

eResourceKeeper

Aplicación para la planificación de recursos humanos y proyectos

Laura Montañana González

Grado en Ingeniería Informática
Ingeniería del Software

Xavier Escudero Sabadell
Santi Caballe Llobet

Enero de 2023



Universitat
Oberta
de Catalunya



Esta obra está sujeta a una licencia de R
reconocimiento-NoComercial-SinObraDerivada
[3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

Título del trabajo	eResourceKeeper. Aplicación para la planificación de recursos humanos y proyectos
Nombre del autor	Laura Montañana González
Nombre del consultor	Xavier Escudero Sabadell
Fecha de entrega	01/2023
Área del Trabajo Final	Ingeniería del Software
Titulación	Grado en Ingeniería Informática
Palabras clave	Aplicación web, gestión de recursos humanos, especificación de requisitos, análisis y diseño orientado a objetos, base de datos relacional, arquitectura en capas.
Resumen del trabajo	
<p>Este proyecto propone el desarrollo de eResourceKeeper, una aplicación para gestionar los recursos humanos y proyectos de una empresa, usando como punto de partida el caso concreto de una empresa que actualmente utiliza una hoja de cálculo para desempeñar esta tarea. Se analiza en detalle la hoja de cálculo, con el fin de incorporar en el nuevo software todas sus funcionalidades y superar sus limitaciones. También se lleva a cabo un estudio comparativo de varios productos comerciales, en aras a detectar funcionalidades nuevas que puedan ser de interés. Como resultado de todo ello se obtiene una serie de requisitos, de entre los cuales se realiza una selección priorizándolos en relación con criterios de coste/valor. Con los requisitos seleccionados se confeccionan casos de uso, a partir de los cuales se elaboran <i>wireframes</i>, con lo que se obtiene una descripción detallada de la interfaz de usuario. A continuación, se diseña el mapa navegacional haciendo uso de un diagrama de estados. Luego se presenta el correspondiente diagrama de clases UML, y se enumeran las reglas de integridad que deben cumplirse. También se muestra el diseño lógico y conceptual de la base de datos, así como su esquema relacional. Por último, se propone utilizar una arquitectura en capas, y se sugiere un conjunto de productos y tecnologías a utilizar con vistas a una posible implementación futura.</p>	
Abstract	
<p>This project proposes the development of eResourceKeeper, a software to manage the human resources and projects of a company, using as a starting point the specific case of a company that currently uses a spreadsheet to perform this task. The spreadsheet is analyzed in detail, in order to incorporate all its functionalities into the new software and overcome its limitations. A comparative study of various commercial products is also carried out, in order to detect new functionalities that may be of interest. As a result of all this, a series of requirements is obtained, from among which a selection is made, prioritizing them based on cost/value criteria. Once requirements have been selected use cases are made, from which wireframes are presented, to provide a detailed description of the user interface. Next, the navigational map is designed using a state diagram. Then the relevant UML class diagram is presented, and the integrity rules that must be followed are listed. The logical and conceptual design of the database used is also shown, as well as its relational diagram. Finally, a layered architecture is proposed, and a set of products and technologies are suggested to be used in case of a possible future implementation.</p>	

Índice

1. Introducción	8
1.1 Contexto y justificación del trabajo	8
1.2 Objetivos de eResourceKeeper	9
1.3 Objetivos del trabajo	9
1.3.1 Objetivo general	9
1.3.2 Objetivos específicos	9
1.4 Enfoque y metodología a seguir	9
1.5 Planificación del trabajo	10
1.6 Breve resumen de los productos obtenidos	14
1.7 Breve descripción de los otros capítulos de la memòria	14
2. Requisitos	15
2.1 Obtención de requisitos	15
2.1.1 Identificación de los stakeholders	15
2.1.2 Identificación de los actores/roles	16
2.1.3 Modelado de perfiles de usuario	17
2.1.4 Análisis de mercado	18
2.1.4