el sistema para lograrlo.
- **Secuencia normal:** Consiste en la secuencia de pasos más común que los actores y el sistema siguen para alcanzar el objetivo del caso de uso, incluyendo acciones específicas de los actores y las respuestas del sistema. Se describe de manera cronológica y sin desviaciones.
- **Flujos alternativos:** Detallan las excepciones o variaciones que pueden surgir durante la ejecución del caso de uso. Representan caminos alternativos que se toman si ocurre una condición especial o un error.
- **Precondiciones:** Son los requisitos o condiciones que deben cumplirse antes de que el caso de uso pueda ejecutarse. Describen el estado del sistema o del entorno previo al inicio del caso.
- **Postcondiciones:** Describen el estado del sistema una vez que el caso de uso ha finalizado, ya sea de manera exitosa o no. Indican los cambios resultantes después de la ejecución.

Los casos de uso no solo ayudan en la fase de diseño, sino que además sirven como base para las pruebas y validación del sistema [23]. Son un recurso valioso que asegura que todas las partes interesadas tengan la misma perspectiva del comportamiento esperado del software.

A lo largo de esta sección, se presentarán un diagrama de casos de uso y las descripciones detalladas de estos a fin de tener una visión completa del funcionamiento del sistema y como interactuar con él.

3.3.1. Diagrama de flujo de una incidencia

El diagrama de flujo de una incidencia describe el ciclo completo de gestión de una incidencia, desde su creación hasta su finalización, mostrando todas las acciones que se deben realizar en cada etapa del proceso así como las interacciones entre los diferentes actores del sistema. Este diagrama proporciona una visión clara y estructurada del flujo de trabajo, facilitando su comprensión y ayudando a identificar posibles mejoras en el proceso.

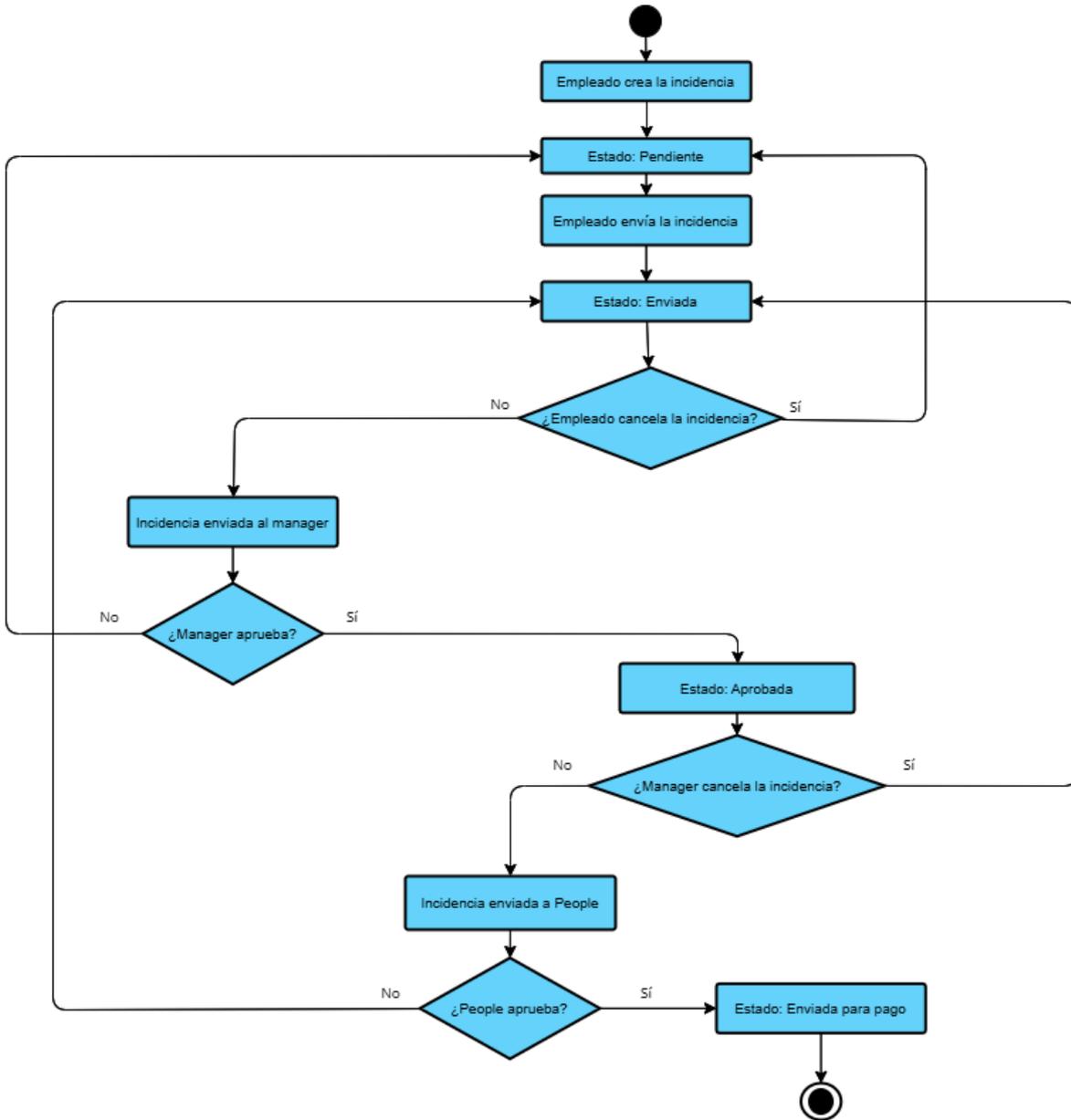


Figura 3.1: Diagrama de flujo de una incidencia

3.3.2. Diagrama de casos de uso

El diagrama de casos de uso es una representación visual utilizada en el modelado de sistemas para describir las interacciones entre los actores y las funcionalidades principales del sistema. Muestra qué acciones pueden realizar los actores en el sistema a través de los *Casos de uso*. El diagrama nos da una visión clara y estructurada de cómo los diferentes usuarios, ya vistos en la sección 3.1, pueden aprovechar las funcionalidades del sistema y resulta esencial para la planificación y diseño en etapas tempranas de desarrollo.

En nuestro caso particular veremos que el *Empleado* va a ser el encargado de crear las incidencias y editarlas, mientras que los *Managers* y *Personal de People* tendrán la responsabilidad de ir aprobando

la solicitud del Empleado y tendrán la capacidad de obtener informes. Dado que las funcionalidades de identificarse y consultar los detalles de una incidencia son funcionalidades comunes a los 3 tipo de actor, estas se han agrupado en el actor: usuario general de la app.

Visual Paradigm Standard(migue(Universidad de Valladolid))

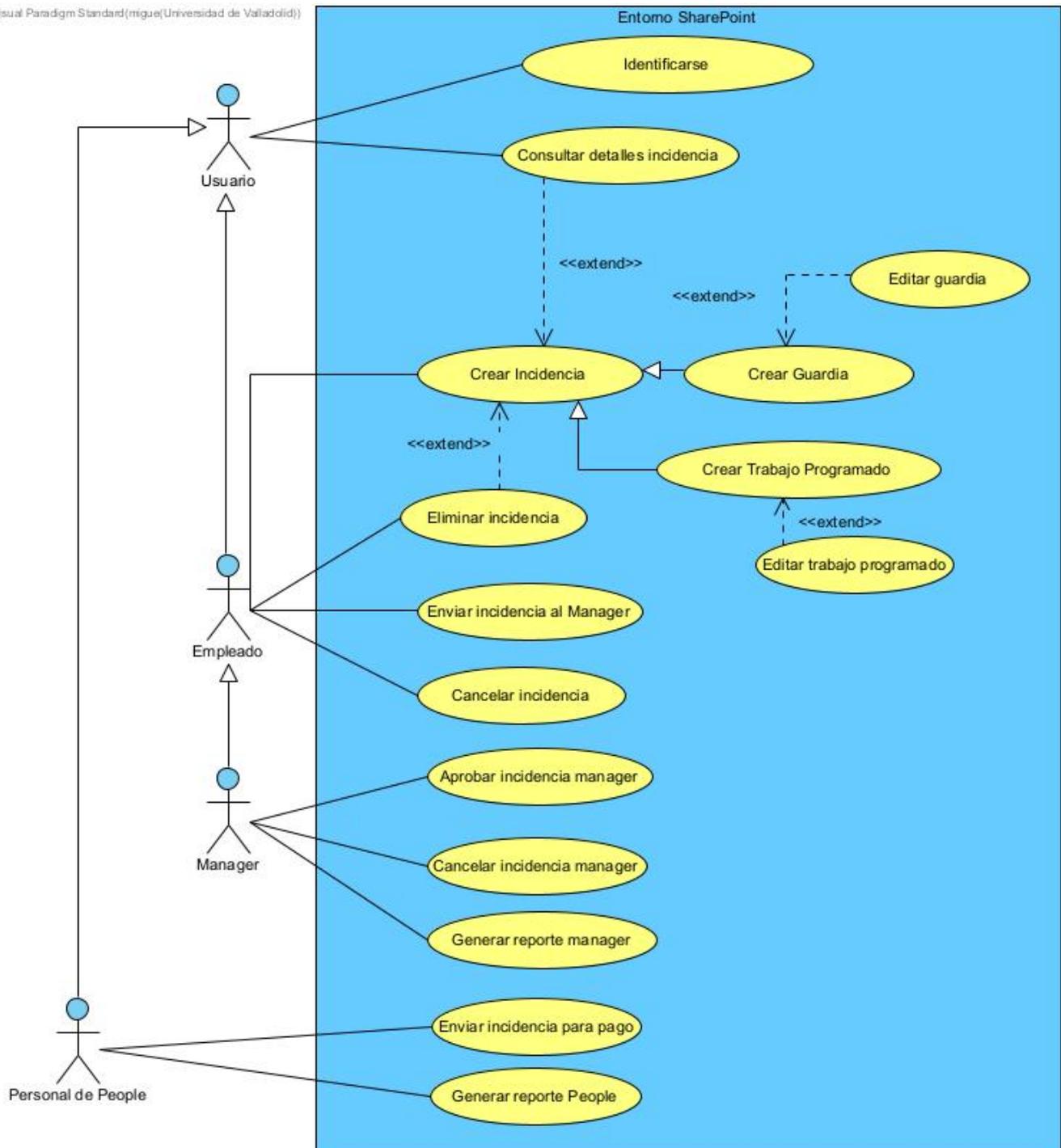


Figura 3.2: Diagrama de casos de uso

3.3.3. Descripción de los casos de uso

Caso de Uso 01: Identificarse	
Actor	Usuario
Descripción	El usuario se identifica en el sistema mediante su email. Es el paso previo antes de acceder a la funcionalidad de la aplicación.
Precondición	El usuario debe estar registrado en el sistema con un email válido.
Postcondición	El sistema autentica al usuario y le permite acceder a sus funcionalidades personalizadas.
Secuencia normal	<ol style="list-style-type: none"> 1. El usuario introduce su email. 2. El sistema valida que el email existe y envía un correo al usuario para que acepte el inicio de sesión. 3. El usuario acepta el inicio de sesión en su email. 4. El sistema lleva al usuario a la pantalla correspondiente a su rol.
Flujos alternativos	<ol style="list-style-type: none"> 1a. Si el email es incorrecto o no existe, el sistema muestra un mensaje de error y solicita al usuario que vuelva a introducirlo. 3a. Si el usuario rechaza el inicio de sesión en su correo, el caso de uso queda sin efecto.

Tabla 3.1: Caso de Uso 01: Identificarse

Caso de Uso 02: Consultar detalles incidencia	
Actor	Usuario de la aplicación
Descripción	Tanto el Personal de People, como los Managers como los Empleados, pueden consultar los detalles de las incidencias que aparezcan en su pantalla.
Precondición	El usuario está identificado en el sistema. Existe una o más incidencias en la pantalla del usuario
Postcondición	El sistema muestra los detalles de la incidencia seleccionada.

Secuencia normal	<ol style="list-style-type: none"> 1. El actor hace click en el botón de <i>Detalles</i> de la incidencia que desea consultar 2. El sistema muestra los detalles de la incidencia seleccionada 3. Si la incidencia es de tipo <i>Guardia</i> el actor puede hacer click en el botón <i>Intervenciones</i>. 4. El sistema muestra las intervenciones. 5. El actor hace click en el botón de cerrar o fuera de la pantalla de intervenciones. 6. El sistema vuelve a mostrar los detalles de la incidencia. 7. El actor hace click en el icono de volver. 8. El sistema muestra la pantalla del rol del actor.
Flujos alternativos	<ol style="list-style-type: none"> 2a. Si la incidencia es de tipo <i>Trabajo programado</i> o el usuario es Personal de People se va al paso 5. 3a. Si el actor hace click fuera del menú de detalles, el caso de uso queda sin efecto.

Tabla 3.2: Caso de Uso 02: Consultar detalles incidencia.

Caso de Uso 03: Crear Guardia	
Actor	Empleado
Descripción	Crear una guardia es una de las funcionalidades principales de la aplicación, solo los Empleados pueden crearlas.
Precondición	El Empleado está identificado en el sistema.
Postcondición	La nueva guardia se ha registrado correctamente en el sistema, con el estado <i>pendiente</i> .

<p>Secuencia normal</p>	<ol style="list-style-type: none"> 1. El Empleado hace click en el botón de <i>Nuevo registro</i> 2. El sistema muestra la pantalla de nuevo registro. 3. El Empleado selecciona <i>Guardia</i> en el desplegable de tipo de registro. 4. El sistema muestra los campos específicos a rellenar de la guardia. 5. El Empleado introduce los datos en los campos de <i>Fecha de inicio, Festivos, Intervenciones y Comentarios</i>. 6. El sistema calcula el número de horas de la guardia. 7. El Empleado hace click en <i>Guardar</i>. 8. El sistema registra la guardia, cierra la pantalla de creación y lleva al Empleado a su pantalla principal.
<p>Flujos alternativos</p>	<p>3a., 5a Si el Empleado hace click fuera de la pantalla, el caso de uso queda sin efecto.</p> <p>7a. Si alguno de los campos está vacío, el sistema notifica al Empleado que no puede haber campos en blanco y cierra el registro.</p> <p>FA1. Si el Empleado en cualquier momento pulsa <i>Cancelar</i>, el caso de uso queda sin efecto.</p>

Tabla 3.3: Caso de Uso 03: Crear Guardia

<p>Caso de Uso 04: Crear Trabajo Programado</p>	
<p>Actor</p>	<p>Empleado</p>
<p>Descripción</p>	<p>Crear un trabajo programado es una de las funcionalidades principales de la aplicación, solo los Empleados pueden crearlos.</p>
<p>Precondición</p>	<p>El Empleado está identificado en el sistema.</p>
<p>Postcondición</p>	<p>El nuevo trabajo programado se ha registrado en el sistema en el estado <i>pendiente</i>.</p>

Secuencia normal	<ol style="list-style-type: none"> 1. El Empleado hace click en el botón de <i>Nuevo registro</i>. 2. El sistema muestra la pantalla de nuevo registro. 3. El Empleado selecciona <i>Trabajo programado</i> en el desplegable de tipo de registro. 4. El sistema muestra los campos específicos a rellenar del trabajo programado. 5. El Empleado introduce los datos en los campos <i>Fecha, Hora inicio, Hora fin y Comentarios</i>. 6. El sistema calcula el número de horas del trabajo programado. 7. El usuario hace click en <i>Guardar</i>. 8. El sistema registra el trabajo programado, cierra la pantalla de creación y lleva al Empleado a su pantalla principal.
Flujos alternativos	<p>7a. Si alguno de los campos está vacío, el sistema notifica al Empleado que no puede haber campos en blanco y cierra el registro.</p> <p>FA1. Si el Empleado en cualquier momento pulsa <i>Cancelar</i> o hace click fuera del menú de nuevo registro, el caso de uso queda sin efecto.</p>

Tabla 3.4: Caso de Uso 04: Crear trabajo programado

Caso de Uso 05: Editar guardia	
Actor	Empleado
Descripción	Una vez que el Empleado ha creado una guardia y todavía no la ha enviado, puede editar sus campos y guardar los cambios.
Precondición	El Empleado está identificado en el sistema. Existe un registro creado de tipo guardia. La guardia no ha sido enviada.
Postcondición	Los cambios realizados en los campos de la guardia quedan registrados en el sistema.

<p>Secuencia normal</p>	<ol style="list-style-type: none"> 1. El Empleado hace click en una guardia no enviada. 2. El sistema muestra el formulario de edición. 3. El Empleado modifica los campos de la guardia. 4. El sistema recalcula las horas totales. 5. El Empleado hace click en <i>Guardar</i>. 6. El sistema registra los cambios, cierra el formulario de edición y lleva al Empleado a la pantalla principal.
<p>Flujos alternativos</p>	<p>5a. Si alguno de los campos está vacío, el sistema notifica al Empleado que no puede haber campos en blanco y cierra el formulario.</p> <p>FA1. Si el Empleado en cualquier momento pulsa <i>Cancelar</i> o hace click fuera del menú de edición, el caso de uso queda sin efecto.</p>

Tabla 3.5: Caso de Uso 05: Editar guardia

El caso de uso referente a la edición de trabajos programados es análogo al de la edición de una guardia.

<p>Caso de Uso 06: Enviar incidencias al Manager</p>	
<p>Actor</p>	<p>Empleado</p>
<p>Descripción</p>	<p>El Empleado puede enviar registros al Manager para su aprobación. Estos registros pueden ser tanto guardias como trabajos programados.</p>
<p>Precondición</p>	<p>El Empleado está identificado en el sistema. Existe una o más incidencias creadas. Ninguna de las incidencias seleccionadas está en el estado <i>Enviada</i>.</p>
<p>Postcondición</p>	<p>Las incidencias seleccionadas pasan al estado <i>Enviada</i> y pasan a aparecer en la pantalla del Manager.</p>

Secuencia normal	<ol style="list-style-type: none"> 1. El Empleado selecciona una o más incidencias para enviar. 2. El sistema hace visible el botón de <i>Enviar</i>. 3. El usuario hace click en el botón de <i>Enviar</i>. 4. El sistema muestra una pantalla de confirmación. 5. El Empleado hace click en <i>Confirmar</i>. 6. El sistema registra el estado de la incidencia como <i>Enviada</i> y notifica al Manager por correo electrónico el envío de la incidencia.
Flujos alternativos	5a. Si el Empleado pulsa <i>Cancelar</i> , o hace click fuera de la pantalla de confirmación, el caso de uso queda sin efecto.

Tabla 3.6: Caso de Uso 06: Enviar incidencias al Manager

Caso de Uso 07: Eliminar incidencia	
Actor	Empleado
Descripción	El Empleado puede eliminar registros que haya creado, siempre y cuando estos no hayan sido enviados a su Manager.
Precondición	<p>El Empleado está identificado en el sistema.</p> <p>Existe uno o más registros creados.</p> <p>Ninguna de las incidencias seleccionadas está en el estado <i>Enviada</i>.</p> <p>Ninguna de las guardias a eliminar tiene intervenciones asociadas.</p>
Postcondición	Los registros eliminados desaparecen de la pantalla del Empleado, y son borrados del sistema.

Secuencia normal	<ol style="list-style-type: none"> 1. El Empleado selecciona una o más incidencias para borrar. 2. El sistema hace visible el icono de borrar, representado con una papelera. 3. El usuario hace click en el icono de borrar. 4. El sistema muestra una pantalla de confirmación. 5. El Empleado hace click en <i>Confirmar</i>. 6. El sistema elimina la incidencia de forma permanente.
Flujos alternativos	<ol style="list-style-type: none"> 5a. Si el Empleado pulsa <i>Cancelar</i> o hace click fuera de la pantalla de confirmación, el caso de uso queda sin efecto.

Tabla 3.7: Caso de Uso 07: Eliminar incidencia

Caso de Uso 08: Cancelar incidencia	
Actor	Empleado
Descripción	Un Empleado puede cancelar los registros que hayan sido enviados a su Manager para ser aprobados. Estos pasarán al estado <i>Pendiente</i> .
Precondición	<p>El Empleado está identificado en el sistema.</p> <p>Existe uno o más registros creados.</p> <p>Todas las incidencias seleccionadas están en el estado <i>Enviada</i>.</p>
Postcondición	Las incidencias canceladas pasan al estado <i>Pendiente</i> y el Manager recibe un correo informándole que la incidencia ha sido cancelada.
Secuencia normal	<ol style="list-style-type: none"> 1. El Empleado selecciona una o más incidencias enviadas para cancelarlas. 2. El sistema muestra el botón <i>Cancelar</i>. 3. El usuario hace click en el botón <i>Cancelar</i>. 4. El sistema muestra la pantalla de confirmación. 5. El Empleado hace click en <i>Confirmar</i>. 6. El sistema registra el estado de la incidencia como <i>Pendiente</i> y notifica al Manager por correo electrónico de la cancelación de la incidencia.

Flujos alternativos	5a. Si el Empleado pulsa <i>Cancelar</i> o hace click fuera de la pantalla de confirmación, se va al paso 2.
----------------------------	--

Tabla 3.8: Caso de Uso 08: Cancelar incidencia

Caso de Uso 09: Aprobar incidencia Manager	
Actor	Manager
Descripción	El Manager puede aprobar o rechazar incidencias que le hayan enviado los Empleados.
Precondición	El Manager está identificado en el sistema. Existe una o más incidencias en la pantalla principal del Manager en estado <i>Enviada</i> .
Postcondición	Si el Manager aprueba la incidencia esta pasa al estado de <i>Aprobada</i> , si la rechaza, la incidencia pasa al estado <i>Pendiente</i> . En ambos casos se notifica al Empleado.
Secuencia normal	<ol style="list-style-type: none"> 1. El Manager selecciona una o más incidencias en estado <i>Enviada</i>. 2. El sistema hace visibles los botones de <i>Aprobar</i> y <i>Rechazar</i>. 3. El Manager hace click en el botón de <i>Aprobar</i>. 4. El sistema muestra un menú de confirmación en el que se permite añadir un comentario. 5. El Manager introduce un comentario si procede y hace click en <i>Confirmar</i>. 6. El sistema registra el nuevo estado, el comentario si existiera y devuelve al Manager a su pantalla principal. También notifica al Empleado que emitió esa incidencia.
Flujos alternativos	3a. El Manager hace click en <i>Rechazar</i> . Se va al paso 4. 5a El Manager hace click en el botón <i>Cancelar</i> o fuera del menú de confirmación. Se va al paso 2.

Tabla 3.9: Caso de Uso 09: Aprobar incidencia Manager

En esta ocasión, debido a que aprobar y rechazar son casos análogos se ha optado por mostrarlos en el mismo caso de uso.

Caso de Uso 10: Cancelar incidencia Manager	
Actor	Manager
Descripción	El Manager puede cancelar incidencias que hayan sido aprobadas y enviadas al Personal de People. Al cancelarlos pasarán al estado <i>Enviada</i> . También se notificará al Empleado que la emitió que se ha cancelado su incidencia.
Precondición	El Manager está identificado en el sistema. Existen una o más incidencias con el estado <i>Aprobada</i> .
Postcondición	La incidencia pasa a estar en el estado <i>Enviada</i> y el emisor recibe una notificación.
Secuencia normal	<ol style="list-style-type: none"> 1. El Manager selecciona una o más incidencias que tengan de estado <i>Aprobada</i>. 2. El sistema hace visible el botón de <i>Cancelar</i>. 3. El Manager hace click en el botón de <i>Cancelar</i>. 4. El sistema muestra el menú de confirmación. 5. El Manager hace click en <i>Confirmar</i> 6. El sistema registra la incidencia con el estado <i>Enviada</i> y notifica al Empleado emisor que su incidencia ha sido cancelada por su Manager.
Flujos alternativos	5a. Si el Manager hace click en <i>Cancelar</i> o fuera del menú de confirmación, se va al paso 2.

Tabla 3.10: Caso de Uso 10: Cancelar incidencia Manager

Caso de Uso 11: Generar reporte Manager	
Actor	Manager
Descripción	Los Managers pueden crear reportes personalizados indicando un periodo. El reporte generado será enviado por correo en formato .csv al Manager.

Precondición	El Manager está identificado en el sistema. Existen registros en la pantalla principal del Manager.
Postcondición	El Manager recibe el informe por correo.
Secuencia normal	<ol style="list-style-type: none"> 1. El Manager hace click en el botón <i>Generar reporte</i>. 2. El sistema abre el menú de selección de fechas. 3. El Manager introduce la fecha de inicio y fin del periodo que desea consultar y hace click en confirmar. 4. El sistema genera un informe con las incidencias de el periodo seleccionado y se lo envía al Manager por correo.
Flujos alternativos	<ol style="list-style-type: none"> 3a. Si el Manager hace click en el icono de cancelar, o fuera del menú, el caso de uso queda sin efecto. 3b. Si el Manager no introduce o la fecha de inicio o la de fin, el sistema notifica que se deben escoger las fechas.

Tabla 3.11: Caso de Uso 11: Generar reporte Manager

Caso de Uso 12: Generar reporte People	
Actor	Personal de People
Descripción	El Personal de People tiene la capacidad de recibir reportes por correo con todas las incidencias de su panel.
Precondición	El Personal de People está identificado en el sistema. Existen incidencias en los estados <i>Enviada para pago o Aprobada</i>
Postcondición	Se recibe un correo con el informe en formato .csv.
Secuencia normal	<ol style="list-style-type: none"> 1. El actor hace click en el botón <i>Generar reporte</i> 2. El sistema ejecuta un flujo y envía un reporte al actor en el que aparecen todas las incidencias en los estados <i>Aprobada y enviada para pago</i>.
Flujos alternativos	No hay

Tabla 3.12: Caso de Uso 12: Generar reporte People

Caso de Uso 13: Enviar incidencia para pago	
Actor	Personal de People
Descripción	El Personal de People es el encargado de enviar para pago o de rechazar todas las incidencias aprobadas por los Managers. Si tras realizar esta acción, tanto el Manager como el Empleado emisor, reciben una notificación por correo.
Precondición	El Personal de People está identificado en el sistema. Existen una o más incidencias en el estado <i>Aprobada</i> .
Postcondición	Las incidencias pasan al estado <i>Enviada para pago o Enviada</i> y tanto el Manager como el emisor son notificados.
Secuencia normal	<ol style="list-style-type: none"> 1. El actor selecciona una o más incidencias que tengan el estado <i>Aprobada</i>. 2. El sistema hace visibles los botones de <i>Enviar y Rechazar</i>. 3. El actor hace click en el botón de <i>Enviar</i>. 4. El sistema muestra el menú de confirmación. 5. El actor hace click en <i>Confirmar</i> 6. El sistema actualiza el estado de la incidencia como <i>Enviada para pago</i>, notifica al Manager y al Empleado por correo electrónico el nuevo estado de la incidencia y devuelve al actor a su pantalla principal.
Flujos alternativos	<p>3a. El actor hace click en el botón de Rechazar. El sistema muestra el menú de confirmación. El actor introduce un comentario si fuera necesario y hace click en <i>Confirmar</i>. El sistema actualiza el estado de la incidencia como <i>Enviada</i>, registra el comentario si lo hubiera y notifica tanto al Manager como al emisor.</p> <p>5a. Si el actor hace click en <i>Cancelar</i> o fuera del menú de confirmación, se va al paso 2.</p>

Tabla 3.13: Caso de Uso 13: Enviar incidencia para pago

3.4. Diagramas de actividad

Los diagramas de actividad son una herramienta gráfica que permite modelar el flujo de trabajo de un proceso, representando la secuencia de actividades, decisiones y eventos que ocurren dentro de este.

Cada diagrama muestra como las actividades están conectadas entre sí y van variando su estado. De esta forma se pueden identificar las interacciones del usuario con el sistema de forma clara.

Estos diagramas son útiles para:

- Visualizar procesos de forma clara y estructurada.
- Identificar puntos de decisión o bifurcaciones.
- Detectar cuellos de botella o áreas de mejora del sistema.

En esta sección se presentan los principales diagramas de actividad que ilustran los casos de uso más relevantes de la aplicación, omitiendo los muy similares entre sí.

■ **Identificarse:**

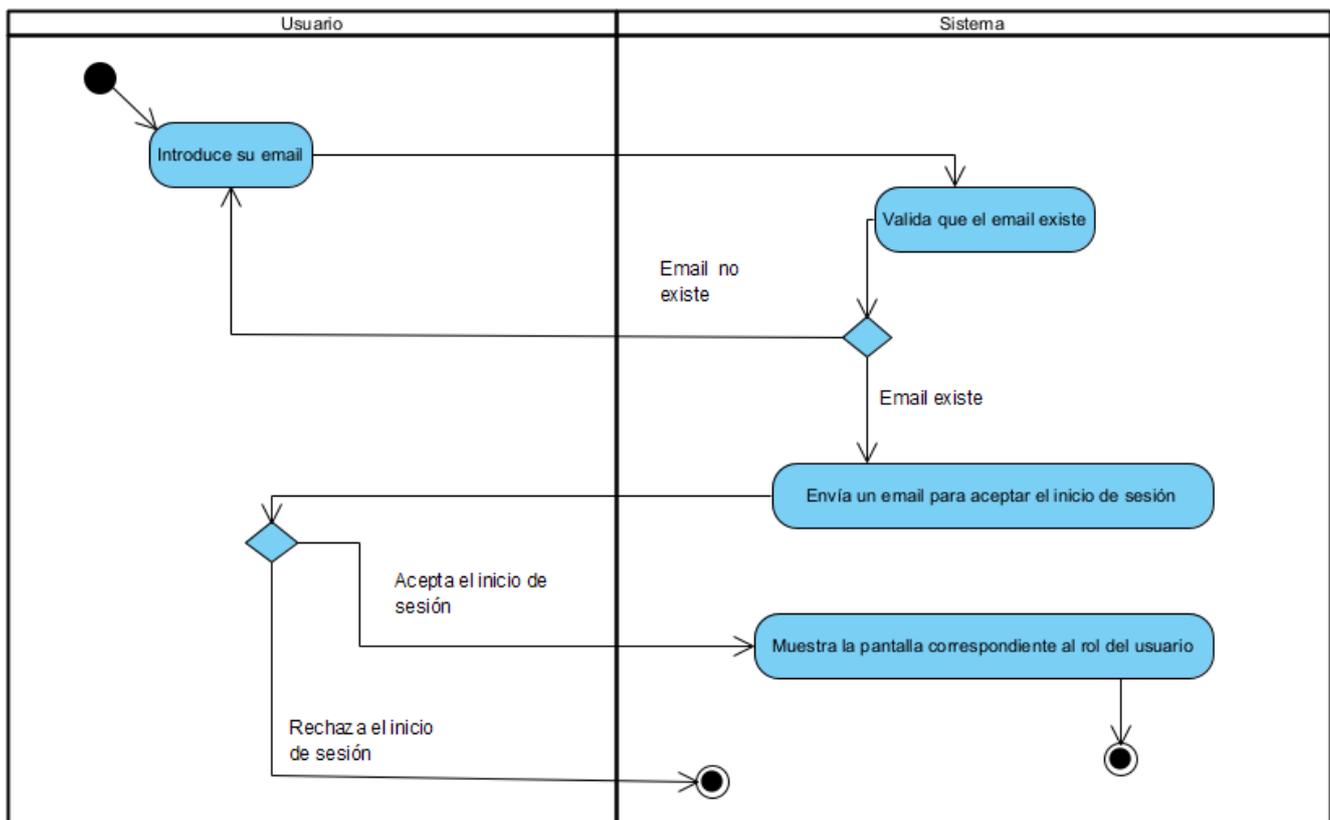


Figura 3.3: Diagrama de actividad de identificarse

- **Crear incidencia:** dado que este caso de uso es similar para guardias y trabajos, se muestra un diagrama de actividad para una incidencia genérica.

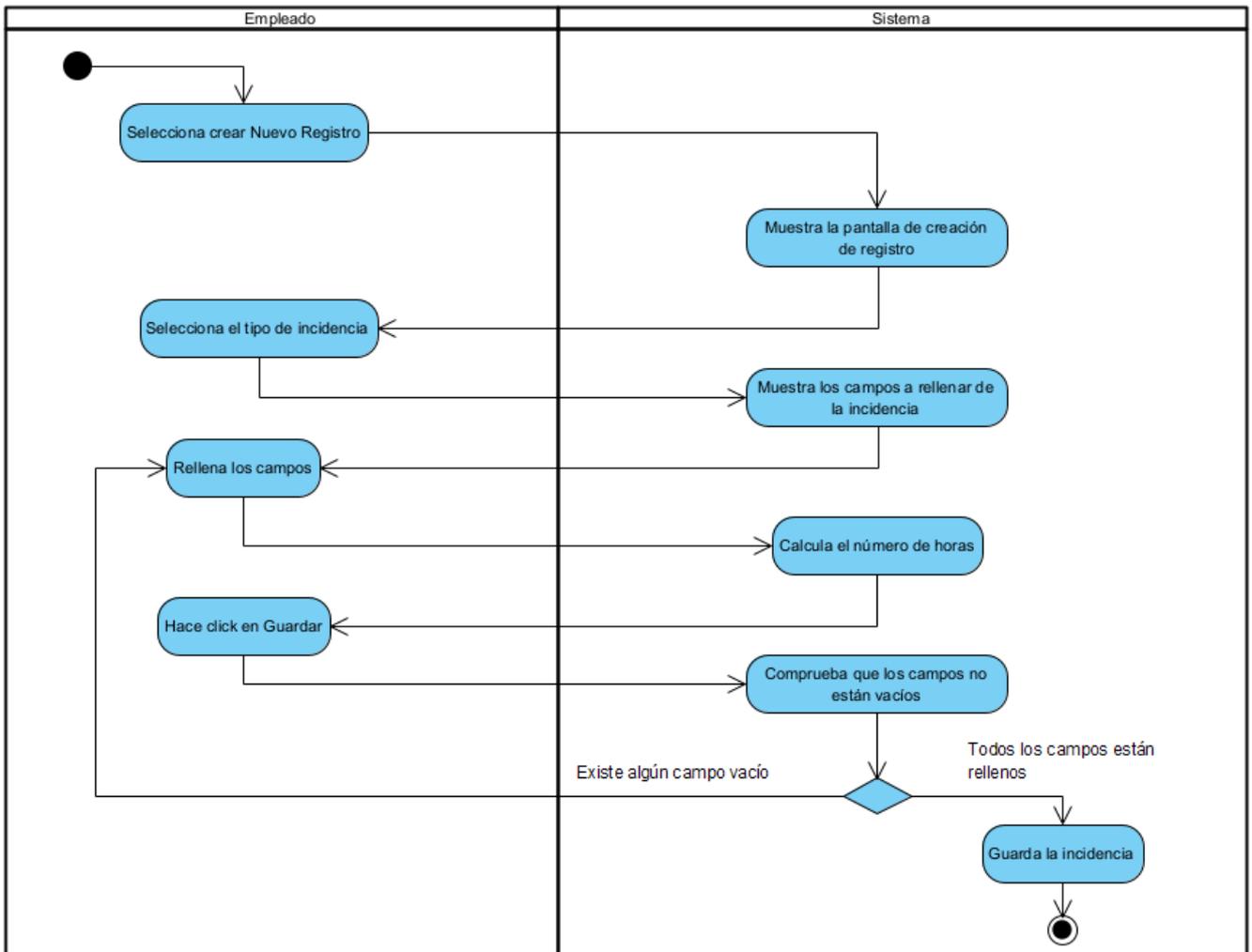


Figura 3.4: Diagrama de actividad de crear incidencia

■ Eliminar incidencia:

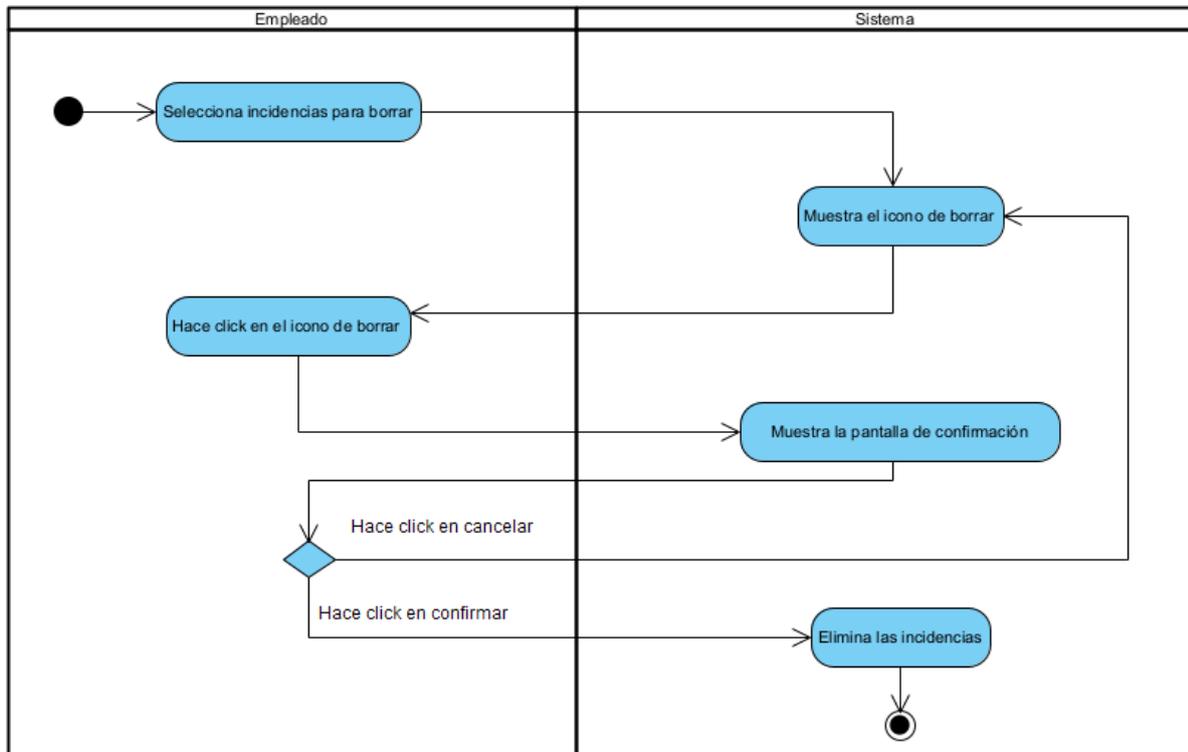


Figura 3.5: Diagrama de actividad de eliminar una incidencia

■ Generar reporte Manager:

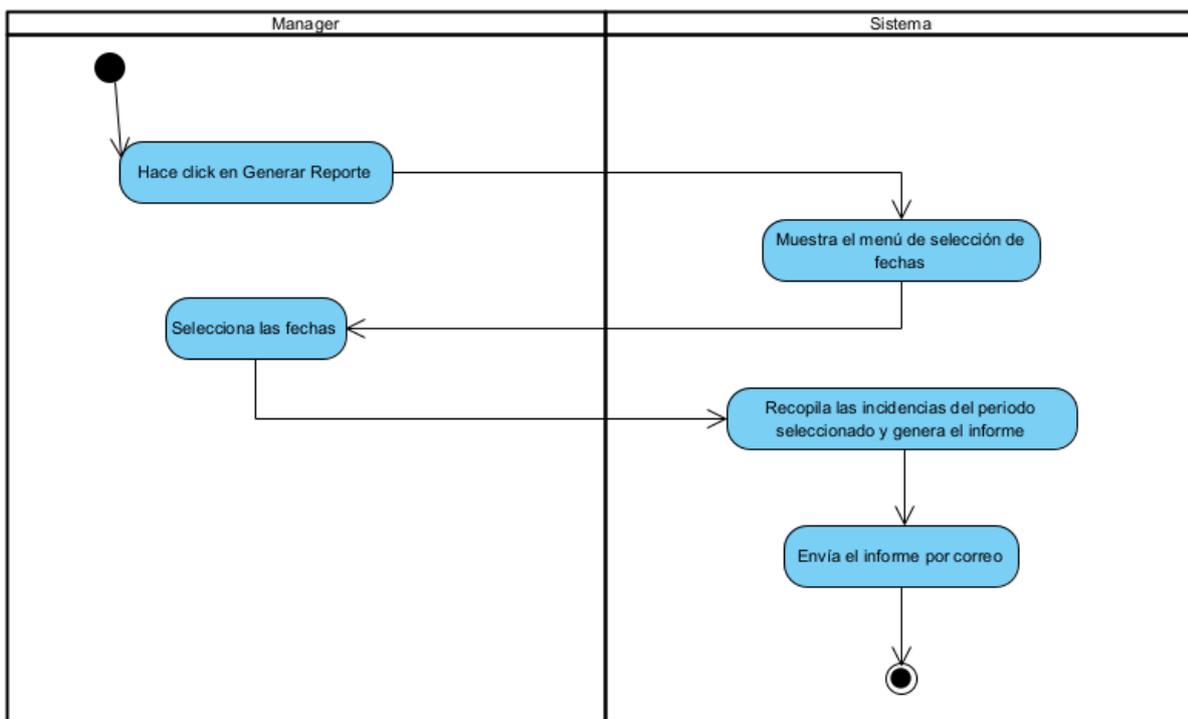


Figura 3.6: Diagrama de actividad de generar reporte Manager

- Aprobar o rechazar una incidencia Manager:** en este diagrama, se muestra el flujo de un Manager para aprobar o rechazar una incidencia que ha sido enviada por un Empleado.

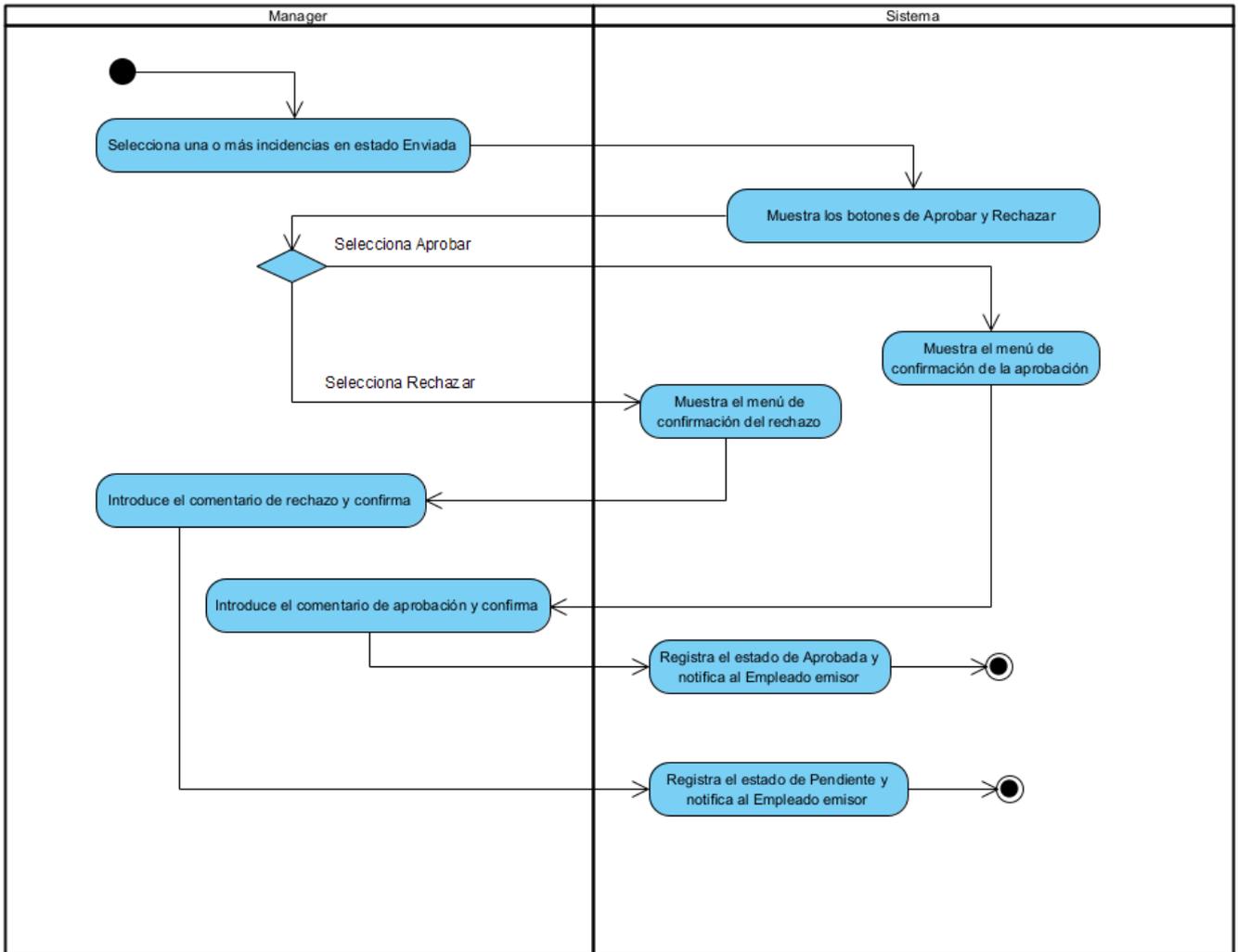


Figura 3.7: Diagrama de actividad de aprobar o rechazar una incidencia Manager

3.5. Modelo de dominio

Un *modelo de dominio* es una representación conceptual que describe las principales entidades, atributos y relaciones en un sistema. Este modelo facilita la comprensión del problema y sirve como base para el diseño del software [22].

En nuestro caso, disponemos de unas incidencias que pueden ser guardias o trabajos programados, las cuales comparten una serie de atributos comunes, pero también unos atributos propios. Cada guardia puede tener una serie de intervenciones realizadas durante su transcurso. Los usuarios de la aplicación pueden ser tanto Personal de People, como Managers o Empleados, los cuales disponen de atributos comunes como se observa en la generalización, salvo en el caso del Empleado, que además dispone de un Manager, representado en la relación Empleado-Manager. Finalmente se han definido una serie de enumerados para modelar los tipos de incidencia, su estado y el rol de los usuarios.

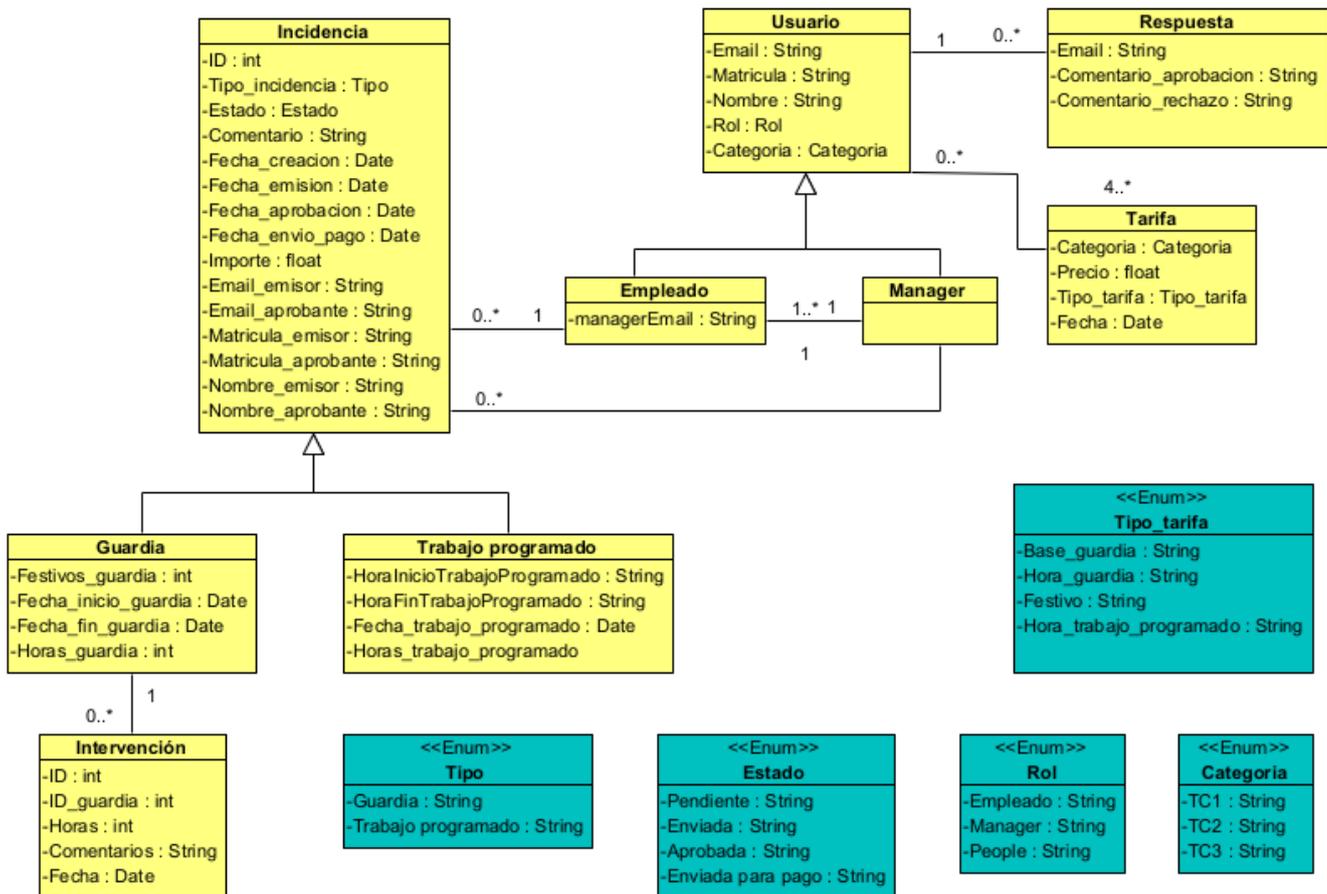


Figura 3.8: Modelo de dominio

Capítulo 4

Diseño

La fase de diseño resulta crucial en el desarrollo de software, dado que traduce los requisitos elaborados en la fase de análisis 3.2 a una estructura técnica detallada que guiará la implementación del sistema. A lo largo de este proceso se tomarán decisiones sobre la arquitectura, y los componentes, asegurando de esta forma que el sistema cumpla con los objetivos y necesidades definidos anteriormente. En el diseño no debemos hacer hincapié solamente en la funcionalidad, sino que tenemos que tener en cuenta aspectos esenciales como son la usabilidad y la accesibilidad. De esta forma nuestra aplicación será intuitiva y permitirá que todos los perfiles de usuario la puedan utilizar sin problemas, independientemente de sus limitaciones individuales.

En este caso la aplicación está desarrollada mediante PowerApps. Esto añade consideraciones específicas. PowerApps, como se explicará a continuación en la sección 5.1, es una plataforma que permite construir soluciones low-code que se integran en ecosistemas como SharePoint. Dado que esta aplicación se integrará en SharePoint, el diseño debe contemplar:

- Compatibilidad con los componentes de SharePoint, como listas, bibliotecas y flujos de Power Automate.
- El cumplimiento de las reglas de negocio y los procesos definidos en la organización.
- La adaptabilidad a los dispositivos móviles, ya que PowerApps ofrece aplicaciones para distintos entornos.

Adicionalmente, PowerApps trabaja con un lenguaje específico, Power Fx [24], por lo que en esta fase tenemos que considerar las limitaciones técnicas de la plataforma, como el rendimiento, la correcta gestión de los datos y la integración con servicios como Power Automate, si fuera necesario.

Por último, el diseño de la aplicación debe ajustarse en SharePoint a las políticas de seguridad y permisos de la empresa, asegurando de esta forma que los datos sensibles se manejen correctamente.

4.1. Patrones de diseño en PowerApps

Los patrones de diseño en PowerApps [25] son enfoques predefinidos que nos permiten crear aplicaciones eficientes y bien estructuradas. Facilitan la implementación de soluciones comunes como flujos de aprobación, administración de datos o filtrado de información. Al aplicar los patrones conseguimos que las aplicaciones sean coherentes, escalables y facilitamos su mantenimiento, a la par que optimizamos procesos repetitivos y mejoramos la experiencia de usuario, dado que son reutilizables y adaptables a distintos contextos.

En el caso de nuestra aplicación los tres patrones en los que podemos enmarcarla son Aprobación [26], Inspección/Auditoría [27] y administración del ciclo de vida [28].

1. **Aprobación:** el Patrón Aprobación [26] permite gestionar flujos en los que una solicitud pasa por múltiples niveles de aprobación. Se estructura en tres fases principales, la creación, el envío y la aprobación. En muchos casos se integra con Power Automate de forma que se automatice el flujo de trabajo, enviando notificaciones, o registrando las etapas del proceso. Además permite definir diferentes acciones y roles.



Figura 4.1: Patrón Aprobación; tomada de [26]

2. **Inspección/Auditoría:** el Patrón Inspección/Auditoría [27] permite realizar revisiones o evaluaciones de las actividades. Esto permite que los supervisores inspeccionen y auditen el estado de las incidencias. Permite también registrar información clave como fechas, notas y comentarios, facilitando el registro estructurado de datos y el cumplimiento de las tareas.

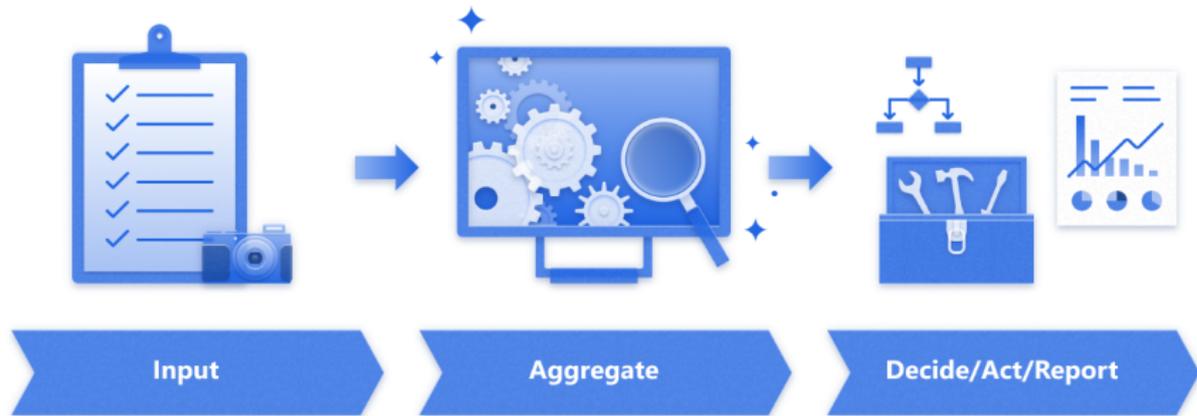


Figura 4.2: Patrón Inspección/Auditoría; tomada de [27]

3. **Administración del ciclo de vida o del flujo de trabajo:** este patrón [28] resulta adecuado para gestionar el ciclo completo de las incidencias en la aplicación. Asegura que cada solicitud pase por fases claramente definidas, como creación, envío, y diferentes niveles de aprobación, incluyendo reglas de negocio para controlar sus estados. PowerApps, de la mano de Power Automate, permite automatizar el flujo, con envíos de notificaciones y actualizaciones de estado en cada paso.

Todos estos patrones pueden ser combinados a fin de adaptarse a las necesidades del proyecto, de forma que sienten unas bases comunes sobre las que trabajar y que faciliten el éxito de este.

4.2. Diseño de la base de datos

El diseño de la base de datos es una fase fundamental para garantizar el correcto funcionamiento y la eficiencia de cualquier aplicación. En este proyecto la base de datos se estructura de una forma que permita gestionar las relaciones entre los diferentes usuarios así como con sus incidencias. Para ello y basándonos en el modelo de dominio descrito anteriormente en la sección 3.5, se ha optado por implementar la base de datos mediante listas de SharePoint, que permiten un enfoque compatible con las herramientas del entorno de Microsoft 365.

Al diseñar la base de datos conseguimos los siguientes objetivos:

- Organizar la información de forma estructurada.
- Garantizar la integridad de los datos.
- Optimizar la gestión de los datos, facilitando de esta forma la creación, modificación y consulta de estos por parte de los usuarios de la aplicación.

En SharePoint, cada lista tiene un funcionamiento similar a una tabla en una base de datos relacional. Las listas están formadas por filas, en la que cada una representa un registro o entrada único, y columnas, que cada una representa un atributo del elemento.

Al usar SharePoint obtenemos una serie de ventajas, dado que los datos son accesibles desde cualquier dispositivo compatible con Microsoft 365, es fácilmente integrable con PowerApps y Power Automate, lo que facilita la creación de flujos de trabajo e interfaces personalizadas, y finalmente favorecen la colaboración, permitiendo que los diferentes actores puedan acceder a los datos.

A través de esta implementación, conseguimos una solución flexible y adaptable a las necesidades del proyecto.

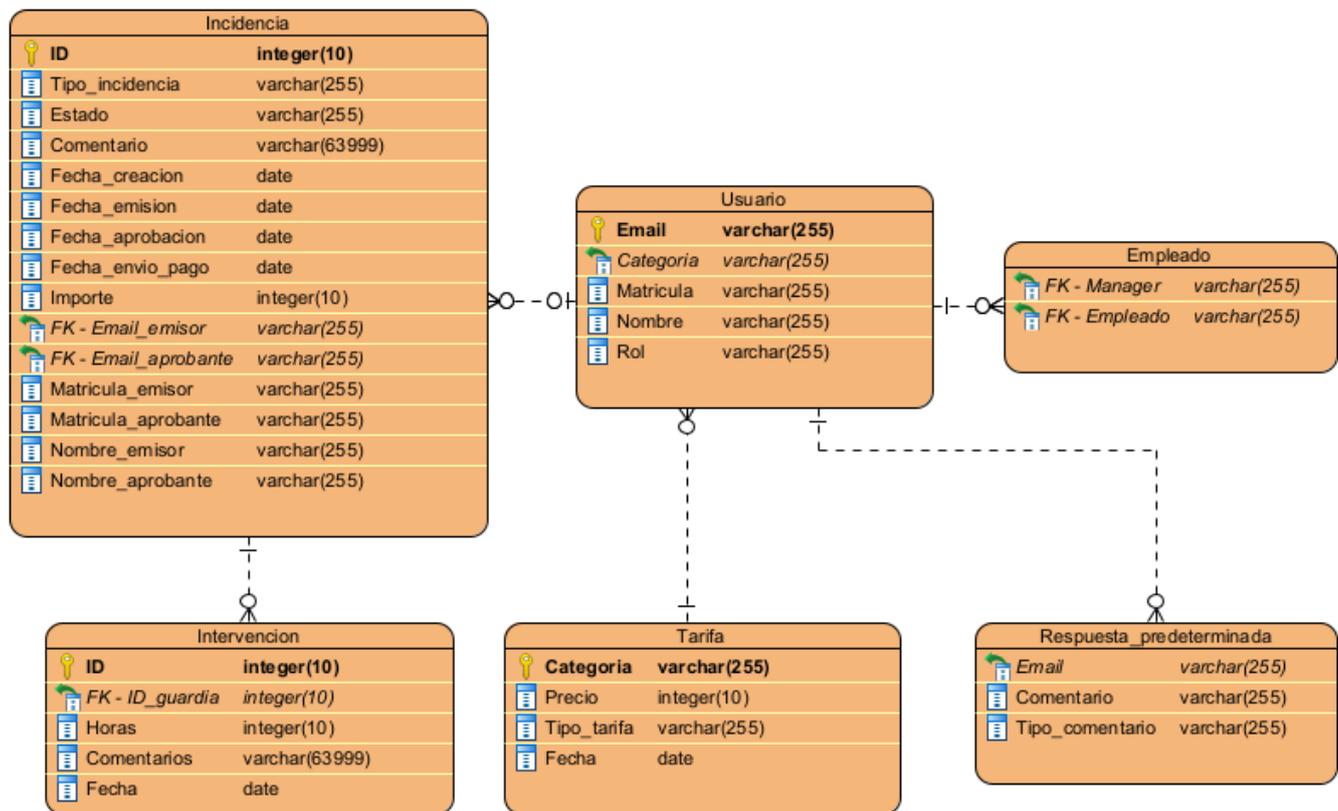


Figura 4.3: Modelo relacional

A continuación se presenta una descripción de las tablas que aparecen en la figura anterior 4.3 con las que va a contar nuestra base de datos:

- **Incidencia:** en esta lista se almacenan todos los datos relevantes de una incidencia, como son su tipo, el estado, las fechas en que se creo y aprobó, o los datos tanto del Empleado que creó la incidencia como del responsable encargado de aprobarla. Cada incidencia cuenta con un identificador único que la hace distinguible del resto.
- **Intervención:** en esta lista podemos encontrar información de las diferentes intervenciones que tienen las incidencias de tipo guardia. Se almacena su identificador único, pero también almacenan el de la guardia a la que pertenecen, además contiene las horas de cada incidencia, la fecha y los comentarios.
- **Usuario:** en esta lista registramos los distintos usuarios que tiene la aplicación, sirve tanto para realizar el *login*, como para obtener los datos de los usuarios de cara a rellenar los campos de las

incidencias. En ella se almacena el email de cada usuario, que resulta un identificador único, así como su matrícula, el nombre, su rol y su contraseña.

- **Empleado:** esta lista se encarga de modelar la relación Manager-Empleado, teniendo en cuenta que cada Empleado puede tener un único Manager, pero un manager puede tener varios Empleados a su cargo. Se almacenan los correos tanto del Empleado como del Manager, que son ambos identificadores únicos en la aplicación.
- **Tarifa:** en esta lista encontramos las tarifas de cada tipo de intervención, se almacena también la fecha en la que se ha registrado de forma que se lleve un histórico y se pueda automatizar el recuperar la tarifa más reciente. Cada tarifa se asocia al usuario por medio de su categoría.
- **Respuesta predeterminada:** esta última lista almacena las respuestas de aprobación y rechazo tanto de los Managers como de People, de forma que las más habituales se puedan recuperar. Se clasifican dependiendo de si son de aprobación o de rechazo.

4.3. Diseño de la interfaz de usuario

La interfaz de usuario o UI, es el punto de interacción entre el usuario y el sistema, por lo que su diseño resulta crucial para el éxito de cualquier aplicación. Una interfaz bien diseñada, no solo facilita la ejecución de tareas, sino que también mejora la experiencia general del usuario, contribuyendo a que la aplicación sea eficiente, intuitiva y satisfactoria.

A lo largo de esta sección, veremos como diseñar una interfaz de usuario, prestando especial atención a los atributos de usabilidad y elaborando bocetos que sirvan como base para la construcción de la UI.

En resumen, el objetivo es lograr una interfaz que no solo sea visualmente atractiva, sino que también optimice la experiencia de usuario, garantizando que la aplicación se use de manera fluida y sin dificultades.

4.3.1. Atributos de usabilidad

La usabilidad de una interfaz se refiere a la facilidad con la que los usuarios pueden aprender a utilizarla y ser eficientes usándola. De acuerdo con la norma **ISO 9241-11** [29], los principales atributos de usabilidad que se han considerado en este proyecto son:

1. **Eficacia:** hace referencia a la capacidad de la interfaz para permitir que los usuarios completen sus tareas correctamente. A través de un diseño eficaz se consigue minimizar errores y facilitar la consecución de las tareas.
2. **Eficiencia:** mide el grado de esfuerzo y tiempo que requiere el usuario para completar las tareas. En nuestro caso el objetivo es tener una interfaz eficiente que reduzca el número de pasos

necesarios para ejecutar una acción, agilizando el proceso e incrementando la productividad.

3. **Satisfacción:** se traduce como el agrado y comodidad que experimentan los usuarios al interactuar con la interfaz. Una UI satisfactoria es fácil de entender, intuitiva y visualmente atractiva.

Además de estos tres atributos principales, se han tenido en cuenta también factores como la facilidad de aprendizaje, una característica clave para que los nuevos usuarios puedan familiarizarse rápidamente con la interfaz, y la facilidad de recuerdo, que evalúa la facilidad para recordar cómo utilizar la aplicación tras un periodo de inactividad [30].

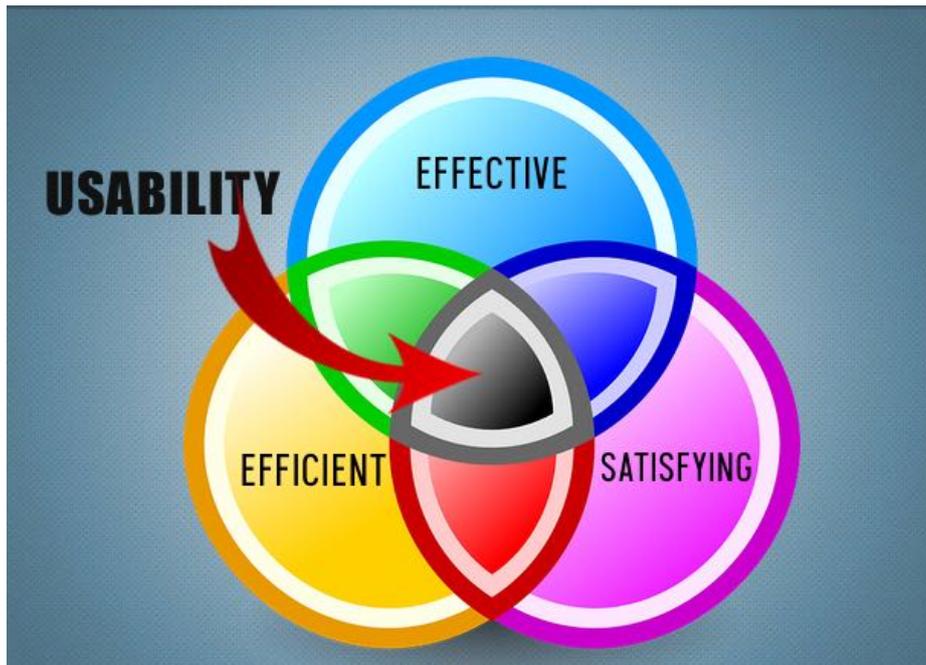


Figura 4.4: Atributos de usabilidad; tomada de [31].

4.3.2. Elaboración de bocetos

El proceso de diseño de la interfaz comenzó con la elaboración de bocetos. Esto permitió visualizar ideas y definir la experiencia de usuario de forma preliminar. Los bocetos son dibujos esquemáticos que permiten visualizar la disposición de los elementos, las jerarquías visuales y los flujos de interacción sin necesidad de implementar código.

Al elaborar bocetos conseguimos poder probar distintas ideas de diseño antes de pasar a la fase de implementación, temporalmente más costosa. Facilitan también un desarrollo más rápido de las iteraciones, al poder realizar cambios y ajustes basados en las primeras impresiones y finalmente, en un caso empresarial real, servirían para favorecer la comunicación entre los desarrolladores del proyecto y los clientes.

En nuestro caso, nos hemos servido de los bocetos para definir una estructura clave de las diferentes pantallas de la aplicación, enfocándonos en que sean simples y claras, y asegurando que los usuarios sean capaces de acceder a la funcionalidad principal en pocos clicks.

Pantalla Login En esta pantalla es donde el usuario, sea cual sea su rol, va a autenticarse, para posteriormente pasar a la pantalla principal de su usuario, su única función es introducir el usuario y la contraseña. Como funcionalidad se planea añadir un icono que permita hacer la contraseña visible o invisible. Esta pantalla por lo tanto va a contar con entradas de texto para el usuario y la contraseña, así como el icono para cambiar el formato de la contraseña y un botón para aceptar.

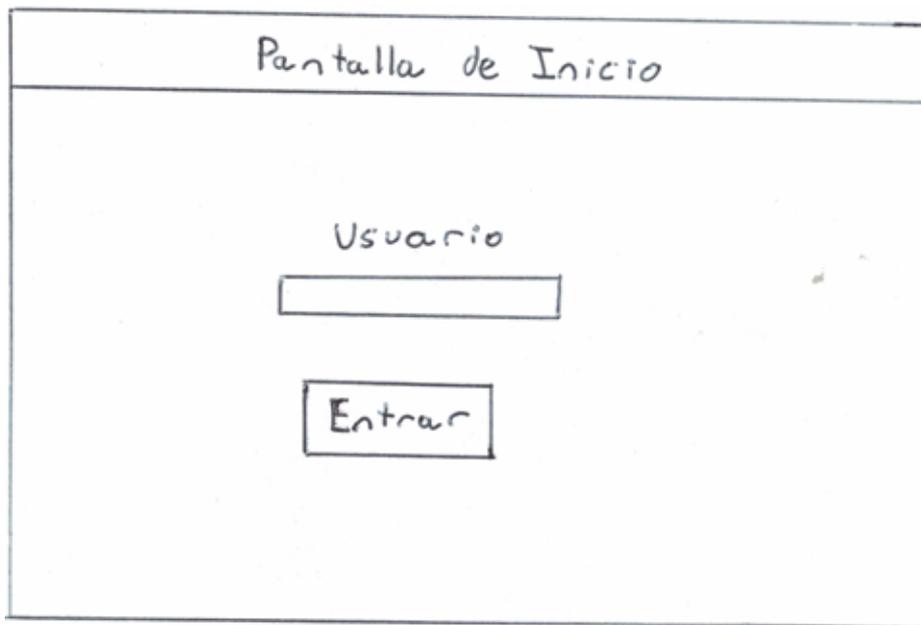


Figura 4.5: Boceto de la Pantalla de Login

Pantalla People

Esta pantalla tiene como propósito que el personal de recursos humanos o People, pueda enviar para pago o rechazar las incidencias de su panel.

En ella se presenta una lista de incidencias, en la que también se puede consultar los detalles de cada una individualmente. Adicionalmente cuenta con una serie de filtros en la sección superior que se aplican a la lista de incidencias.

Para rechazar o enviar para pago las incidencias, se habilitarán los botones *Aprobar* y *Rechazar* situados en la parte inferior de la pantalla. Estos harán aparecer las pantallas auxiliares de confirmación que se ven en las imágenes 4.8 y 4.9, esta última con la opción de añadir un comentario.

Por último, en la parte superior derecha se incluirá un botón que permitirá generar un reporte de forma automática.

En las siguientes imágenes se presentan la Pantalla de People 4.10 y la vista de los detalles de la incidencia 4.7, así como las pantallas de aprobar 4.8 y rechazar 4.9.

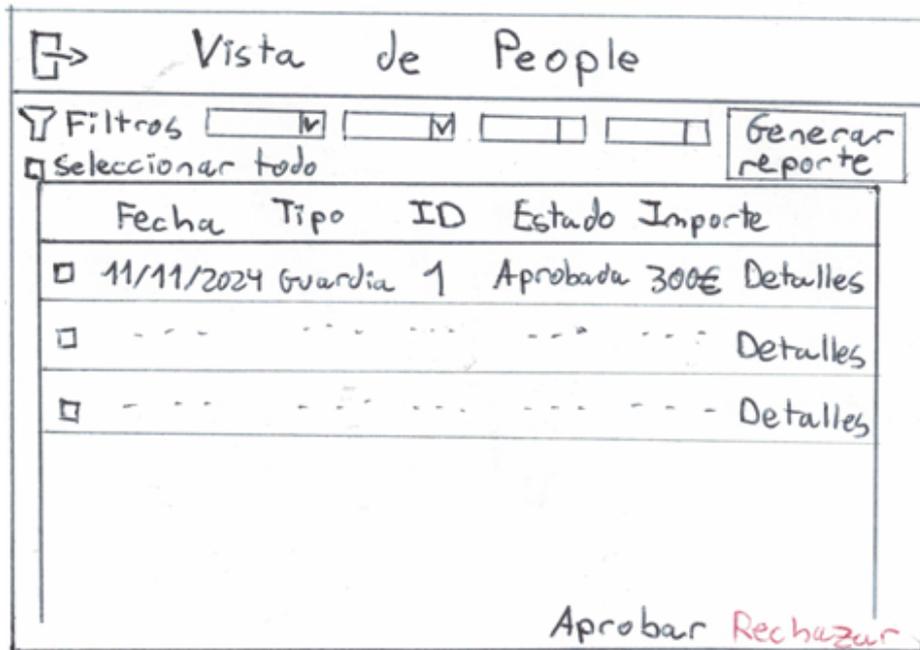


Figura 4.6: Boceto de la Pantalla de People

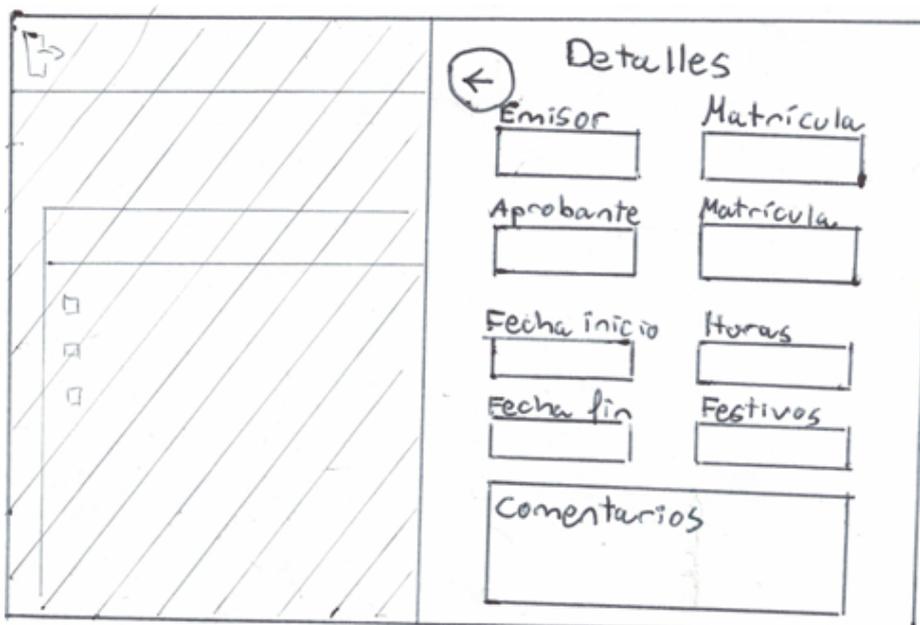


Figura 4.7: Boceto Consultar Detalles



Figura 4.8: Boceto Enviar Incidencias

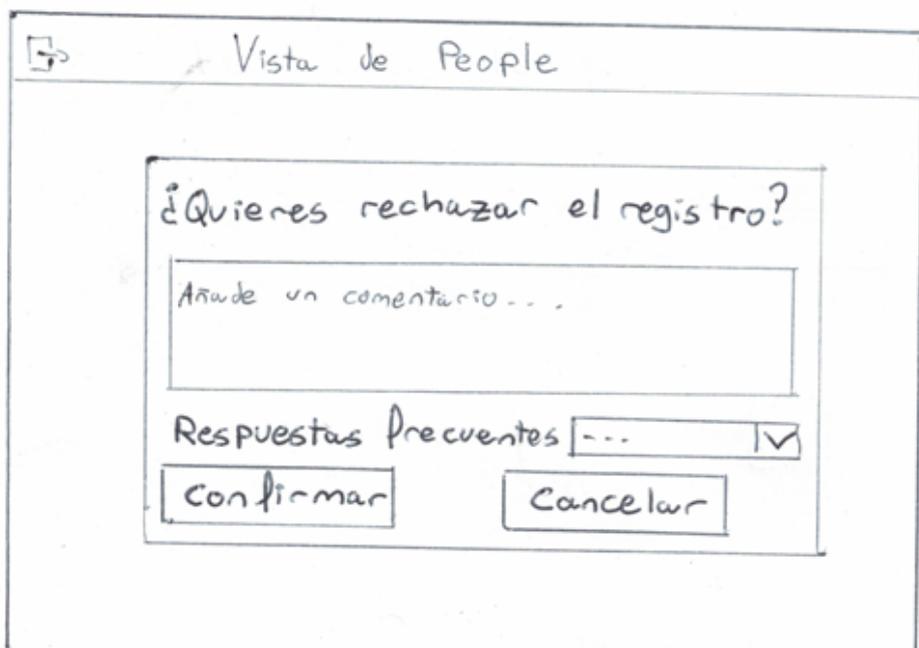


Figura 4.9: Boceto Rechazar Incidencias

Pantalla Manager

Esta pantalla es similar a la anterior en cuanto al uso de una galería para la presentación de las incidencias a aprobar o rechazar por parte del Manager. También cuenta con una serie de filtros para la lista de incidencias que se nos presentan y la opción de consultar los detalles de estas, de igual manera que se vió anteriormente 4.7. Para aprobarlas o rechazarlas se usarán botones como los descritos anteriormente que habilitarán pantallas auxiliares como las presentadas en las figuras 4.8 4.9.

4.3. Diseño de la interfaz de usuario

La particularidad de esta pantalla reside en una funcionalidad más compleja para la obtención de reportes, en la que a través de una pantalla auxiliar 4.11 se permitirá introducir el periodo en el que se quieren consultar las incidencias. También dispone de un botón en la parte superior central que permitirá al Manager ir a su propia pantalla de Empleado en la que crear incidencias.

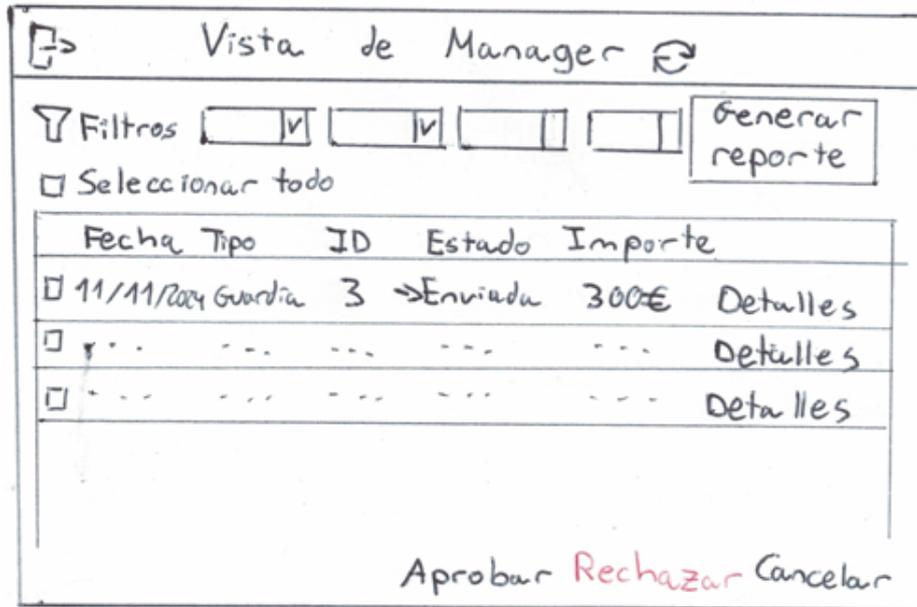


Figura 4.10: Boceto de la Pantalla de Manager

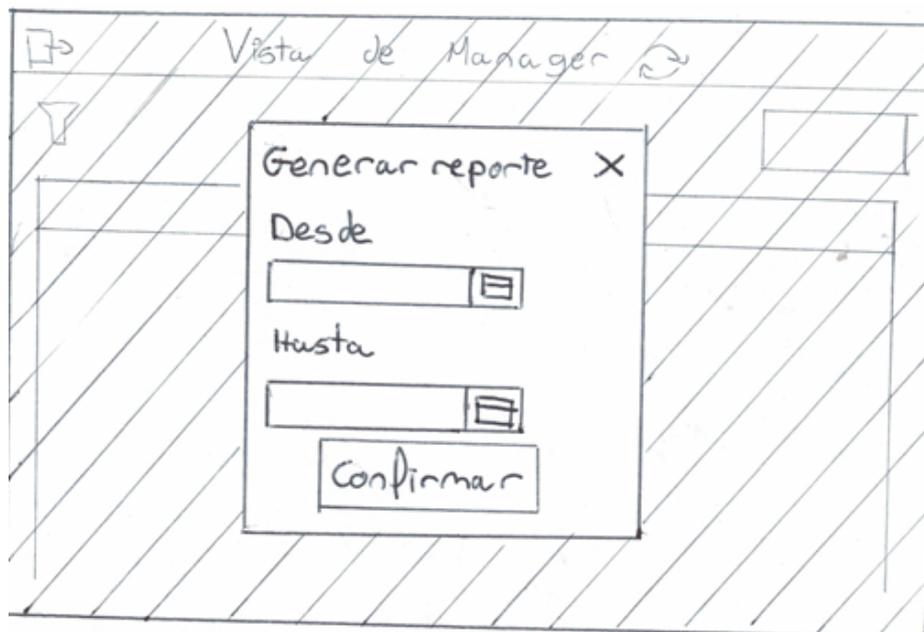


Figura 4.11: Boceto de la Pantalla Generar Reporte Manager

Pantalla Empleado

Esta última pantalla es el punto en el que un Empleado crea las incidencias. Tiene tres funcionalidades clave:

- Galería de incidencias del Empleado: esta galería es similar a la de las pantallas de People y Manager, de forma que aquí se presentan todas las incidencias creadas por el Empleado independientemente del estado en el que estén. Esta galería cuenta también con una serie de filtros que nos permiten seleccionar las incidencias en base a varios criterios. Podremos consultar también los detalles de las incidencias ya creadas de la misma manera que en los casos anteriores y según se muestra en la figura 4.7.
- Botones de enviar, cancelar y eliminar: estos botones nos permiten enviar las incidencias creadas a que sean aprobadas por el Manager, cancelar las ya enviadas o borrar incidencias que no hayamos enviado, se encuentran en la parte inferior derecha. Habilitan pantallas similares a las de las figuras 4.8 4.9.
- Crear y editar incidencias: tal y como se muestra en la figura 4.13, un Empleado tiene la posibilidad tanto de crear una nueva incidencia como de editar una ya existente, a través de una pantalla auxiliar que se le abre al hacer click en los botones de crear o de editar. Para ello el Empleado tendrá que rellenar los campos y hacer click en guardar.

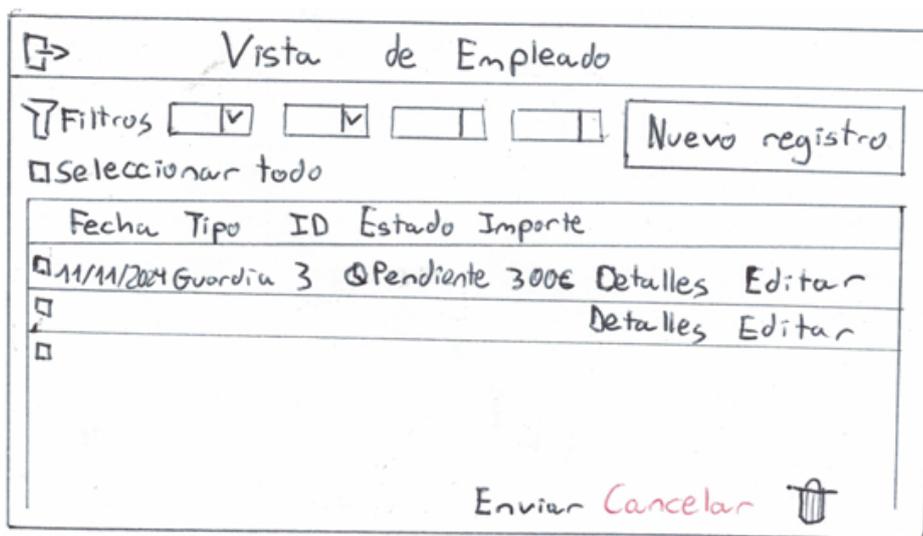


Figura 4.12: Boceto de la Pantalla de Empleado

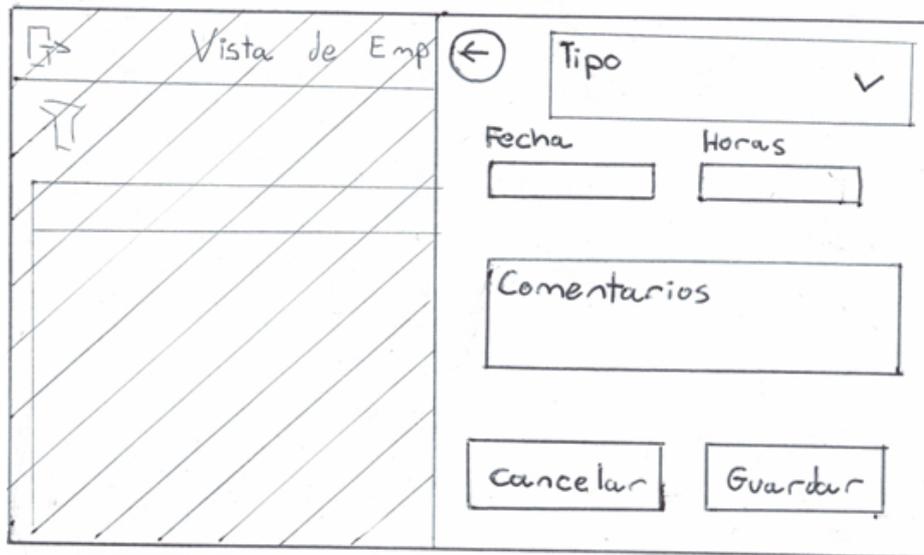


Figura 4.13: Boceto de la Pantalla de Crear/Editar Incidencia

Capítulo 5

Implementación

5.1. Tecnologías empleadas

5.1.1. Microsoft PowerApps

En el contexto empresarial actual, donde la transformación digital es una prioridad, la necesidad de herramientas de desarrollo de aplicaciones que sean rápidas y accesibles ha crecido exponencialmente. Es aquí donde entran herramientas low code, como PowerApps, que permiten a las empresas adaptarse a la demanda del mercado sin necesidad de depender de equipos especializados en desarrollo. Esto permite que usuarios sin unos conocimientos profundos de programación sean capaces de crear soluciones que optimicen procesos internos, haciéndolos más eficientes y menos costosos, mejorando de esta forma su competitividad.

PowerApps es una plataforma de desarrollo de aplicaciones low code perteneciente al entorno de Microsoft 365, con el objetivo de que todo tipo de desarrolladores y usuarios puedan crear aplicaciones personalizadas sin necesidad de recurrir a lenguajes de programación avanzados consiguiendo la funcionalidad con técnicas drag and drop, en la que se sitúan visualmente los diferentes elementos que aportan funcionalidad.

A través de una interfaz gráfica intuitiva, PowerApps hace más fácil diseñar aplicaciones funcionales y atractivas visualmente a través de un entorno en red que permite la colaboración de distintos editores. Esto consigue reducir tanto tiempos de desarrollo, como ahorrar costes.

Para las empresas, PowerApps se ha convertido en una herramienta clave para digitalizar procesos, permitiendo automatizar tareas, gestionar datos de forma similar a un Excel y crear flujos de trabajo eficientes, a través de la posibilidad de interconectar distintas fuentes de datos [32].

Estas aplicaciones son soluciones muy flexibles, que pueden ser ejecutadas en una amplia gama de plataformas, como navegadores, móviles o tablets. También tienen capacidad para conectarse a una amplia gama de fuentes de datos, entre las que se pueden encontrar SharePoint, SQL Server, OneDrive o Excel entre otros, permitiendo a los usuarios disponer de aplicaciones que permitan la

gestión de la información en tiempo real, a través de un entorno corporativo seguro y centralizado [33]. Adicionalmente a las fuente de datos, estas aplicaciones también pueden integrarse con otros servicios del entorno de Microsoft, como pueden ser Power Automate 5.1.3, Teams o Azure.

PowerApps obtiene su funcionalidad a través de la conjunción de una serie de elementos. A continuación se presenta una lista con los componentes principales:

1. **Pantallas:** una aplicación está compuesta por una o más pantallas y cada una representa una vista de esta. Cada pantalla suele tener su propia funcionalidad, a partir de una lógica definida, y los usuarios podrán navegar entre ellas dependiendo del flujo de la aplicación.
2. **Controles:** son los elementos interactivos que se encuentran en las pantallas, tales como botones, entradas de texto, galerías o listas desplegadas, entre otras opciones. Estos controles permiten la interacción del usuario con la aplicación así como introducir o visualizar datos.
3. **Galerías y formularios:** estos controles permiten mostrar datos de forma estructurada. Mientras que las galerías sirven para mostrar listas de datos, por ejemplo los registros de una lista de SharePoint, los formularios sirven para ingresar y editar la información.
4. **Conectores:** PowerApps dispone de una amplia variedad de conectores que permiten integrar las aplicaciones con fuentes de datos y servicios pertenecientes al entorno de Microsoft. Algunos de los conectores más habituales que podemos encontrar son los de SharePoint, OneDrive, SQL Server, DataVerse, Power Automate, Microsoft Teams o Excel entre otros. Estos conectores permiten acceder y manipular datos de forma remota desde la aplicación [33].

Ventajas de usar PowerApps:

- **Reducción de tiempo y costes:** dado que es una plataforma low code, una empresa que quiera implementar una solución a través de PowerApps no necesita contar con equipos especializados en desarrollo de software, lo que reduce los costos asociados a contratar desarrolladores tanto internos como externos. Además, dado que la mayoría de la funcionalidad es a través de fórmulas y configuraciones visuales, el tiempo de desarrollo se reduce [34].
- **Interfaz intuitiva e iteración rápida:** la interfaz de PowerApps está diseñada para ser intuitiva, teniendo características con varias aplicaciones del entorno de Microsoft que pueden resultar familiares a la mayor parte de usuarios, como por ejemplo Power Point. También permite desarrollar y probar las aplicaciones en tiempo real. De esta forma se facilita la iteración continua, garantizando de esta forma que los resultados estén alineados con las necesidades de los usuarios finales [35].
- **Conectividad e integración:** PowerApps cuenta con una amplia variedad de conectores que permiten integrar la aplicación con otros servicios de Microsoft 365 como se mencionó anteriormente. De esta forma conseguimos, a parte de mucha funcionalidad y personalización, disponer de un almacén centralizado de datos que pueden ser usados por distintas aplicaciones [36].

- **Despliegue multiplataforma:** las aplicaciones, al ser en red, pueden ejecutarse en distintas plataformas sin necesidad de tener que desarrollar versiones específicas para cada una. Las aplicaciones están diseñadas para funcionar tanto en dispositivos móviles (iOS y Android) como en navegadores web, lo que permite a los usuarios acceder a las herramientas desde cualquier lugar y dispositivo [37].
- **Integración con flujos automáticos:** PowerApps tiene una sinergia excelente con Power Automate, lo que permite automatizar tareas y flujos de trabajo. Esto resulta especialmente útil en procesos de aprobación de solicitudes, donde se pueden desencadenar acciones automáticas basadas en los datos ingresados en la aplicación de Power Apps. [38].

A pesar de contar con una amplia gama de ventajas, PowerApps presenta una serie de **limitaciones** que deben ser tenidas en cuenta.

- **Limitaciones en la personalización:** a pesar de la gran flexibilidad en el diseño, en personalizaciones muy complejas o específicas, el enfoque low code puede quedarse corto. Esto hace que las posibilidades de personalización sean limitadas en comparación con lenguajes como C# o JavaScript [39].
- **Escalabilidad:** las aplicaciones son ideales para escenarios de pequeña y mediana escala, sin embargo el rendimiento puede empeorar en situaciones con grandes volúmenes de datos o un amplio número de usuarios simultáneos, en especial cuando el manejo de los datos se realiza en tiempo real o con operaciones complejas de backend [40].
- **Dependencia del ecosistema de Microsoft:** a pesar de que PowerApps se puede conectar con servicios externos, su potencial reside en utilizar la aplicación en conjunto con otras herramientas del ecosistema de Microsoft, como SharePoint, SQL Server, Azure o Power Automate. Esto puede suponer un problema para empresas cuyas tecnologías están fuera del entorno Microsoft, dado que la integración con servicios externos no es tan fluida.
- **Dificultad de mantenimiento:** al ser aplicaciones cuya funcionalidad se distribuye a lo largo de diferentes componentes, cada uno con fragmentos de código propios, estas aplicaciones dependen de una documentación precisa para entender el flujo de su funcionamiento, lo que puede complicar mezclar nueva funcionalidad con otra ya existente.

PowerApps utiliza PowerFX [41], que es un lenguaje de programación funcional, declarativo, de tipo fuerte y uso general. Está diseñado para facilitar el desarrollo de aplicaciones mediante fórmulas y expresiones intuitivas similares a las de Excel. Esto lo convierte en una opción idónea tanto para usuarios técnicos como no técnicos. PowerFX permite manipular datos, crear lógica personalizada y agregar funcionalidades avanzadas, todo dentro de un enfoque declarativo en el que el desarrollador define "qué" debe ocurrir en lugar de "cómo" debe hacerse. Esto que simplifica el desarrollo y reduce la necesidad de manejar detalles técnicos complejos.

Algunas de sus características clave son:

1. **Lenguaje de tipado fuerte:** PowerFX realiza comprobaciones en tiempo de diseño para prevenir errores y garantizar que las fórmulas se comporten correctamente [42].
2. **Familiaridad con Excel:** al estar basado en las fórmulas de Excel, los usuarios pueden reciclar su conocimiento previo para programar usándolo.
3. **Declarativo y funcional:** PowerFX es un lenguaje declarativo que permite definir el comportamiento deseado sin necesidad de especificar secuencias de control detalladas, lo que reduce la complejidad del desarrollo [43].

Dentro de nuestra aplicación, hay una serie de funciones que se han utilizado y que son especialmente relevantes:

- **Patch:** se trata de una función que sirve para crear y actualizar registros en la base de datos, permitiendo la inserción y modificación de incidencias en la aplicación sin necesidad de utilizar formularios [44], lo que nos permite un mayor grado de personalización. Patch simplifica la interacción con las fuentes de datos [45].
- **Filter:** se ha utilizado para la búsqueda en la galería de incidencias y la aplicación de filtros por criterios como estado, fechas, o tipo de incidencia. Es clave para proporcionar una experiencia de búsqueda satisfactoria [46].
- **LookUp:** esta función permite recuperar de las fuentes información específica a través de una correspondencia entre estas y una entrada en la aplicación. LookUp facilita la extracción de información [46].
- **Navigate:** se trata de la función esencial para navegar entre pantallas [47].
- **Collect:** las colecciones son estructuras que almacenan conjuntos de datos temporalmente en memoria. Resultan muy útiles cuando tenemos que manejar varias filas de la tabla a la vez para modificarlas, guardarlas o eliminarlas de forma simultánea [48].

En resumen, PowerApps se trata de una plataforma low code que permite a los usuarios crear aplicaciones personalizadas de una forma rápida y eficiente facilitando el acceso a estas soluciones a todo tipo de empresas.

5.1.2. Microsoft SharePoint

SharePoint es una plataforma de colaboración y gestión de documentos desarrollada por Microsoft, diseñada para facilitar el trabajo en equipo y la administración de contenido dentro de las organizaciones [49]. Permite a los usuarios crear, compartir y gestionar sitios web, bibliotecas de documentos y listas, optimizando la comunicación y el flujo de trabajo.

Estas listas y basado en el diseño de la base de datos de la sección 4.2, se integran con PowerApps y sirven como tablas de una base de datos, proporcionando una forma de crear, consultar y modificar datos eficiente y centralizada.

Entre las principales características de SharePoint se encuentran:

- **Colaboración:** los diferentes usuarios pueden trabajar de forma conjunta, accediendo, editando y creando documentos simultáneamente. Esto consigue mejorar la eficiencia y la productividad, eliminando la necesidad de enviar archivos por correo electrónico y eliminando versiones desactualizadas de documentos. Además, cuenta con la posibilidad de asignar tareas y añadir comentarios, lo que fomenta el trabajo en equipo.
- **Gestión de documentos:** SharePoint dispone de herramientas potentes para almacenar, organizar y acceder a documentos. A través de bibliotecas, los usuarios pueden clasificar y buscar archivos de manera eficiente, a la par que estos se mantienen actualizados gracias a las herramientas de control de versiones que ofrece. Los usuarios pueden volver a las versiones anteriores en caso de ser necesario, asegurando de esta forma la integridad de la información. Adicionalmente, proporciona un nivel más de seguridad en el caso de borrado (tanto accidental como malicioso) de documentos y su posterior recuperación.
- **Automatización de flujos de trabajo:** A través de Microsoft Power Automate, SharePoint permite la creación de flujos de trabajo automatizados que pueden simplificar procesos complejos y rutinarios. Por ejemplo, una empresa puede automatizar la aprobación de documentos, el envío de notificaciones o la recopilación de datos para la elaboración de informes. Esto consigue, no solo reducir la carga sobre los trabajadores, sino que también mejora la consistencia y la transparencia en los procesos, lo que a su vez puede conducir a una mayor satisfacción del cliente y una mejor gestión del tiempo [50].
- **Integración con otras herramientas:** SharePoint se integra muy bien con otras aplicaciones del ecosistema de Microsoft, tales como PowerApps, Teams o Power BI. A través de esta integración, las empresas pueden crear un entorno de trabajo unificado donde las aplicaciones comparten datos y funcionalidad de manera fluida y en tiempo real. Los usuarios podrían por ejemplo acceder a documentos de SharePoint desde Teams y utilizar Power Apps para realizar modificaciones en ellos, o visualizarlos a través de Power BI. El disponer de toda la información en un único lugar mejora la comunicación y facilita la toma de decisiones [51].
- **Personalización:** SharePoint ofrece un alto grado de personalización, lo que permite a los usuarios adaptar sus sitios y bibliotecas según las necesidades específicas de su empresa. Los administradores del espacio pueden elaborar listas y bibliotecas o definir flujos de manera personalizada, así como definir roles con permisos específicos. De esta forma se garantiza la protección de la información sensible, asegurando que los datos críticos sean solamente accesibles para las personas designadas. Además, las capacidades de diseño de SharePoint permiten a las empresas personalizar la apariencia y la funcionalidad de sus sitios para alinearse con su marca y cultura corporativa.

En resumen, SharePoint es una herramienta potente para el entorno empresarial que mejora la comunicación, la gestión de documentos y la eficiencia operativa, adaptándose a las necesidades específicas de cada compañía.

5.1.3. Microsoft Power Automate

Power Automate es una herramienta de automatización previamente conocida como Microsoft Flow, que forma parte del ecosistema de Microsoft 365. Permite a los usuarios crear flujos de trabajo automatizados entre distintas aplicaciones y servicios. Estos flujos conectan aplicaciones como PowerApps, SharePoint, Outlook, Dynamics 365, Microsoft Teams y muchas otras, reduciendo la necesidad de realizar tareas manuales como enviar correos electrónicos y reduciendo los tiempos de respuesta [52].

Una de las grandes ventajas de Power Automate, que comparte con el resto de herramientas de Microsoft, es su enfoque *low code*, que permite a todo tipo de usuarios, independientemente de su nivel técnico, diseñar e implementar flujos. Esto lo convierte en una herramienta accesible tanto para desarrolladores como para usuarios sin experiencia previa en programación. Los flujos se diseñan a través de una interfaz gráfica, en la que a través de un enfoque *drag and drop* (arrastrar y dejar) y una configuración sencilla de cada componente se va elaborando el flujo de trabajo [53].

Power Automate permite la automatización de una amplia gama de procesos, como el envío de notificaciones automáticas, el acceso o modificación de datos entre aplicaciones, la aprobación de documentos, o incluso la realización de tareas más complejas que involucren inteligencia artificial o análisis de datos. Su integración con otros productos del ecosistema de Microsoft, como PowerApps, SharePoint o Power BI, aumenta significativamente la capacidad de las organizaciones para automatizar procesos en todos los niveles [54].

Entre sus principales características se destacan:

- **Automatización de flujos:** Power Automate permite la creación de flujos de trabajo automatizados que conectan múltiples aplicaciones y servicios, eliminando tareas manuales repetitivas y permitiendo a los usuarios centrarse en actividades de mayor valor.
- **Aprobaciones:** Los flujos de aprobación permiten a las organizaciones automatizar la aprobación de documentos, gastos, solicitudes y otros procesos. Los aprobadores pueden recibir notificaciones y aprobar o rechazar elementos directamente desde su dispositivo móvil o correo electrónico. Esto genera un proceso más robusto, eliminando el error humano y eficiente.
- **Integración con inteligencia artificial:** Power Automate ofrece conectores y capacidades que permiten integrar servicios de inteligencia artificial, como la detección del idioma en correos y documentos o el análisis de sentimientos para saber si la opinión es positiva, negativa o neutral, en los flujos de trabajo. También puede usar IA para la extracción de información de documentos, como nombres o fechas [55].
- **Flujos basados en eventos:** Los flujos pueden configurarse para dispararse automáticamente en función de eventos específicos, como la llegada de un correo electrónico, la modificación de una lista de SharePoint o una mención en Microsoft Teams.
- **Conectores:** Con más de 500 conectores disponibles, Power Automate permite integrar flujos de trabajo con una amplia variedad de servicios y aplicaciones de terceros, como pueden ser Drop-

box, Google Drive, Trello, o SQL Server, facilitando la automatización en entornos heterogéneos [56].

A pesar de sus numerosas ventajas, Power Automate presenta una serie de desventajas que deben ser tenidas en cuenta antes de tomarse como la herramienta base para la automatización de flujos de trabajo:

- **Limitaciones en escenarios complejos:** aunque Power Automate funciona bien en tareas básicas o no muy complejas, puede no ser la solución ideal para automatizaciones extremadamente sofisticadas o personalizadas. Además, en amplios volúmenes de datos o tareas con un amplio número de pasos, la ejecución puede demorarse y no ser la herramienta más eficiente.
- **Dependencia del ecosistema Microsoft:** al igual que pasa con PowerApps 5.1.1, Power Automate, aunque soporta aplicaciones de terceros, obtiene su potencial máximo cuando se usa junto a otras herramientas de Microsoft 365, lo que puede ser un inconveniente para organizaciones cuya infraestructura tecnológica esté fuera del entorno de Microsoft [54].
- **Costes adicionales:** algunas funcionalidades avanzadas, como los conectores premium o los modelos de IA, tienen un coste adicional, lo que podría aumentar los gastos en ciertas implementaciones.

A pesar de sus desventajas, Power Automate, no solo mejora la eficiencia de las operaciones empresariales, sino que también permite a las organizaciones adaptarse más rápidamente a las necesidades cambiantes del mercado, mejorando su agilidad y capacidad de respuesta.

Para nuestra aplicación, se han usado cuatro flujos de Power Automate, todos ellos se ejecutan manualmente desde Power Apps. Cada uno dispone de sus propios conectores y se muestran a continuación:

1. **Acceso usuarios:** el flujo *AccesoUsuarios* sirve para que los usuarios accedan a la aplicación, a través de un correo de aprobación que se le envía al usuario para verificar que es él quien está intentando acceder a la aplicación y que devuelve su respuesta a Power Automate. Finalmente, a través del conector *Responder a una Power App o un flujo*, se actualiza la respuesta del usuario en la aplicación. Es una mejora importante pues permite prescindir de la gestión de contraseñas, validando la identidad a través de correo, cuyo acceso ya requiere autenticación.

The screenshot shows the configuration for the 'Start and wait for an approval' connector. The title bar reads 'Start and wait for an approval'. The configuration is as follows:

- Approval type:** Custom Responses – Wait for all responses
- Response options Item 1:** Si, he sido yo
- Response options Item 2:** No, rechazar el inicio de sesión
- Add new item:** + Add new item
- Title:** Confirme que ha sido usted quien acaba de iniciar sesión.
- Assigned to:** {x} email x ;
- Details:** Si confirma, tendrá acceso a la aplicación de gestión de horas extras con el usuario indicado anteriormente. Para ello, recuerde hacer click en "Submit" tras confirmar si ha sido usted o no quien desea aprobar el inicio de sesión. Por favor, por seguridad es muy importante que no comparta este correo con nadie.
- Item link:** Add a link to the item to approve
- Item link description:** Describe the link to the item
- Show advanced options:** Show advanced options

Figura 5.1: Conector SharePoint para esperar por una aprobación

2. **Generar reporte:** el flujo *GenerarReporte* crea un reporte en .csv en el que aparecen todas las incidencias de la pantalla de People y lo envía por correo al usuario de People que lo solicita. Para ello usa los conectores de SharePoint y de Outlook. La única entrada desde PowerApps es el correo al que se envía el archivo adjunto. Tanto para este flujo como para el siguiente, se prefirió optar por la opción del envío por correo, dado que crear un archivo que se descargara desde SharePoint era una función premium como las citadas anteriormente.

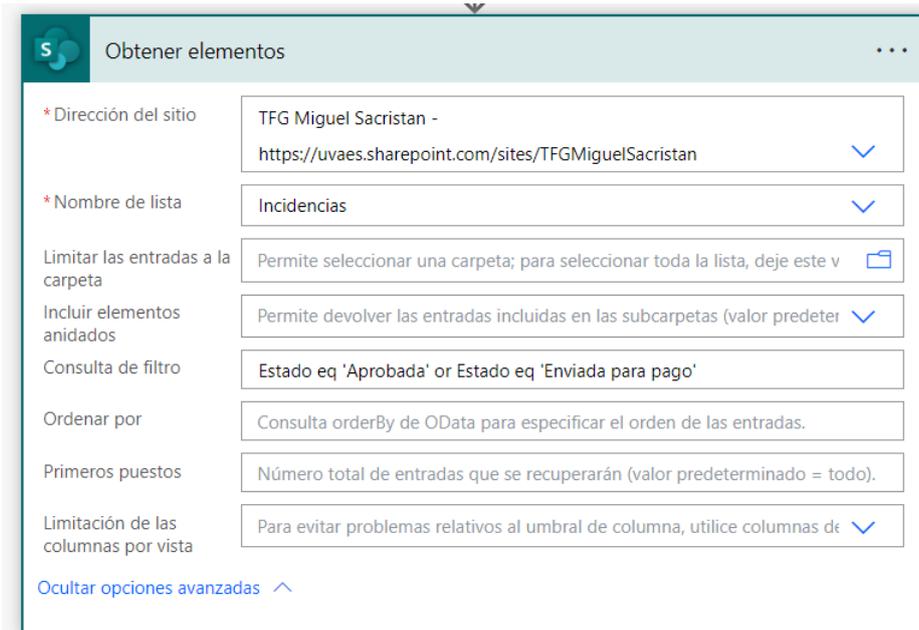


Figura 5.2: Conector SharePoint obtención de elementos

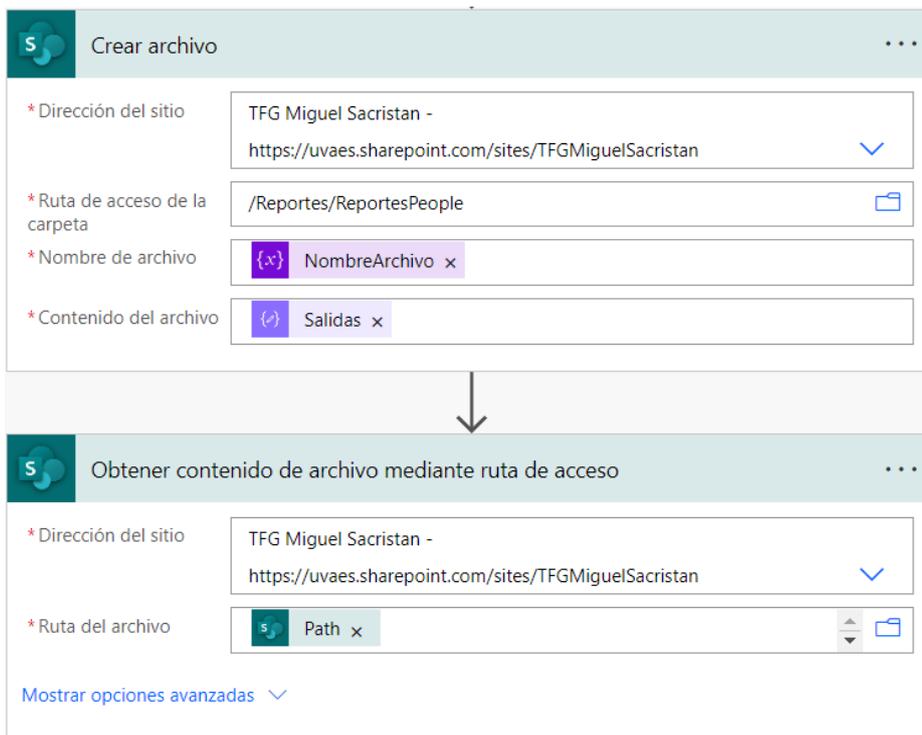


Figura 5.3: Conectores SharePoint para crear el archivo y obtener su ruta

Enviar correo electrónico (V2) ...

* A

* Asunto

* Cuerpo

Remitente (enviar como)

CC

CCO

Datos adjuntos Nombre

Datos adjuntos Contenido

+ Agregar un nuevo elemento

Confidencialidad

Responder a

Importancia

[Ocultar opciones avanzadas](#) ^

Figura 5.4: Conector Outlook archivo adjunto

3. **Generar reporte Manager:** el flujo *GenerarReporteManager* actúa de forma similar al anterior, pero con la particularidad de que los reportes muestran solo las incidencias del Manager, y adicionalmente, este puede elegir el intervalo de fechas en que las quiere. Estas fechas serán parámetros de entrada en el conector de PowerApps con Power Automate y se usarán para seleccionar las incidencias que cumplan los criterios en la lista *Incidencias* de SharePoint.

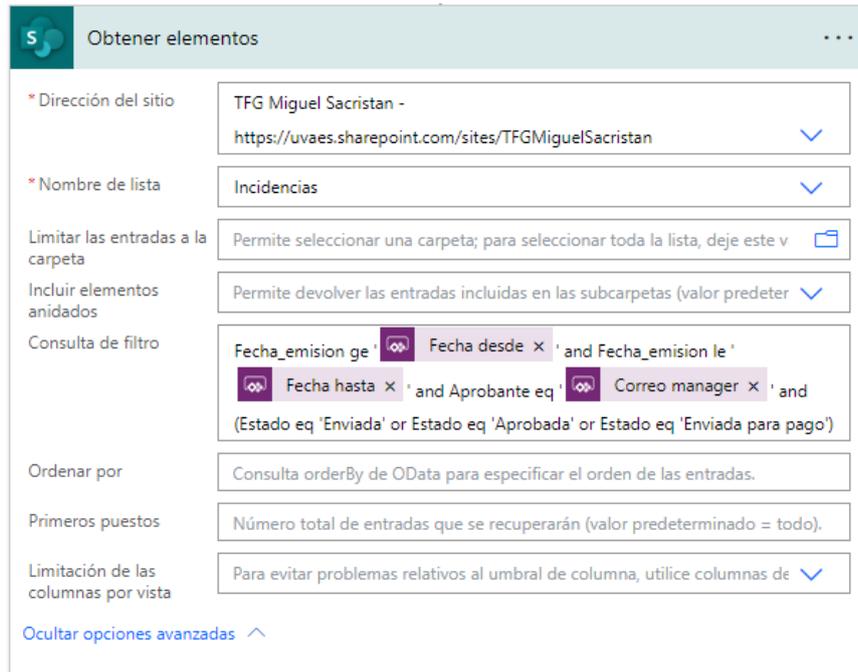


Figura 5.5: Conector SharePoint obtener elementos Manager

4. **Envío de notificaciones:** el flujo *EnvioDeNotificaciones*, dependiendo del evento del flujo de aprobación, tiene la función de enviar una notificación personalizada al empleado, al Manager o a ambos en la que se informe del estado. Esta notificación se realizará a través del correo electrónico usando un conector de Outlook. El flujo tendrá como parámetros de entrada la dirección de email a la que se debe enviar el correo, el estado en que se encuentra la incidencia, el usuario que creó la incidencia, la fecha de emisión, y el identificador de la incidencia.

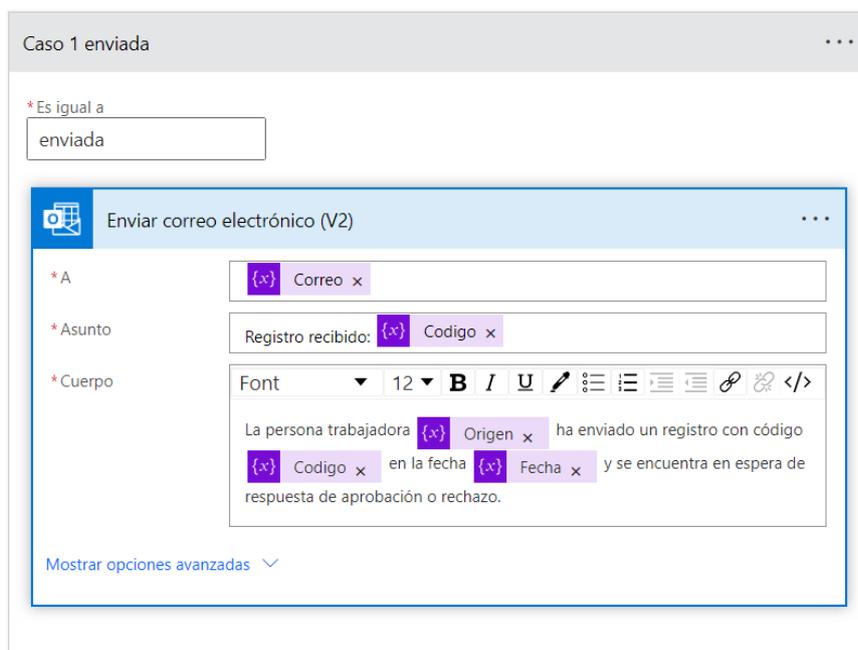


Figura 5.6: Conector de Outlook en Power Automate

5.2. Control de versiones

El control de versiones que ofrece PowerApps es una herramienta necesaria para garantizar la integridad, la correcta gestión de los cambios y la colaboración. Para esto, Microsoft ofrece una serie de funcionalidades que permiten gestionar las diferentes versiones, entre ellas se encuentran:

- **Historial de versiones:** PowerApps dispone de un registro de las versiones publicadas de la aplicación. Es importante destacar que no son las versiones guardadas si no las que se publican. Esto permite retroceder a versiones previas en caso de ser necesario, haciendo que las modificaciones no sean irreversibles. El proceso de restauración es sencillo y garantiza que se pueda volver a un estado funcional en caso de problemas críticos.
- **Comentarios en las versiones:** cuando se publica una nueva versión, se pueden añadir comentarios que expliquen los cambios realizados en la aplicación desde la versión anterior. Esto mejora la documentación y la comunicación entre los miembros del equipo, haciendo que los cambios se entiendan mejor y facilitando correcciones en el futuro.
- **Integración con GitHub y Azure DevOps:** aunque se trata de una función experimental, los desarrolladores pueden autenticarse en Git y esto permite que la aplicación no se bloquee para el resto de desarrolladores mientras uno la está editando. A medida que se realizan los cambios, al sincronizar, estos se aplican a las versiones de todos los usuarios, de esta forma Git facilita la edición y supone una herramienta más de respaldo para la aplicación [57].

En mi caso el historial de versiones, unido a los comentarios, permitió llevar un seguimiento de la funcionalidad que se iba implementando según lo planificado en la sección 8, lo que facilitó también la realización de la memoria. Además sirvió como copia de seguridad ante posibles pérdidas de información recogidas en los riesgos 2.7 y 2.9 al ser versiones publicadas y por lo tanto más robustas que las que solamente se guardan en el entorno de trabajo virtual sin publicar. Finalmente, este historial de versiones ayudó a revertir cambios en funcionalidades nuevas cuyo desempeño difería del esperado e incluso interferían negativamente con otras en las que su comportamiento era el correcto y estaba ya probado.

5.3. Seguridad en el entorno Microsoft

El entorno de Microsoft 365 cuenta con una serie de medidas y herramientas para proteger la información y aumentar la seguridad. Entre ellas destaca la autenticación multifactor (MFA) [58], que añade una capa más de seguridad, obligando a que el usuario verifique su identidad a través de más de un método, a parte de su usuario y contraseña. Estos métodos suelen ser un SMS o una llamada al teléfono móvil. También cuenta con un cifrado de datos que evita accesos no autorizados y un sistema de acceso basado en roles, que permite restringir el acceso a los recursos de forma personalizada, asegurando de esta forma que solo los usuarios con permisos puedan acceder a la información.

Implementación

Microsoft cuenta también con herramientas para garantizar que se cumple con la normativa del Reglamento General de Protección de Datos (RGPD) [59], a través de funcionalidades para la auditoría de datos y la gestión de identidades con herramientas como Azure Active Directory. Esto asegura que las organizaciones puedan gestionar adecuadamente los datos personales y cumplir con los requisitos de privacidad y seguridad establecidos por el RGPD [60].

Capítulo 6

Pruebas

6.1. Pruebas

Las pruebas [61] resultan fundamentales en cualquier ciclo de desarrollo software, dado que garantizan que la aplicación se ajuste a los requisitos y no presente errores críticos. Al realizarlas a lo largo de todo el desarrollo, no solo al final, los defectos se pueden identificar y corregir en etapas tempranas del desarrollo, reduciendo de esta forma el coste en fases avanzadas y mejorando el resultado final. Hacer pruebas ayuda a que el sistema sea más fiable y fácil de mantener, reduciendo la dificultad de adaptaciones futuras o escalabilidad de la aplicación.

6.1.1. Pruebas en PowerApps

En este proyecto, al aplicar la metodología Scrum con entregas incrementales y desarrollo en iteraciones, las pruebas unitarias se realizaron desde las primeras etapas de forma paralela al desarrollo del código, tal como se describe en la sección 2.4. Este enfoque permitió verificar que la funcionalidad implementada durante y al final de cada sprint cumpliera con los requisitos, además de facilitar mejoras continuas basadas en los resultados obtenidos. Cada pantalla se evaluó de forma individual para asegurar el correcto funcionamiento y ajuste a los objetivos definidos de diseño y funcionalidad. Durante el último sprint (Sprint 6), se desarrolló un plan de pruebas integral para comprobar la interacción entre los diferentes módulos y garantizar la cohesión del sistema en su totalidad.

A continuación, veremos las pruebas unitarias asociadas a los casos de uso más representativos. El resto de pruebas son análogas a las realizadas.

Prueba 1: Crear Trabajo programado	
Descripción	Esta prueba sirve para verificar que la interfaz de creación de los trabajos programados funciona correctamente. También se comprueba si los datos se almacenan en SharePoint.
Entrada	<ul style="list-style-type: none"> ■ Tipo de registro: Trabajo programado ■ Fecha: 04/09/2024 ■ Hora inicio: 18:15 ■ Hora fin: 20:30 ■ Comentarios: Prueba integrada: Crear Trabajo programado
Salida esperada	El sistema calcula las horas y la remuneración y crea un nuevo registro de tipo <i>Trabajo programado</i> en estado <i>Pendiente</i>
Salida obtenida	Correcta

Tabla 6.1: Prueba 1: Crear Trabajo programado

Prueba 2: Crear Guardia	
Descripción	Esta prueba sirve para verificar que la interfaz de creación de las guardias y sus intervenciones funciona correctamente. También se comprueba si los datos se almacenan en SharePoint.

Entrada	<ul style="list-style-type: none"> ■ Tipo de registro: Guardia ■ Fecha: 02/09/2024 ■ Festivos: 1 ■ Fecha Intervención 1: 02/09/2024 ■ Horas Intervención 1: 2 ■ Fecha Intervención 2: 05/09/2024 ■ Horas Intervención 1: 1 ■ Comentarios: Prueba integrada: Crear Guardia
Salida esperada	El sistema calcula las horas y la remuneración y crea un nuevo registro de tipo <i>Guardia</i> en estado <i>Pendiente</i> . También registra las intervenciones asociadas a esa guardia.
Salida obtenida	Correcta

Tabla 6.2: Prueba 2: Crear Guardia

Prueba 3: Eliminar registros	
Descripción	Esta prueba tiene el objetivo de eliminar por parte del Empleado uno o más registros que no hayan sido enviados todavía.
Entrada	Registros con ID 33 y 34
Salida esperada	Los registros seleccionados son eliminados de la tabla de incidencias.
Salida obtenida	Correcta

Tabla 6.3: Prueba 3: Eliminar registros

A continuación se verán las pruebas asociadas a los envíos, aprobaciones, rechazos y cancelaciones de los diferentes perfiles de usuarios de la aplicación. Para no sobrecargar el documento se mostrarán solo algunos de estos casos.

Prueba 4: Enviar incidencias al Manager	
Descripción	Esta prueba verifica que una o más incidencias se envían de forma correcta al Manager, modificando su estado y notificándole por correo.
Entrada	Registros con ID 31 y 32.
Salida esperada	Las incidencias seleccionadas pasan al estado <i>Enviada</i> y el Manager recibe un correo indicándole que las puede aprobar.
Salida obtenida	Correcta

Tabla 6.4: Prueba 4: Enviar incidencias al Manager

La prueba de aprobar por parte del Manager es análoga a esta, con la diferencia de que es el Empleado emisor quien recibe el correo notificándole que esta ha sido aprobada.

Prueba 5: Cancelar incidencias al Manager	
Descripción	En esta prueba comprobamos que, como Empleado, podemos cancelar una o más incidencias que hayan sido enviadas al Manager y que se le notifica correctamente por correo.
Entrada	Registros con ID 31 y 32.
Salida esperada	Las incidencias seleccionadas pasan al estado <i>Pendiente</i> , la fecha de emisión se elimina y el Manager recibe un correo indicándole que han sido canceladas por el Empleado.
Salida obtenida	Correcta

Tabla 6.5: Prueba 5: Cancelar incidencias al Manager

La prueba de cancelar por parte del Manager es análoga, pero es el Empleado quien es notificado que su registro ha sido cancelado por el Manager una vez aprobado.

Prueba 6: Rechazar incidencias	
Descripción	Esta prueba verifica que como Manager podemos rechazar una o más incidencias que hayan sido enviadas por el Empleado y que a este se le notifica correctamente por correo.

Entrada	Registros con ID 31 y 32.
Salida esperada	Las incidencias seleccionadas pasan al estado <i>Pendiente</i> , la fecha de emisión se elimina y el Empleado recibe un correo indicándole que han sido rechazadas por su Manager.
Salida obtenida	Correcta

Tabla 6.6: Prueba 6: Rechazar incidencias

Prueba 7: Enviar para pago incidencias People	
Descripción	Esta prueba verifica que como Personal de People podemos enviar para pago una o más incidencias que hayan sido enviadas para pago. Se comprueba que se notifica tanto al emisor como al aprobante que estas han sido enviadas para pago.
Entrada	Registros con ID 31 y 32.
Salida esperada	Las incidencias seleccionadas pasan al estado <i>Enviada para pago</i> y tanto el Manager como el Empleado reciben un correo de notificación.
Salida obtenida	Correcta

Tabla 6.7: Prueba 7: Enviar para pago incidencias People

La prueba de rechazar las incidencias por parte del Personal de People es análoga, se modifica el estado a *Enviada* y se notifica tanto al emisor como al aprobante.

Prueba 8: Generar reporte Manager	
Descripción	Esta prueba verifica que el Manager recibe por correo un reporte con las incidencias de su panel cuyas fecha de emisión se encuentre entre las seleccionadas.
Entrada	<ul style="list-style-type: none"> ■ Fecha desde: 02/09/2024 ■ Fecha hasta: 23/09/2024

Salida esperada	Archivo .csv recibido por correo con las incidencias cuya fecha de solicitud se encuentra en ese periodo.
Salida obtenida	Correcta

Tabla 6.8: Prueba 8: Generar reporte Manager

La prueba de generar reporte People es análoga a esta con la diferencia de que en ese caso no es necesario indicar fechas de inicio y fin.

Prueba 9: Inicio de sesión	
Descripción	En esta prueba se comprueba que el usuario puede iniciar correctamente la sesión y la aplicación le lleva a la pantalla asignada a su rol.
Entrada	<ul style="list-style-type: none"> ▪ Usuario: miguel.sacristan@estudiantes.uva.es
Salida esperada	La aplicación reconoce el usuario, envía el correo de autorización a la dirección indicada, y tras aprobar el inicio de sesión, nos lleva a la Pantalla de Manager.
Salida obtenida	Correcta

Tabla 6.9: Prueba 9: Inicio de sesión

De forma análoga a esta prueba, se comprobó también de forma exitosa que, al introducir credenciales incorrectas, la aplicación le mostraba el mensaje de error.

Por último, las pruebas del plan integral incluyeron las realizadas en los diferentes filtros de las pantallas para comprobar que las galerías funcionaban correctamente y mostraban los datos conforme a las limitaciones que estos filtros imponían.

Capítulo 7

Despligue

7.1. Despligue

Para desplegar la aplicación y dada la naturaleza sensible de los datos, esta será compartida por correo a los usuarios que la vayan a usar. Para ello, debemos hacer click en el botón de Compartir de la pestaña de aplicaciones de PowerApps.

Dado que la aplicación se ha desarrollado en el entorno de Microsoft 365 de la Universidad, los únicos usuarios que pueden tener acceso a ella son los que dispongan de una cuenta con su dominio, como son las cuentas terminadas en @estudiantes.uva.es o @uva.es.

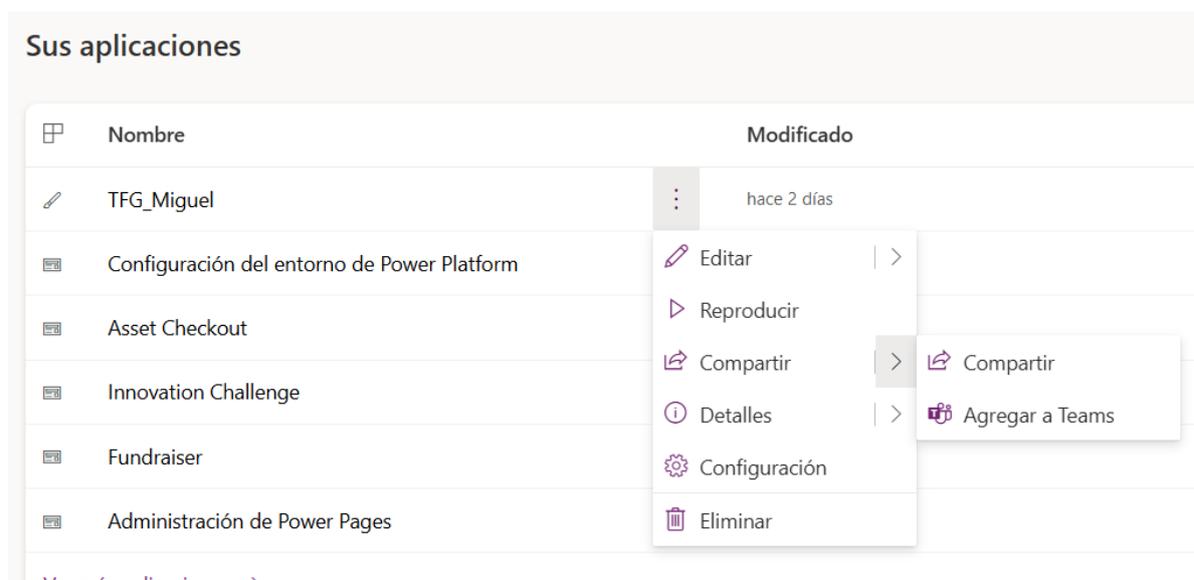


Figura 7.1: Compartir aplicación

Para compartir la aplicación tendremos una serie de opciones:

- **Envío de la aplicación a usuarios específicos:** esta opción permite un control más preciso

sobre quién puede acceder a la aplicación y por ello es la opción elegida. Para ello se envía el acceso a la aplicación a los usuarios seleccionados por correo electrónico. Al hacer esto se les concede permiso de uso o edición según se decida.

- **Compartir vínculo:** se trata de un enlace directo a la aplicación, lo que permite un acceso rápido y flexible siempre y cuando los usuarios tengan los permisos adecuados. Los permisos se pueden ajustar en la misma ventana de compartir.
- **Despliegue en Microsoft Teams:** es una opción muy conveniente para aplicaciones empresariales, permitiendo el acceso a la aplicación desde el entorno colaborativo de Teams. Tras hacer click en *Compartir y Añadir a Teams*, se agrega la aplicación como pestaña en un equipo de Teams específico. Una vez añadido, los miembros del equipo pueden acceder a la aplicación desde la pestaña de Teams correspondiente. En nuestro caso no es la opción elegida, pero para un despliegue real a gran escala sería una de las mejores opciones.

7.1.1. Permisos de SharePoint

Para que los usuarios accedan a los datos almacenados en SharePoint, un proceso necesario para iniciar sesión, acceder a incidencias o registrarlas, es necesario configurar los permisos específicos en las listas de SharePoint vinculadas a la aplicación.

Para ello se debe acceder al sitio y en la *Configuración, Permisos del sitio* y se añaden los usuarios o grupos que pueden acceder a este sitio.

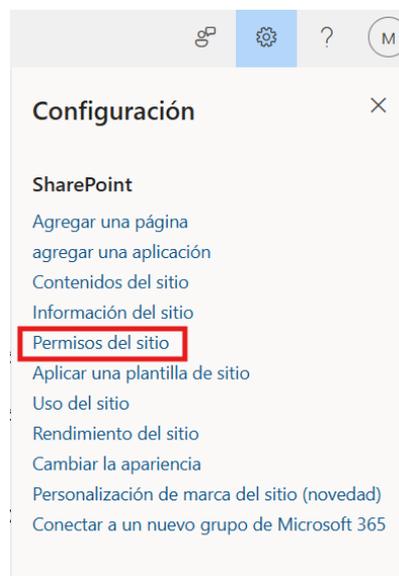


Figura 7.2: Permisos SharePoint

7.1.2. Permisos de Power Automate

Para que los usuarios puedan ejecutar los flujos, es necesario:

Despligue

1. Compartir cada flujo.
2. Añadir a los usuarios o grupos de usuarios, con permisos de *Ejecutar solo*.
3. Revisar los permisos de los conectores.

De esta forma, los usuarios que tengan permitido el acceso a la aplicación, también podrán ejecutar los flujos.

7.1.3. Publicación

Finalmente, para implementar la aplicación, esta tiene que ser publicada, como ya se explicó en la sección 5.2. Una vez publicada, todos los usuarios con permiso para usarla o editarla, pueden ejecutar la aplicación.

El proceso de publicación [62] consta de los siguientes pasos:

1. En la pantalla de edición de PowerApps, se hace click en el botón de *Publicar*. Si se ha realizado cualquier modificación la aplicación la guarda automáticamente.
2. Al pulsar en *Publicar*, nos aparece un menú de confirmación en el que podemos escribir un comentario que reflejará los cambios realizados en la versión.

Publicar ✕

Esta versión de la aplicación se publicará en el entorno Entorno de MIGUEL SACRISTAN DE FRUTOS.
[Obtenga más información sobre la publicación.](#)

Para mejorar el rendimiento de la aplicación, los datos de la aplicación se pueden almacenar localmente en el almacenamiento del explorador y es posible que algunos recursos de la aplicación se carguen antes de que los usuarios finalicen la autenticación.
[Más información](#)

Nombre e icono de aplicación

 TFG_Miguel

Descripción

Proporcione una descripción para ayudar a los usuarios finales a buscar y usar la aplicación.

Cambios v3, corrección interfaz Pantalla Usuario

Publicar esta versión **Editar detalles**

Figura 7.3: Publicar aplicación

7.1.4. Control de versiones

PowerApps dispone también de un historial de versiones, como se mencionó anteriormente en la sección 5.2 al que podemos acceder a través del icono de los puntos suspensivos de la página principal de administradores. Dentro de las opciones, seleccionamos *Detalles ->Versiones* y nos aparecerá el historial de versiones de la aplicación. Desde esta ventana podemos restaurar versiones anteriores en el caso de querer revertir algún cambio o encontrar un problema crítico.

Versión	Modificado	Modificado por	Versión de Power Apps	Publicado
Versión 140	... 12/11/2024, 23:24:46	MIGUEL SACRISTAN DE FRUTOS	3.24104.14	Activo
Versión 139	... 9/11/2024, 13:29:00	MIGUEL SACRISTAN DE FRUTOS	3.24104.14	
Versión 138	... 9/11/2024, 13:28:45	MIGUEL SACRISTAN DE FRUTOS	3.24104.14	
Versión 137	... 9/11/2024, 12:54:21	MIGUEL SACRISTAN DE FRUTOS	3.24104.14	

Figura 7.4: Historial de versiones

Capítulo 8

Seguimiento del proyecto

En esta sección se describe el seguimiento del proyecto, basado en los *sprints* planificados en la sección 2.4. En cada *sprint* se muestra su duración, las tareas a realizar, el tiempo estimado y el empleado, y finalmente, comentarios relevantes sobre lo ocurrido en el *sprint*.

8.0.1. Sprint 0

Aquí incluimos tareas previas al inicio del proyecto:

- Ampliación de conocimientos de Power Apps y Power Automate aplicados al proyecto.
- Creación de un sitio de SharePoint.
- Definición de objetivos de forma abstracta y estudio de la viabilidad.
- Reunión con el tutor para plantear el alcance del proyecto.

8.0.2. Sprint 1

Sprint 1 Semana 1			
Fecha Inicio Sprint: Lunes 30 Septiembre, 2024			
Fecha Fin Sprint: Viernes 4 Octubre, 2024			
Tarea	Horas Estimadas	Horas Empleadas	Estado
Definición de requisitos	4	6	Completa
Diseño de las listas de Sharepoint	4	3	Completa
Implementación de las listas de Sharepoint	2	3	En progreso
Esbozo <i>Pantalla People</i>	2	2	Completa
Análisis de riesgos	8	10	Completa
Elaboración del presupuesto	4	3	Completa

Tabla 8.1: Sprint 1 Semana 1

Sprint 1 Semana 2			
Fecha Inicio Sprint: Lunes 7 Octubre, 2024			
Fecha Fin Sprint: Viernes 11 Octubre, 2024			
Tarea	Horas Estimadas	Horas Empleadas	Estado
Implementación de la interfaz <i>Pantalla People</i>	6	5	Completa
Funcionalidad de la <i>Pantalla People</i>	10	12	En progreso
Diseño del flujo <i>Generar Reporte</i>	2	3	Completa

Tabla 8.2: Sprint 1 Semana 2

En este primer *sprint* se materializaron los riesgos 1 2.3 y 13 2.15, ya que nuestra inexperiencia llevó a estimar mal las horas de trabajo, retrasando la funcionalidad completa de la *Pantalla People*. Sin embargo, logramos que aporte cierta funcionalidad acorde con los objetivos de SCRUM 2.3.

8.0.3. Sprint 2

<p style="text-align: center;">Sprint 2 Semana 3 Fecha Inicio Sprint: Lunes 14 Octubre, 2024 Fecha Fin Sprint: Viernes 18 Octubre, 2024</p>			
Tarea	Horas Estimadas	Horas Empleadas	Estado
Implementación del flujo <i>Generar Reporte</i>	4	6	Completa
Funcionalidad de la <i>Pantalla People</i>	6	8	Completa
Plan de pruebas de la <i>Pantalla People</i>	2	3	Completa
Esbozo de la <i>Pantalla Manager</i>	2	3	Completa
Redacción del capítulo 1 de la memoria	5	4	Completa

Tabla 8.3: Sprint 2 Semana 3

<p style="text-align: center;">Sprint 2 Semana 4 Fecha Inicio Sprint: Lunes 21 Octubre, 2024 Fecha Fin Sprint: Viernes 25 Octubre, 2024</p>			
Tarea	Horas Estimadas	Horas Empleadas	Estado
Implementación de la interfaz de la <i>Pantalla Manager</i>	8	5	Completa
Diseño del flujo <i>Generar Reporte Manager</i>	3	5	Completa
Funcionalidad de la <i>Pantalla Manager</i>	16	4	En progreso

Tabla 8.4: Sprint 2 Semana 4

En este segundo *sprint*, debido a una usencia por vacaciones durante la segunda semana, se ha podido dedicar poco tiempo al desarrollo de la aplicación, ocasionando retrasos en la planificación y

materializándose el riesgo 4 2.6. Sin embargo ya se ha conseguido la funcionalidad completa de la *Pantalla People*. También se tiene implementada la interfaz visual de la *Pantalla Manager* y una parte de su funcionalidad.

8.0.4. Sprint 3

<p style="text-align: center;">Sprint 3 Semana 5 Fecha Inicio Sprint: Lunes 28 Octubre, 2024 Fecha Fin Sprint: Viernes 1 Noviembre, 2024</p>			
Tarea	Horas Estimadas	Horas Empleadas	Estado
Funcionalidad de la <i>Pantalla Manager</i>	12	14	Completa
Implementación del flujo <i>Generar Reporte Manager</i>	5	7	Completa
Plan de pruebas de la <i>Pantalla Principal Manager</i>	3	3	Completa

Tabla 8.5: Sprint 3 Semana 5

<p style="text-align: center;">Sprint 3 Semana 6 Fecha Inicio Sprint: Lunes 4 Noviembre, 2024 Fecha Fin Sprint: Viernes 8 Noviembre, 2024</p>			
Tarea	Horas Estimadas	Horas Empleadas	Estado
Redacción del capítulo 2 de la memoria	8	7	Completa
Esbozo de la <i>Pantalla Empleado</i>	3	3	Completa
Implementación de la interfaz <i>Pantalla Empleado</i>	6	6	Completa

Tabla 8.6: Sprint 3 Semana 6

Seguimiento del proyecto

Este tercer *sprint* se ha conseguido la funcionalidad completa de la *Pantalla Manager* en la primera semana y se ha desarrollado la interfaz de la *Pantalla Empleado*, cuya funcionalidad será implementada en el siguiente *sprint*.

8.0.5. Sprint 4

Sprint 4 Semana 7			
Fecha Inicio Sprint: Lunes 11 Noviembre, 2024			
Fecha Fin Sprint: Viernes 15 Noviembre, 2024			
Tarea	Horas Estimadas	Horas Empleadas	Estado
Funcionalidad de la <i>Pantalla Empleado</i>	20	22	Completa
Diseño del flujo <i>Envío de notificaciones</i>	4	3	En progreso

Tabla 8.7: Sprint 4 Semana 7

Sprint 4 Semana 8			
Fecha Inicio Sprint: Lunes 18 Noviembre, 2024			
Fecha Fin Sprint: Viernes 22 Noviembre, 2024			
Tarea	Horas Estimadas	Horas Empleadas	Estado
Diseño del flujo <i>Envío de notificaciones</i>	1	1	Completa
Implementación del flujo <i>Envío de notificaciones</i>	6	5	Completa
Plan de pruebas de la <i>Pantalla Empleado</i>	4	5	Completa
Revisión <i>Pantalla People</i>	2	1	Completa
Revisión <i>Pantalla Manager</i>	2	2	Completa
Redacción del capítulo 3 de la memoria	8	7	En progreso

Tabla 8.8: Sprint 4 Semana 8

Para el *sprint 4* se ha completado la *Pantalla Empleado* con funcionalidad completa y se ha implementado el último flujo de Power Automate, también se ha realizado una revisión de las pantallas previas a fin de revisar la funcionalidad y hacer la estética uniforme, finalmente, se ha avanzado en la redacción del capítulo 3 de la memoria.

8.0.6. Sprint 5

<p style="text-align: center;">Sprint 5 Semana 9 Fecha Inicio Sprint: Lunes 25 Noviembre, 2024 Fecha Fin Sprint: Viernes 29 Noviembre, 2024</p>			
Tarea	Horas Estimadas	Horas Empleadas	Estado
Redacción del capítulo 3 de la memoria	1	1	Completa
Esbozo de la <i>Pantalla Login</i>	2	1	Completa
Implementación de la interfaz <i>Pantalla Login</i>	2	3	Completa
Funcionalidad de la <i>Pantalla Login</i>	5	7	Completa
Plan de pruebas de la <i>Pantalla Login</i>	2	2	Completa
Redacción del capítulo 4 de la memoria	7	6	En progreso
Realización de los diagramas utilizados	5	6	En progreso

Tabla 8.9: Sprint 5 Semana 9

Seguimiento del proyecto

<p style="text-align: center;">Sprint 5 Semana 10 Fecha Inicio Sprint: Lunes 2 Diciembre, 2024 Fecha Fin Sprint: Viernes 6 Diciembre, 2024</p>			
Tarea	Horas Estimadas	Horas Empleadas	Estado
Redacción del capítulo 4 de la memoria	1	1	Completa
Realización de los diagramas utilizados	2	3	Completa
Redacción del capítulo 5 de la memoria	8	6	En progreso
Modificación de la <i>Pantalla Empleado</i>	1	1	Completa
Modificación de la <i>Pantalla Manager</i>	1	1	Completa
Modificación de la <i>Pantalla People</i>	1	1	Completa
Revisión de la <i>Pantalla Empleado</i>	2	3	Completa
Revisión de la <i>Pantalla Login</i>	2	1	Completa

Tabla 8.10: Sprint 5 Semana 10

A lo largo de este *sprint*, los objetivos han sido, acabar con la funcionalidad de la aplicación, añadiendo una pantalla que permita loguearse a los diferentes usuarios, añadiendo los botones que permiten acceder a esta funcionalidad en las otras pantallas. Finalmente se ha avanzado en la redacción de los capítulos 4 y 5 de la memoria.

8.0.7. Sprint 6

Sprint 6 Semana 11			
Fecha Inicio Sprint: Lunes 9 Diciembre, 2024			
Fecha Fin Sprint: Viernes 13 Diciembre, 2024			
Tarea	Horas Estimadas	Horas Empleadas	Estado
Elaboración de un plan de pruebas integral	6	5	Completa
Documentación y refinamiento del código	10	15	Completa
Redacción del capítulo 5 de la memoria	3	3	Completa
Redacción del capítulo 6 de la memoria	6	2	En progreso

Tabla 8.11: Sprint 6 Semana 11

Sprint 6 Semana 12			
Fecha Inicio Sprint: Lunes 16 Diciembre, 2024			
Fecha Fin Sprint: Viernes 20 Diciembre, 2024			
Tarea	Horas Estimadas	Horas Empleadas	Estado
Elaboración del manual de usuario	12	10	Completa
Redacción del capítulo 6 de la memoria	6	5	Completa
Redacción del capítulo 7 de la memoria	2	3	Completa

Tabla 8.12: Sprint 6 Semana 12

Este *sprint* final ha consistido sobre todo en la documentación del proyecto. Esto se ha realizado a diferentes niveles, tanto documentando el código, como redactando los capítulos restantes de la memoria, de esta forma conseguimos realizar el proyecto sin necesidad de recurrir al sprint adicional que se había planificado en la sección 2.4.

Capítulo 9

Conclusiones y trabajo futuro

9.1. Conclusiones

El objetivo principal de desarrollar una aplicación para la gestión de horas extras en una empresa se ha cumplido satisfactoriamente. La aplicación cubre todos los requisitos establecidos para una gestión efectiva de los trabajos fuera de horario dentro de una empresa y está lista para ser desplegada en un entorno empresarial real, constituyendo de esta forma una recompensa al duro trabajo dedicado a su desarrollo y documentación.

En primer lugar la aplicación permite registrar tanto guardias como trabajos programados y automatizar su proceso de aprobación y envío para pago a diferentes niveles, proporcionando a la empresa una herramienta eficiente para gestionar estos procesos de forma sencilla. Se ha conseguido también facilitar y optimizar la tarea de aprobación o rechazo al permitir introducir las respuestas más frecuentes de los aprobadores, lo cual mejora su experiencia de usuario.

En segundo lugar, al ser un sitio centralizado, se facilita el proceso de consultar toda la información referente a las horas extras. Esto se ha conseguido a través de un acceso segmentado por rol, la posibilidad de disponer de filtros con los que elegir que incidencias queremos y el poder consultar los detalles de cada una de estas individualmente a través de una interfaz atractiva e intuitiva. El diseño de esta, centrado en la eficiencia, ha tomado especial protagonismo.

En tercer lugar, se ha conseguido desarrollar una herramienta que facilite la comunicación entre los diferentes actores del proceso de aprobación a través de flujos de notificación de correo electrónico y comentarios, asegurando un seguimiento efectivo de la incidencia desde su creación hasta su aprobación. De esta forma se favorece que el proceso sea transparente y efectivo.

Unido a lo anterior, la aplicación proporciona también un mecanismo de modificación de las incidencias en caso de que sean denegadas por los aprobadores, para poder adecuarlas a los requisitos de aprobación sin tener que crear una nueva incidencia.

Adicionalmente, mediante flujos automáticos de Power Automate, permite obtener informes deta-

llados sobre horas extras, importes y fechas, contribuyendo así a la planificación y análisis de costes laborales en la empresa.

También se ha conseguido mantener un registro de las tarifas que puede ser actualizado con nuevos valores así como acceder al histórico a través de una lista de SharePoint sin necesidad de realizar modificaciones en la aplicación.

Finalmente, al disponer de un login, permite que desde una misma cuenta de Microsoft, diferentes usuarios con sus credenciales puedan acceder a la aplicación, consiguiendo que, por ejemplo, esta tarea sensible se pueda realizar desde el mínimo número de equipos posibles. También añade una capa adicional de seguridad en caso de filtración de las claves de Microsoft de algún usuario. Por último, permite que, en caso de no ser Microsoft en entorno predeterminado de una compañía, y que por lo tanto las cuentas de correo de los usuarios no estén vinculadas a las funcionalidades completas de Microsoft 365, es decir que no todas las cuentas tengan las licencias necesarias, estos puedan acceder a la aplicación desde una cuenta común y luego a sus funcionalidades a través de los usuarios particulares de la aplicación.

A nivel personal, este proyecto ha sido especialmente enriquecedor, ya que me ha proporcionado la experiencia de aprender, profundizar y superar desafíos asociados a una tecnología que para mí era nueva y diferente a lo aprendido durante mis estudios en la universidad. Al trabajar con una herramienta low-code, descubrí una nueva perspectiva de desarrollo que exige adaptarse a limitaciones y adaptarse a requisitos del cliente, pero que cuenta con un gran potencial para la creación rápida y eficiente de soluciones.

A pesar de no poder aplicar los conocimientos adquiridos en algunas de las asignaturas como los conceptos de *Programación Orientada a Objetos* dada la naturaleza de la herramienta, otros como los necesarios para la elaboración de la base de datos o la importancia de diseñar una interfaz intuitiva y funcional sí que han sido de vital importancia.

En conclusión, me siento satisfecho con los resultados obtenidos tanto en el desarrollo de la aplicación como en mi propio crecimiento personal y profesional. He logrado cumplir con los objetivos planteados, mejorando la gestión de las horas extras dentro del entorno empresarial, y aportando un valor añadido en términos de automatización y eficiencia. Creo firmemente que las herramientas low-code tienen un gran futuro por su capacidad de desarrollar soluciones en tiempos reducidos y con excelente efectividad.

9.2. Trabajo futuro

Dado que el desarrollo se ha realizado mediante una entrega incremental de funcionalidad, existen diversas áreas de mejora y características adicionales que podrían implementarse para complementar la aplicación.

En primer lugar, sería muy útil un flujo de notificaciones automáticas que recuerde a los Managers las solicitudes que llevan un tiempo prolongado en su panel sin haber sido aprobadas. Este mecanismo

Conclusiones y trabajo futuro

reduciría la posibilidad de que incidencias queden sin revisión, aumentando la eficiencia en la gestión de solicitudes. Además, podría añadirse un icono distintivo en la interfaz que resalte las incidencias pendientes de revisión, facilitando la identificación rápida de elementos atrasados y promoviendo una acción oportuna.

Otra mejora significativa sería la integración con Power BI, que permitiría una visualización avanzada de los informes generados sobre horas extras y costes asociados. Esto facilitaría a los usuarios la obtención de conclusiones a partir de los datos y mejoraría el proceso de toma de decisiones mediante gráficos interactivos y dashboards

De cara a proporcionar un ciclo completo, que abarque desde la creación de la incidencia hasta su aprobación y posterior pago, la aplicación podría integrarse con plataformas de pago como PayPal, disponibles en Power Automate. Esto permitiría automatizar el proceso de pago, simplificando aún más la gestión administrativa y cerrando de manera eficiente el ciclo de cada incidencia.

Además, la aplicación podría beneficiarse de una capa de autenticación multifactor (MFA) que aumente la seguridad y proteja los datos de los usuarios.

Finalmente, sería interesante desplegar la aplicación en un entorno real, lo que permitiría evaluar su eficacia y adaptabilidad a condiciones de uso cotidiano. La implementación en un contexto real brindaría la oportunidad de obtener retroalimentación de los usuarios finales, permitiendo realizar ajustes y optimizaciones basados en su experiencia.

Bibliografía

- [1] H. M. Hilanderas Pizarro, *Desarrollo de Power Virtual Agents y manejo de herramientas de Power Platform para distintos casos de uso*. Universidad de Valladolid, 2022. [Online]. Available: <https://uvadoc.uva.es/handle/10324/57315>
- [2] V. Montequi Rodríguez, *Aplicación para la gestión y seguimiento de dossiers de análisis de riesgos en producción de una empresa industrial*. Universidad de Valladolid, 2022. [Online]. Available: <https://uvadoc.uva.es/handle/10324/57326>
- [3] T. Castaño Sánchez, *Aplicación para la gestión de los equipos de prevención de incendios en una factoría utilizando Microsoft Power Apps*. Universidad de Valladolid, 2023. [Online]. Available: <https://uvadoc.uva.es/handle/10324/62869>
- [4] W. Royce, *Managing the Development of Large Software Systems*, 1970. [Online]. Available: <https://dl.acm.org/doi/10.5555/41765.41801>
- [5] K. Beck, M. Beedle, A. van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R. C. Martin, S. Mellor, K. Schwaber, J. Sutherland, and D. Thomas, "Manifesto for agile software development," 2001. [Online]. Available: <https://agilemanifesto.org/>
- [6] K. Schwaber and J. Sutherland, *The Scrum Guide: The Definitive Guide to Scrum*. Scrum.org, 2020. [Online]. Available: <https://www.scrum.org/resources/scrum-guide>
- [7] Glassdoor, "Salarios de desarrollador junior," 2024. [Online]. Available: https://www.glassdoor.es/Sueldos/junior-software-developer-sueldo-SRCH_KO0,25.htm
- [8] Microsoft, "Microsoft 365 business standard," 2024. [Online]. Available: <https://www.microsoft.com/es-es/microsoft-365/business/microsoft-365-business-standard?activetab=pivot:informaci%C3%B3ngeneral>
- [9] —, "Power apps pricing," 2024. [Online]. Available: <https://www.microsoft.com/es-es/power-platform/products/power-apps/pricing/>
- [10] Amazon, "Monitor philips 243v7qdsb/00 - 24"full hd, tecnología ips," 2024. [Online]. Available: <https://www.amazon.es/Philips-243V7QDSB-00-resoluci%C3%B3n-tecnolog%C3%ADa/dp/B06Y13QBYL/>

- [11] A. Jaafari, "Management of risks, uncertainties and opportunities on projects: time for a fundamental shift," *International Journal of Project Management*, 2001.
- [12] D. Howell, C. Windahl, and R. Seidel, "A project contingency framework based on uncertainty and its consequences," *International Journal of Project Management*, 2010.
- [13] J. Jiang and G. Klein, "Risks to different aspects of system success," *Information & Management*, 1999.
- [14] C. W. Ibbs and Y. H. Kwak, "Assessing project management maturity," *Project Management Journal*, 2000.
- [15] T. W. Kwan and H. K. N. Leung, "A risk management methodology for project risk dependencies," *IEEE Transactions on Software Engineering*, 2011.
- [16] SafetyCulture. (2023) Guía para entender una matriz de riesgo 5 x 5. [Online]. Available: <https://safetyculture.com/es/temas/evaluacion-de-riesgos/matriz-de-riesgo/>
- [17] J. G. March and Z. Shapira, "Managerial perspectives on risk and risk taking," *Management Science*, 1987.
- [18] K. Lyytinen, L. Mathiassen, and J. Ropponen, "Attention shaping and software risk: a categorical analysis of four classical risk management approaches," *Information Systems Research*, 1998.
- [19] M. de Trabajo y Economía Social, "Guía sobre horas extraordinarias: Ministerio de trabajo y economía social," 2024. [Online]. Available: https://www.mites.gob.es/es/Guia/texto/guia_6/contenidos/guia_6_14_3.htm
- [20] I. Jacobson, M. Christerson, P. Jonsson, and G. Overgaard, *Object-Oriented Software Engineering: A Use Case Driven Approach*. Addison-Wesley, 1992.
- [21] A. Cockburn, *Writing Effective Use Cases*. Addison-Wesley, 2001.
- [22] M. Fowler, *UML Distilled: A Brief Guide to the Standard Object Modeling Language*. Addison-Wesley, 2003.
- [23] I. Jacobson, G. Booch, and J. Rumbaugh, *The Unified Software Development Process*. Addison-Wesley, 2004.
- [24] Microsoft, "Power fx overview," <https://learn.microsoft.com/en-us/power-platform/power-fx/overview>, 2021.
- [25] —, "Power apps: Patrones de diseño," <https://learn.microsoft.com/es-es/power-apps/guidance/patterns/overview>, 2024.
- [26] —, "Power apps: Approval pattern," <https://learn.microsoft.com/es-es/power-apps/guidance/patterns/approval-pattern>, 2024.
- [27] —, "Power apps: Inspection pattern," <https://learn.microsoft.com/es-es/power-apps/guidance/patterns/inspection-pattern>, 2024.

- [28] —, “Power apps: More patterns,” <https://learn.microsoft.com/es-es/power-apps/guidance/patterns/more-patterns>, 2024.
- [29] I. O. for Standardization, “Ergonomics of human-system interaction — part 11: Usability: Definitions and concepts,” <https://www.iso.org/standard/63500.html>, 2018.
- [30] J. Nielsen, *Usability Engineering*. Morgan Kaufmann, 1994.
- [31] J. Brech, “Web usability definition,” <https://www.webwisewording.com/usability-definition/>, 2020.
- [32] Microsoft, “Power apps: Overview,” <https://learn.microsoft.com/en-us/power-apps/powerapps-overview>, 2024.
- [33] —, “Power apps: Connectors,” <https://learn.microsoft.com/en-us/power-apps/maker/canvas-apps/add-manage-connections>, 2024.
- [34] —, “Power apps: Reducing costs,” <https://www.microsoft.com/en-us/power-platform/blog/2024/09/03/reduce-development-times-and-increase-roi-with-microsoft-power-platform/>, 2024.
- [35] —, “Power apps: User experience,” <https://learn.microsoft.com/en-us/power-apps/user/unified-interface>, 2024.
- [36] —, “Power apps: Integration with services,” <https://learn.microsoft.com/en-us/connectors/connector-reference/connector-reference-powerapps-connectors>, 2024.
- [37] —, “Power apps: Multi-platform deployment,” <https://learn.microsoft.com/es-es/power-apps/limits-and-config>, 2024.
- [38] —, “Power apps: Automating workflows,” <https://learn.microsoft.com/en-us/power-automate/create-flows-power-apps>, 2024.
- [39] —, “Power apps: Limitations,” <https://learn.microsoft.com/es-es/power-apps/limits-and-config>, 2024.
- [40] M. Community, “Power apps: Scalability considerations,” <https://community.dynamics.com/blogs/post/?postid=2e341401-07a2-4bfb-9029-3e8451215c8d>, 2024.
- [41] Microsoft, “Lenguaje de programación power fx,” <https://learn.microsoft.com/es-es/power-platform/power-fx/overview>, 2024.
- [42] —, “Power fx: Data types,” <https://docs.microsoft.com/en-us/power-platform/power-fx/data-types>, 2024.
- [43] DataOps, “Declarative vs imperative programming,” <https://www.dataops.live/the-data-engineers-guide-to-declarative-vs-imperative-for-data>, 2024.
- [44] Microsoft, “Create and edit forms in power apps,” <https://learn.microsoft.com/es-es/power-apps/maker/model-driven-apps/create-and-edit-forms>, 2024.
- [45] —, “Patch function in powerapps,” <https://learn.microsoft.com/en-us/powerapps/maker/canvas-apps/functions/function-patch>, 2024.

- [46] —, “Filter and lookup functions in powerapps,” <https://learn.microsoft.com/es-es/power-platform/power-fx/reference/function-filter-lookup>, 2024.
- [47] —, “Navigate function in powerapps,” <https://learn.microsoft.com/en-us/powerapps/maker/canvas-apps/functions/function-navigate>, 2024.
- [48] —, “Create and update collections in power apps,” <https://learn.microsoft.com/es-es/power-apps/maker/canvas-apps/create-update-collection>, 2024.
- [49] —, “What is sharepoint?” <https://support.microsoft.com/en-us/sharepoint>, 2024.
- [50] —, “Create and manage workflows in sharepoint,” <https://support.microsoft.com/es-es/office/introducci%C3%B3n-al-flujo-de-trabajo-de-sharepoint-07982276-54e8-4e17-8699-5056eff4d9e3>, 2024.
- [51] —, “Integrate sharepoint with other applications,” <https://learn.microsoft.com/en-us/power-apps/maker/canvas-apps/sharepoint/scenarios-intro>, 2024.
- [52] —, “What is power automate?” <https://learn.microsoft.com/en-us/power-automate/>, 2024.
- [53] —, “Getting started with power automate,” <https://learn.microsoft.com/en-us/power-automate/getting-started>, 2024.
- [54] —, “Integrating power automate with other microsoft tools,” <https://www.microsoft.com/en-us/power-platform/connectors>, 2024.
- [55] —, “Ai builder overview,” <https://learn.microsoft.com/en-us/ai-builder/overview>, 2024.
- [56] —, “List of all power automate connectors,” <https://learn.microsoft.com/en-us/connectors/>, 2024.
- [57] —, “Version control for power apps with git,” <https://learn.microsoft.com/es-es/power-apps/maker/canvas-apps/git-version-control>, 2024.
- [58] —, “Multifactor authentication (mfa),” <https://learn.microsoft.com/en-us/entra/identity/authentication/>.
- [59] P. E. y Consejo de la Unión Europea, “Reglamento general de protección de datos (rgpd) 2016/679,” 2016. [Online]. Available: <https://eur-lex.europa.eu/legal-content/ES/TXT/?uri=CELEX%3A32016R0679>
- [60] Microsoft, “Microsoft compliance center: Ensuring regulatory compliance,” <https://learn.microsoft.com/en-us/purview/purview>.
- [61] I. Burnstein, *Practical Software Testing: A Process-Oriented Approach*. Springer Science & Business Media, 2003, ISBN: 978-0387951317.
- [62] Microsoft, “Guardar y publicar aplicaciones de lienzo en power apps,” 2024. [Online]. Available: <https://learn.microsoft.com/es-es/power-apps/maker/canvas-apps/save-publish-app>

Todos los enlaces han sido comprobados correctamente el 20 de diciembre de 2024.

Apéndices

Apéndice A

Manual de usuario

Previo al inicio de la aplicación es obligatorio disponer de todos los permisos en la cuenta de Microsoft y disponer de un usuario y contraseña registrado en el sistema.

A.1. Pantalla de Carga

Durante su visualización, se lleva a cabo el cargado de datos necesarios para la aplicación. Posteriormente se mostrará la Pantalla de Inicio.



Figura A.1: Animación Pantalla de Carga

A.2. Pantalla de Inicio

El primer paso para acceder a la aplicación es autenticarse, para ello el usuario tendrá que introducir su usuario y hacer click en el botón de *Entrar*. Una vez validado que el email existe, se le enviará un correo electrónico en el que validará el acceso.

En el caso de que el usuario no exista o sea incorrecto, el sistema lo notificará y pedirá que vuelva a introducir el email de nuevo.

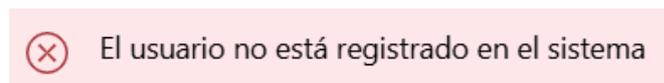


Figura A.2: Notificación de error



Figura A.3: Pantalla de Inicio

A.3. Pantalla People

El rol de People puede ver un listado de registros que han sido aprobados por Managers anteriormente. Se ven los datos de la fecha de aprobación, el tipo de registro, el Empleado que lo emitió, el Manager que lo aprobó y el importe calculado.

Fecha aprobación	Tipo registro	Identificador	Estado	Emitido por	Aprobado por	Importe
<input type="checkbox"/> 27/10/2024	Guardia	32	✔ Aprobada	PRUEBA2, USUARIO	SACRISTAN DE FRUTOS, MIGUEL	340,00€ Detalles
<input type="checkbox"/> 27/10/2024	Trabajo programado	31	✔ Aprobada	PRUEBA2, USUARIO	SACRISTAN DE FRUTOS, MIGUEL	67,50€ Detalles
<input type="checkbox"/> 27/10/2024	Guardia	14	✔ Aprobada	SACRISTAN DE FRUTOS, MIGUEL	SACRISTAN DE FRUTOS, MIGUEL	€ Detalles
<input type="checkbox"/> 22/10/2024	Trabajo programado	28	✔ Aprobada	PRUEBA1, USUARIO	SACRISTAN DE FRUTOS, MIGUEL	67,50€ Detalles
<input type="checkbox"/> 22/10/2024	Guardia	25	✔ Aprobada	PRUEBA2, USUARIO	SACRISTAN DE FRUTOS, MIGUEL	€ Detalles
<input type="checkbox"/> 02/10/2024	Guardia	2	✔ Aprobada	SUAREZ GONZALEZ, ROBERTO	LOPEZ FERNANDEZ, JUAN	17,00€ Detalles
<input type="checkbox"/> 01/10/2024	Guardia	1	✔ Aprobada	SUAREZ GONZALEZ, ROBERTO	LOPEZ FERNANDEZ, JUAN	13,00€ Detalles

Figura A.4: Pantalla de People

A.3.1. Envío registro a pago

El Personal de People puede enviar a pago los registros. Para ello selecciona los registros en la galería y posteriormente pulsa en el botón *Enviar* que aparecerá a la derecha abajo tras seleccionar algún registro. Tras clickar sobre el botón de envío se le pedirá una confirmación. Tanto el empleado como el Manager serán informados de esta acción mediante un correo.

<input checked="" type="checkbox"/>	01/10/2024	Guardia	1	✔ Aprobada	SUAREZ GONZALEZ, ROBERTO	LOPEZ FERNANDEZ, JUAN	13,00€ Detalles
<input checked="" type="checkbox"/>	01/10/2024	Guardia	3	✔ Aprobada	SUAREZ GONZALEZ, ROBERTOAAA	LOPEZ FERNANDEZ, IUANAAAA	17,00€ Detalles

Enviar
Rechazar

Figura A.5: Botón enviar para pago

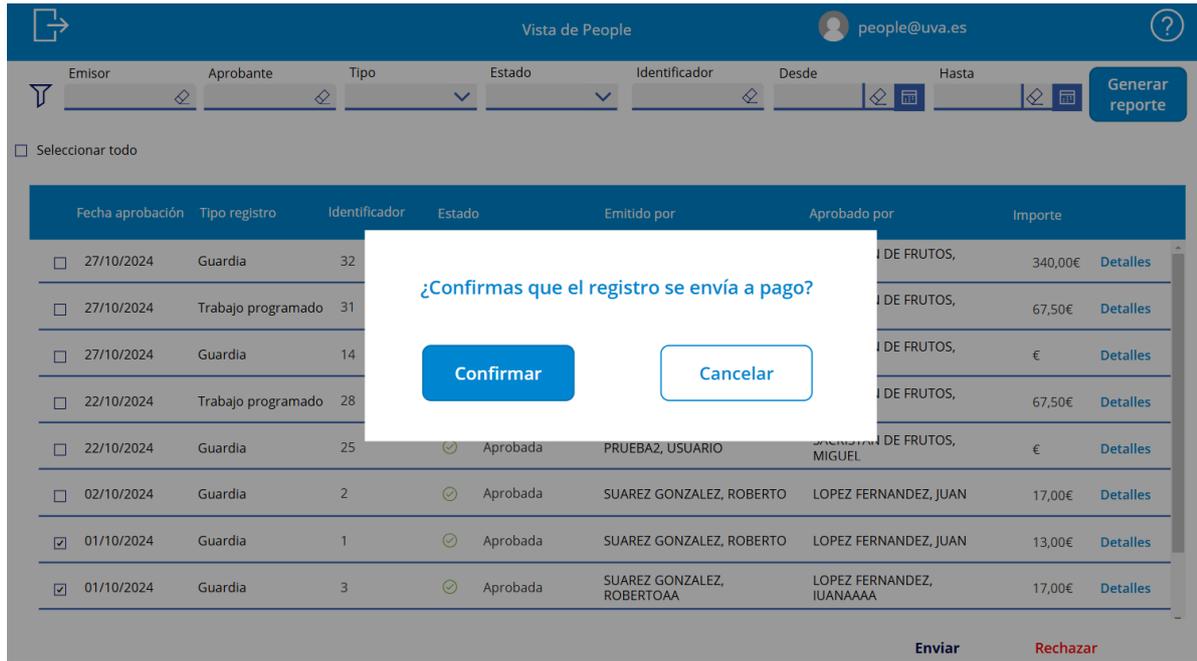


Figura A.6: Envío para pago

A.3.2. Rechazo de registro

El Personal de People puede rechazar registros seleccionándolos en la galería y posteriormente pulsando en el botón *Rechazar* que aparecerá a la derecha abajo tras seleccionar algún registro. Tras hacer click sobre el botón de rechazo se le pedirá una confirmación y se le ofrecerá la posibilidad de modificar los comentarios de las incidencias seleccionadas. Además, dispondrá de un desplegable en el que salgan sus 5 respuestas más utilizadas, en caso de querer usar una respuesta predeterminada. Tanto el empleado como el Manager serán informados de esta acción mediante un correo.

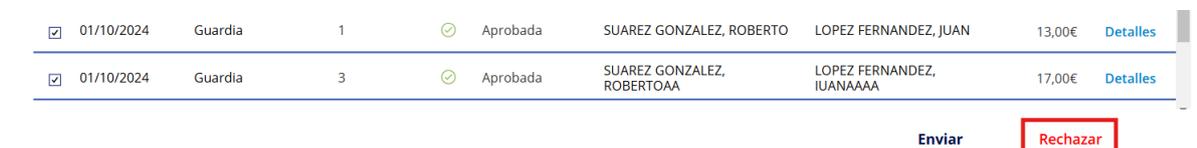


Figura A.7: Botón rechazar registro

Fecha emision	Tipo registro	Identificador	Estado	Emitido por	Importe	
<input type="checkbox"/> 27/10/2024	Guardia	32	<input checked="" type="checkbox"/> Aprobada	PRUEBA2, USUARIO	340,00€	Detalles
<input type="checkbox"/> 27/10/2024	Trabajo programado	31	<input checked="" type="checkbox"/> Aprobada	PRUEBA2, USUARIO	67,50€	Detalles
<input type="checkbox"/> 26/09/2024	Guardia	14	<input checked="" type="checkbox"/> Aprobada	SACRISTÁN DE FRUTOS, MIGUEL	€	Detalles
<input type="checkbox"/> 22/10/2024	Trabajo programado	28	<input checked="" type="checkbox"/> Aprobada	PRUEBA1, USUARIO	67,50€	Detalles
<input type="checkbox"/> 22/10/2024	Guardia	25	<input checked="" type="checkbox"/> Aprobada	PRUEBA2, USUARIO	€	Detalles
<input type="checkbox"/> 21/05/2024	Guardia	2	<input checked="" type="checkbox"/> Aprobada	SUAREZ GONZALEZ, ROBERTO	17,00€	Detalles
<input type="checkbox"/> 27/05/2024	Guardia	1	<input checked="" type="checkbox"/> Aprobada	SUAREZ GONZALEZ, ROBERTO	13,00€	Detalles
<input type="checkbox"/> 09/05/2024	Guardia	3	<input checked="" type="checkbox"/> Aprobada	SUAREZ GONZALEZ, ROBERTOAA	17,00€	Detalles

Figura A.10: Pantalla Manager

A.4.1. Aprobación de incidencia

El Manager puede aprobar incidencias seleccionándolas en la galería y posteriormente pulsando en el botón *Aprobar* que aparecerá a la derecha abajo tras seleccionar uno o más registros. Tras pinchar sobre ese botón, se le pedirá una confirmación, y se le ofrecerá la posibilidad de modificar los comentarios de las incidencias seleccionadas. Al igual que en el caso anterior, dispondrá de un desplegable en el que podrá elegir, en caso de desearlo, una de sus 5 respuestas más utilizadas. El Empleado será informado por correo electrónico de que su registro ha sido aprobado.

<input checked="" type="checkbox"/>	26/09/2024	Trabajo programado	18	→ Enviada	SACRISTÁN DE FRUTOS, MIGUEL	495,00€	Detalles
<input checked="" type="checkbox"/>	26/09/2024	Trabajo programado	19	→ Enviada	SACRISTÁN DE FRUTOS, MIGUEL	52,50€	Detalles

Aprobar
Rechazar
Cancelar

Figura A.11: Botón aprobar incidencia

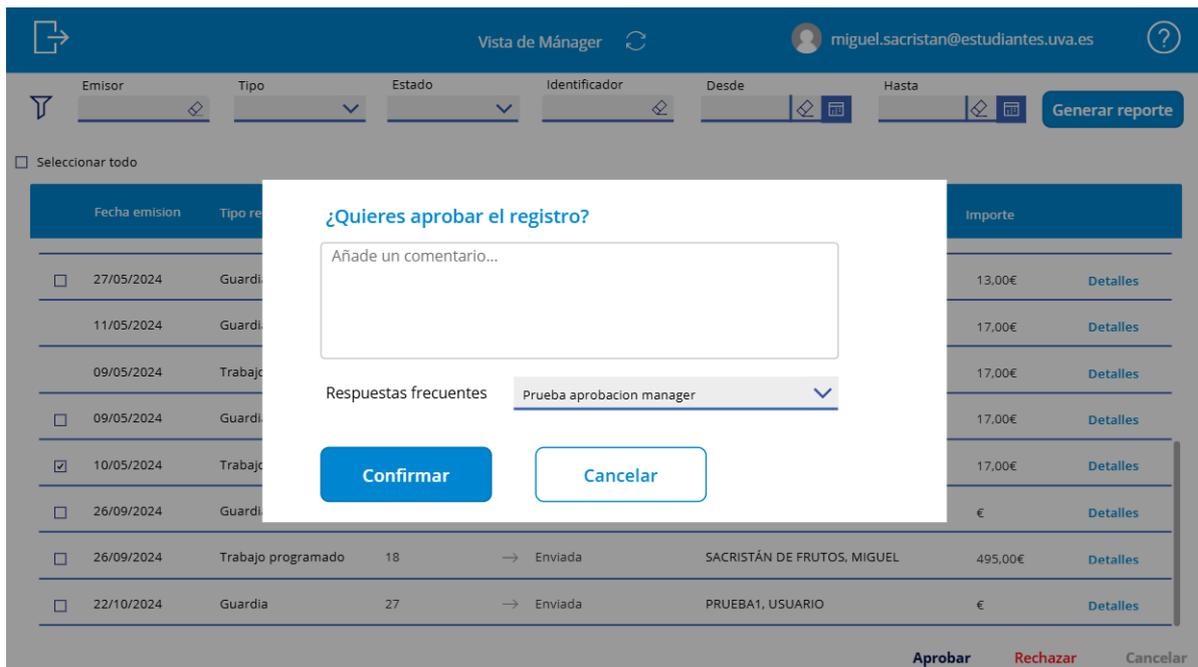


Figura A.12: Aprobación incidencia

A.4.2. Rechazo de incidencia

El Manager puede rechazar registros seleccionándolos en la galería y posteriormente pulsando en el botón *Rechazar* A.11 que aparecerá a la derecha abajo tras seleccionar algún registro. Tras clicar sobre el botón de rechazo se le pedirá una confirmación y se le ofrecerá la posibilidad de modificar los comentarios de las incidencias seleccionadas. Al igual que en el caso anterior, existirá un desplegable en el que el Manager pueda elegir entre sus respuestas de rechazo más frecuentes. El empleado será informado de esta acción mediante un correo.

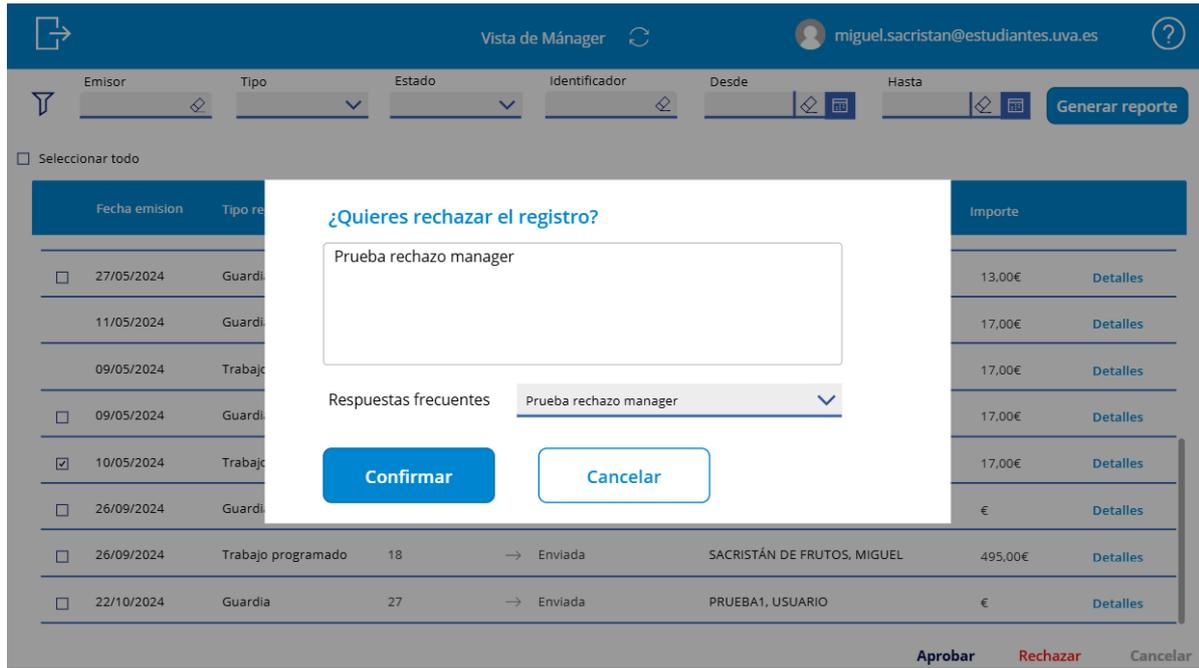


Figura A.13: Rechazo de una incidencia

A.4.3. Cancelación de incidencia

El Manager puede cancelar registros que ya hayan sido enviados al Personal de People seleccionándolos en la galería y después haciendo click en el botón de *Cancelar* que se mostrará abajo tras seleccionar algún registro. Los las incidencias que cancele deberán estar en el estado *Enviada*. Tras esto, aparece una confirmación y el Empleado es notificado por correo que la incidencia ha sido cancelada por el Manager.

<input checked="" type="checkbox"/>	22/10/2024	Trabajo programado	28	<input checked="" type="checkbox"/>	Aprobada	PRUEBA1, USUARIO	67,50€	Detalles
<input checked="" type="checkbox"/>	22/10/2024	Guardia	25	<input checked="" type="checkbox"/>	Aprobada	PRUEBA2, USUARIO	€	Detalles

Aprobar Rechazar **Cancelar**

Figura A.14: Botón Cancelar para confirmar

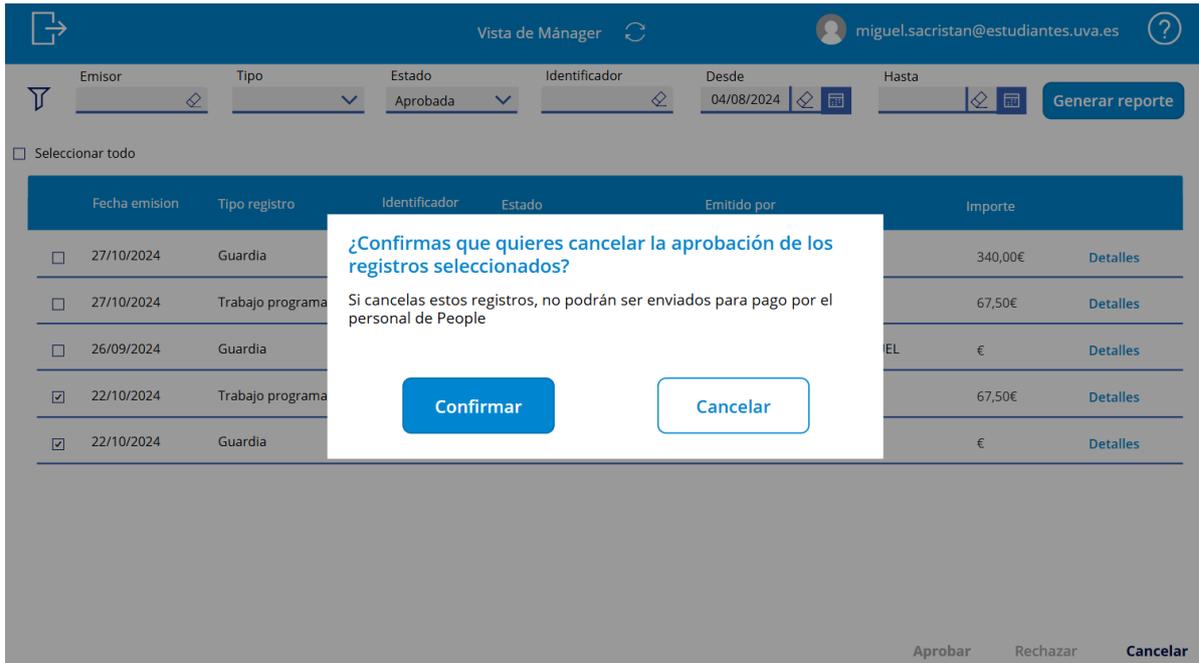


Figura A.15: Confirmar cancelacion

A.4.4. Cambio de pantalla de Manager a Empleado

Debido a que puede ocurrir que un Manager, a parte de gestionar las incidencias de sus empleados a cargo, también tenga la necesidad de crear y tramitar sus incidencias (que serán aprobadas por su responsable), se puede cambiar a su vista de empleado pulsando el siguiente botón:



Figura A.16: Rechazo de una incidencia

A.4.5. Generar reporte Manager

Los Managers pueden generar un reporte con los datos de las incidencias que hayan recibido dentro de un marco de fechas que elijan. Para ello deben hacer click en el botón *Generar reporte*:

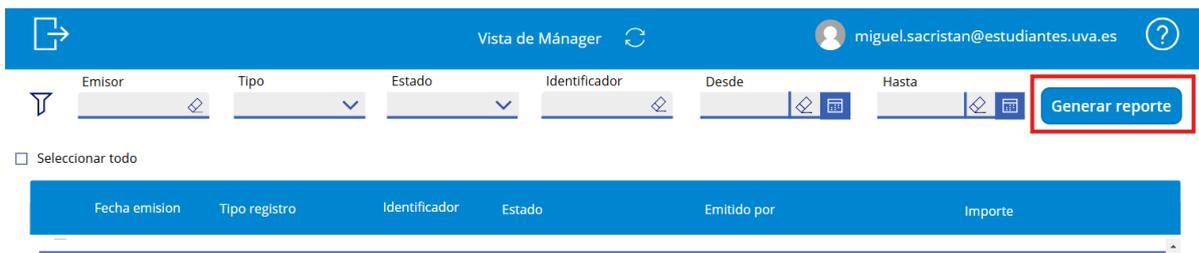


Figura A.17: Botón generar reporte Manager

Esto abrirá una ventana con dos campos para introducir fechas. Éstas serán las fechas que delimiten los registros que saldrán en el reporte.



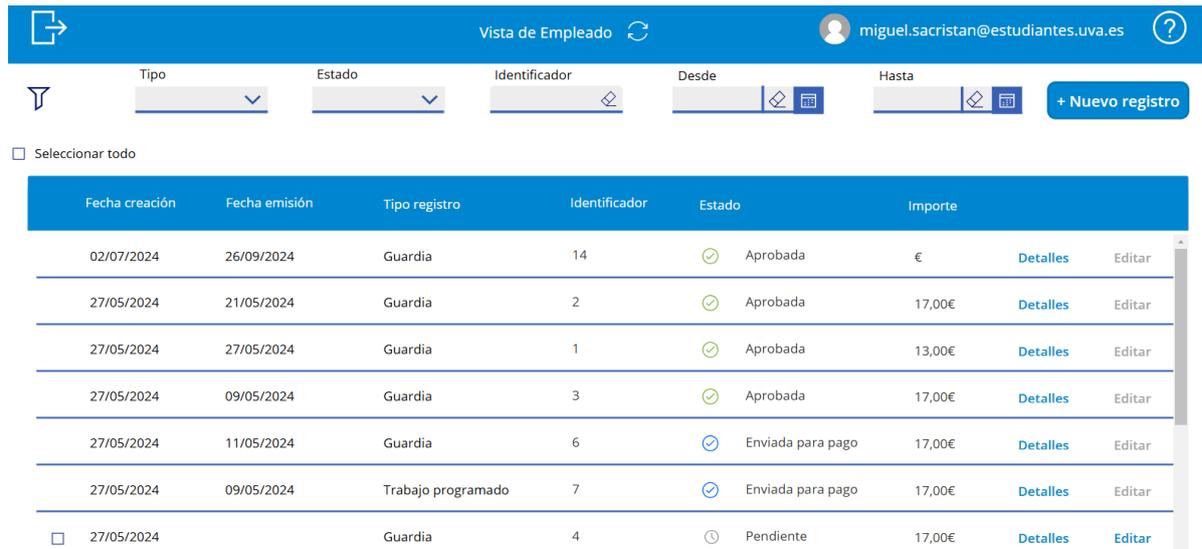
El formulario de reporte está diseñado con un encabezado azul que contiene el texto "Generar reporte" y un ícono de cerrar (una 'X' dentro de un círculo). Debajo de este encabezado, hay dos campos de entrada de texto, uno etiquetado "Desde" y otro "Hasta". Cada campo tiene un ícono de calendario a su derecha para facilitar la selección de fechas. Al final del formulario, hay un botón azul con el texto "Confirmar".

Figura A.18: Formulario de reporte

Tras introducir las fechas y confirmar, se enviará por correo un archivo .csv con la información recuperada. Se recomienda abrir este archivo desde la aplicación de Excel para visualizar los datos de una forma más clara.

A.5. Pantalla Empleado

Desde esta pantalla el empleado puede ver el listado de todos los registros que ha gestionado anteriormente para comprobar el estado en el que están: *Aprobada*, *Enviada*, *Pendiente* o *Enviada para Pago*. También estará habilitado el botón *Nuevo registro*.



The screenshot shows the 'Empleado' screen with a blue header bar containing a home icon, the text 'Vista de Empleado', a refresh icon, a user profile icon for 'miguel.sacristan@estudiantes.uva.es', and a help icon. Below the header is a filter bar with a funnel icon, dropdown menus for 'Tipo', 'Estado', and 'Identificador', date pickers for 'Desde' and 'Hasta', and a '+ Nuevo registro' button. A checkbox labeled 'Seleccionar todo' is positioned above the table. The table has columns for 'Fecha creación', 'Fecha emisión', 'Tipo registro', 'Identificador', 'Estado', and 'Importe'. Each row includes 'Detalles' and 'Editar' links. The data rows are as follows:

Fecha creación	Fecha emisión	Tipo registro	Identificador	Estado	Importe		
02/07/2024	26/09/2024	Guardia	14	✓ Aprobada	€	Detalles	Editar
27/05/2024	21/05/2024	Guardia	2	✓ Aprobada	17,00€	Detalles	Editar
27/05/2024	27/05/2024	Guardia	1	✓ Aprobada	13,00€	Detalles	Editar
27/05/2024	09/05/2024	Guardia	3	✓ Aprobada	17,00€	Detalles	Editar
27/05/2024	11/05/2024	Guardia	6	✓ Enviada para pago	17,00€	Detalles	Editar
27/05/2024	09/05/2024	Trabajo programado	7	✓ Enviada para pago	17,00€	Detalles	Editar
<input type="checkbox"/>	27/05/2024	Guardia	4	⌚ Pendiente	17,00€	Detalles	Editar

Figura A.19: Pantalla Empleado

Es posible filtrar la lista de incidencias que se visualizan desde el menú superior:



Figura A.20: Filtros de la Pantalla de Empleado

A.5.1. Nuevo registro

Quando se haga click sobre el botón *Nuevo registro*, aparecerá una ventana para seleccionar el tipo de registro que quiere añadir: *Guardia* o *Trabajo programado*.

← Nuevo registro

Tipo de registro
Guardia

Fecha de inicio

Festivos
1

Intervenciones

Horas totales

Comentarios
Añade un comentario...

Cancelar Guardar

Figura A.21: Nuevo registro

Una vez creado, estará en el estado *Pendiente* hasta ser confirmado.

Formulario de Guardia

Para la Guardia, se debe introducir la fecha en la que inició la guardia, el número de festivos en la Guardia y un comentario. Si alguno de estos campos estuviera vacío, se le notifica al usuario que se deben rellenar. Además, se han de incluir las intervenciones llevadas a cabo durante esta, introduciendo la fecha de cada una, las horas dedicadas a ella y un comentario opcional. El número total de horas se calculará automáticamente a partir de las horas de las intervenciones. Por último, se guarda el registro.

Fecha creación	Fecha emisión	Tipo registro	Identificador	
<input type="checkbox"/>	11/07/2024	Guardia	12	
<input type="checkbox"/>	11/07/2024	Trabajo programado	17	
<input type="checkbox"/>	11/07/2024	26/09/2024	Trabajo programado	18
<input type="checkbox"/>	26/09/2024	26/09/2024	Trabajo programado	19
<input type="checkbox"/>	26/09/2024	Trabajo programado	20	
<input type="checkbox"/>	02/10/2024	Guardia	21	
<input type="checkbox"/>	03/10/2024	Trabajo programado	22	
<input type="checkbox"/>	04/10/2024	27/10/2024	Guardia	23

Figura A.22: Formulario de Guardia

Fecha	Horas	Comentarios
26/08/2024	2	
29/08/2024	1	
27/08/2024	1	

Figura A.23: Formulario de Intervenciones

Formulario de Trabajo programado

Para la creación de un Trabajo programado, se debe indicar tanto la fecha en la que tuvo lugar, como las horas de inicio y fin, así como un comentario. Si alguno de estos campos estuviera vacío, se le notifica al usuario que se deben rellenar. El número de horas correspondiente al trabajo programado se calculará automáticamente. Una vez completados estos campos, se le da al botón de *Guardar*.

Figura A.24: Formulario de Trabajo programado

A.5.2. Detalles de un registro

Es posible ver los detalles de cada uno de los registros pulsando en el botón *Detalles* que hay a la derecha de cada una de las líneas de registro.

Una vez pulsamos en los detalles, podemos ver detalles tanto del emisor como del aprobante, como el nombre y matricula. También aparecen datos del registro como el día de inicio y fin en el caso de ser una guardia, o el día en que tuvo lugar en el caso de un trabajo programado, las horas trabajadas y el importe calculado.

Fecha creación	Fecha emisión	Tipo registro	Identificador	Estado	Importe	Acciones
02/07/2024	26/09/2024	Guardia	14	✓ Aprobada	€	Detalles Editar
27/05/2024	21/05/2024	Guardia	2	✓ Aprobada	17,00€	Detalles Editar
27/05/2024	27/05/2024	Guardia	1	✓ Aprobada	13,00€	Detalles Editar
27/05/2024	09/05/2024	Guardia	3	✓ Aprobada	17,00€	Detalles Editar
27/05/2024	11/05/2024	Guardia	6	✓ Enviada para pago	17,00€	Detalles Editar

Figura A.25: Botón de detalles



Emisor

Nombre

PRUEBA2, USUARIO

Matrícula

4444

Aprobante

Nombre

SACRISTÁN DE FRUTOS, MIGUEL

Matrícula

1234

Guardia

Fecha de inicio

02/09/2024



Horas

3

Intervenciones



Fecha de fin

09/09/2024



Festivos

1

Importe

340€

Comentarios

Prueba integrada: Crear Guardia

Figura A.26: Detalles de un registro

A.5.3. Edición de un registro

Si un registro no ha sido aún enviado al Manager, puede ser editado por el Empleado. Para ello se hace clic en el botón *Editar* que hay a la derecha de cada uno de los registros. Los campos editables serán los mismos que desde la ventana de creación de nuevo registro, vistos en las figuras A.22 y A.24.

Vista de Empleado 🔄 miguel.sacristan@estudiantes.uva.es ?

Tipo Estado Identificador Desde Hasta + Nuevo registro

Seleccionar todo

Fecha creación	Fecha emisión	Tipo registro	Identificador	Estado	Importe	Detalles	Editar
27/05/2024	11/05/2024	Guardia	6	🕒 Enviada para pago	17,00€	Detalles	Editar
27/05/2024	09/05/2024	Trabajo programado	7	🕒 Enviada para pago	17,00€	Detalles	Editar
<input type="checkbox"/> 27/05/2024		Guardia	4	🕒 Pendiente	17,00€	Detalles	Editar
<input type="checkbox"/> 27/05/2024	10/05/2024	Trabajo programado	5	→ Enviada	17,00€	Detalles	Editar
<input type="checkbox"/> 11/07/2024		Guardia	12	🕒 Pendiente	€	Detalles	Editar
<input type="checkbox"/> 11/07/2024		Trabajo programado	17	🕒 Pendiente	142,50€	Detalles	Editar

Figura A.27: Editar registro

A.5.4. Enviar un registro

Después de crear el registro, para confirmar su envío al Manager hay que seleccionar el registro haciendo click en su casilla de la izquierda correspondiente. Se pueden seleccionar varios registros a la vez y enviarlos. Todos los registros seleccionados deben tener el mismo estado para poder realizar una operación sobre ellos. Al seleccionar los registros aparecerán los botones de las operaciones abajo a la derecha, hay que hacer click sobre *Enviar* y aceptar la confirmación que aparece después. Una vez confirmada la acción, se enviará el registro y el Manager asociado recibirá un correo informándole de ello.

En caso de querer usar la opción de *Seleccionar todo*, se debe filtrar la galería por estado antes de marcar la casilla.

<input checked="" type="checkbox"/>	27/05/2024		Guardia	4	🕒 Pendiente	17,00€	Detalles	Editar
<input type="checkbox"/>	27/05/2024	10/05/2024	Trabajo programado	5	→ Enviada	17,00€	Detalles	Editar
<input checked="" type="checkbox"/>	11/07/2024		Guardia	12	🕒 Pendiente	€	Detalles	Editar
<input checked="" type="checkbox"/>	11/07/2024		Trabajo programado	17	🕒 Pendiente	142,50€	Detalles	Editar

Enviar Cancelar 🗑️

Figura A.28: Botón Enviar para confirmar

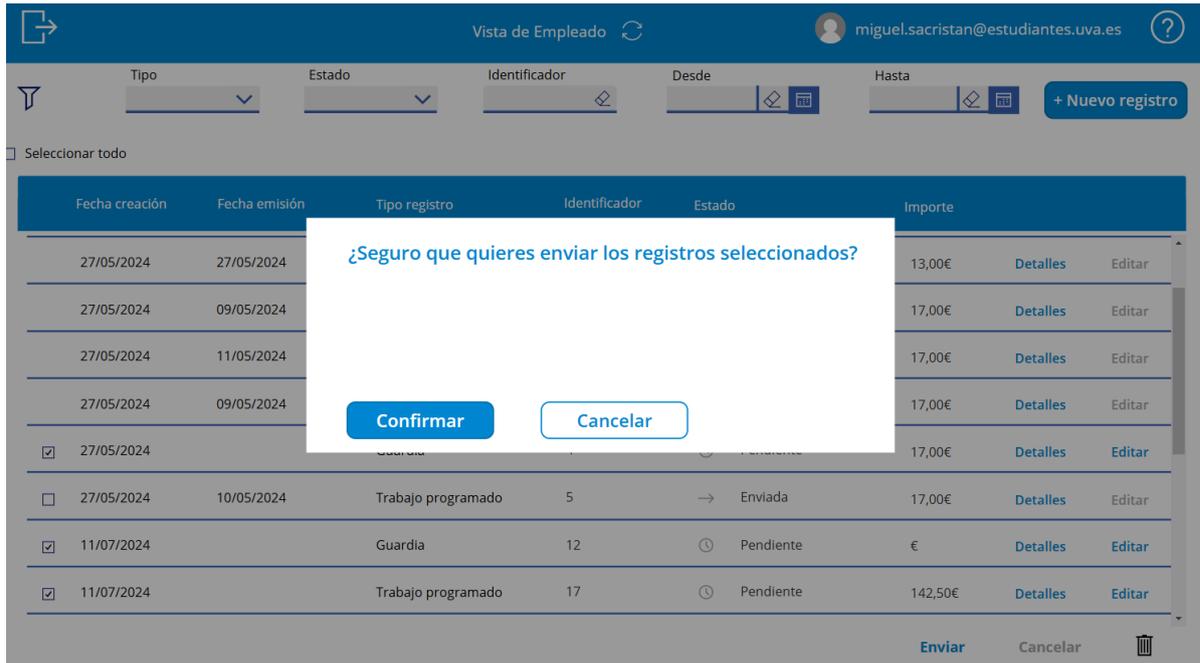


Figura A.29: Confirmar envío

A.5.5. Cancelación de un registro

Los registros que ya hayan sido enviados, pero que todavía no han sido aprobados por el Manager, pueden ser cancelados por el Empleado. Todas las incidencias tienen que tener el mismo estado *Enviada*. Una vez seleccionados, aparecerá en la esquina inferior derecha el botón de *Cancelar*, se hace click sobre él y se acepta la confirmación. Una vez que confirmamos, se notifica al Manager que el registro ha sido cancelado.

Al igual que en el caso anterior, antes de usar la opción de *Seleccionar todo*, se debe filtrar la galería por estado.



Figura A.30: Botón Cancelar para confirmar

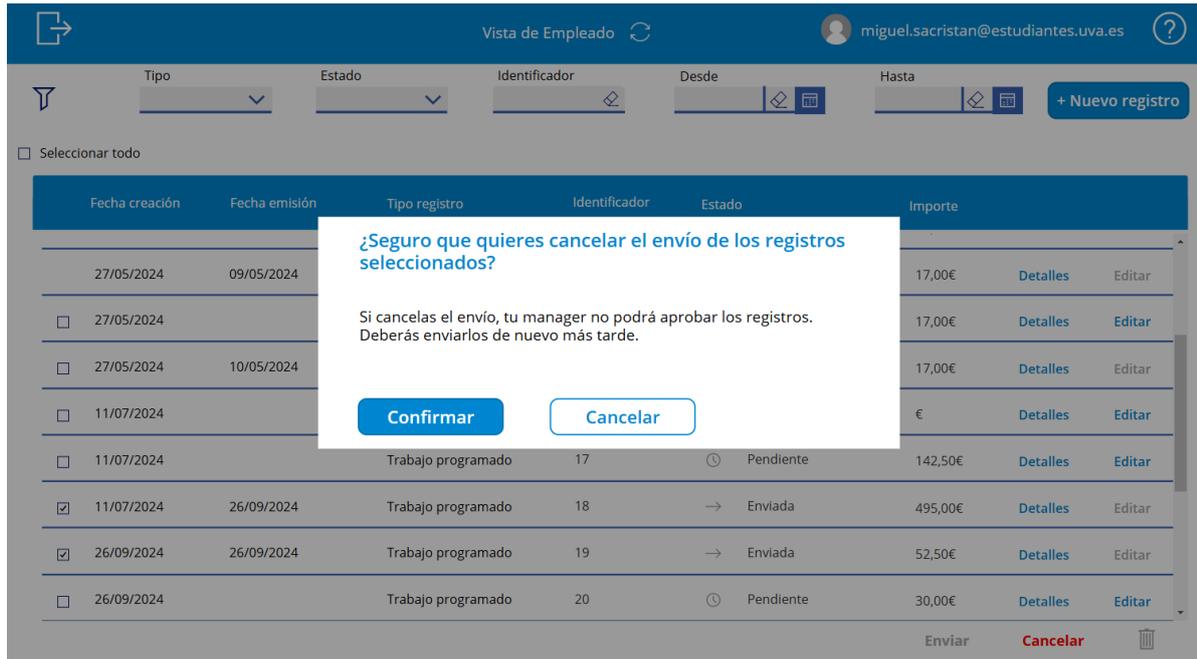


Figura A.31: Confirmar cancelacion

A.5.6. Borrado de un registro

Para borrar registros, el Empleado puede seleccionar una serie de ellos que estén en el estado *Pendiente* y hacer click en el icono de la "Papelera" que aparece en la parte inferior de la pantalla que aparece tras seleccionarlos. Una vez que hemos confirmado la operación, los registros son borrados. Para las guardias que tengan intervenciones, estas se deben borrar en el formulario de edición A.23 antes de ejecutar el borrado.

De la misma manera que en los casos anteriores, para usar la opción de *Seleccionar todo*, la galería debe estar filtrada por estado.



Figura A.32: Botón Borrar para confirmar

Vista de Empleado ↻ miguel.sacristan@estudiantes.uva.es ?

Tipo Estado Identificador Desde Hasta + Nuevo registro

Seleccionar todo

Fecha creación	Fecha emisión	Tipo registro	Identificador	Estado	Importe		
27/05/2024	27/05/2024				13,00€	Detalles	Editar
27/05/2024	09/05/2024				17,00€	Detalles	Editar
27/05/2024	11/05/2024				17,00€	Detalles	Editar
27/05/2024	09/05/2024				17,00€	Detalles	Editar
<input type="checkbox"/> 27/05/2024					17,00€	Detalles	Editar
<input type="checkbox"/> 27/05/2024	10/05/2024	Trabajo programado	5	→ Enviada	17,00€	Detalles	Editar
<input checked="" type="checkbox"/> 11/07/2024		Guardia	12	🕒 Pendiente	€	Detalles	Editar
<input checked="" type="checkbox"/> 11/07/2024		Trabajo programado	17	🕒 Pendiente	142,50€	Detalles	Editar

Enviar Cancelar

¿Seguro que quieres eliminar los registros seleccionados?

Se eliminarán los registros permanentemente y no se podrán recuperar posteriormente.

Confirmar **Cancelar**

Figura A.33: Confirmar borrado

Apéndice B

Código utilizado en la aplicación

Aunque PowerApps es una plataforma de desarrollo low-code, algunas funcionalidades de la aplicación requieren la utilización de código para completar tareas específicas. En este apéndice se presentan algunos de los fragmentos de código más relevantes utilizados en el desarrollo de la aplicación. Estos fragmentos ilustran cómo se implementan ciertas lógicas y procesos dentro de la plataforma.

■ Obtención del precio base de una guardia de la tabla Tarifas

```
1 Set(precioBaseGuardia;  
2 First(  
3     SortByColumns(  
4         Filter(  
5             Tarifas;  
6             Categoria.Value = var_categoria.Value &&  
7             Tipo_incidencia.Value = "Base_guardia"  
8         );  
9         "Fecha";  
10        SortOrder.Descending  
11    )  
12 ).Precio);
```

Listing B.1: Cálculo del precio base de la guardia en PowerApps

■ Registro de una guardia y sus intervenciones en SharePoint desde la aplicación

```
1 Set(  
2     varIncidenciaID;  
3     Patch(  
4         Incidencias;  
5         Defaults(Incidencias);  
6         {  
7             Tipo_incidencia: emp_nuevaIncidencia_tipoIncidencia.  
                SelectedText.Value;
```

```

8      Fecha_creacion: Now();
9      Emisor: var_email;
10     Aprobante: var_aprobante;
11     Comentarios: emp_nuevaIncidencia_comentarios.Text;
12     Estado: "Pendiente";
13     Festivos_guardia: Value(emp_nuevaIncidencia_festivosTrabajados.
14         SelectedText.Value);
15     Inicio_guardia: emp_nuevaIncidencia_inicioGuardia.SelectedDate
16         ;
17     Fin_guardia: DateAdd(emp_nuevaIncidencia_inicioGuardia.
18         SelectedDate; 7; TimeUnit.Days);
19     Horas_guardias: Value(emp_nuevaIncidencia_horasGuardias.Text);
20     Matricula_aprobante: Lookup(Usuarios; Email = var_aprobante).
21         Matricula;
22     Matricula_emisor: Lookup(Usuarios; Email = var_email).Matricula
23         ;
24     Nombre_aprobante: Lookup(Usuarios; Email = var_aprobante).
25         Nombre;
26     Nombre_emisor: Lookup(Usuarios; Email = var_email).Nombre;
27     Importe: precioBaseGuardia + Value(
28         emp_nuevaIncidencia_horasGuardias.Text)*precioHoraGuardia +
29         Value(emp_nuevaIncidencia_festivosTrabajados.SelectedText.
30             Value)*precioFestivo // las remuneraciones correspondientes
31             a cada nivel de empleado se almacenan en la tabla Tarifas
32     }
33     ).ID // Guarda el ID de la nueva incidencia
34 );
35
36 ForAll(
37     col_intervenciones;
38     Patch(
39         Intervenciones;
40         Defaults(Intervenciones);
41         {
42             Fecha: Fecha;
43             Horas: Horas;
44             Comentarios: Comentarios;
45             Guardia: varIncidenciaID // Usa el ID de la incidencia
46                 principal
47         }
48     )
49 );

```

Listing B.2: Registro de una guardia y sus intervenciones en SharePoint desde la aplicación

■ Obtención de las 5 respuestas más usadas por el aprobador

```
1 FirstN(  
2     SortByColumns(  
3         AddColumns(  
4             GroupBy(  
5                 Filter(  
6                     respuestas_predeterminadas;  
7                     email_aprobante = var_email && Not(IsBlank(  
8                         comentario_aprobacion))  
9                 );  
10                comentario_aprobacion;  
11                ComentarioUnico  
12            );  
13            Frecuencia;  
14            CountRows(ComentarioUnico)  
15        );  
16        "Frecuencia";  
17        SortOrder.Descending  
18    );  
19    5  
20 )
```

Listing B.3: Cálculo de las 5 respuestas predeterminadas más frecuentes en PowerApps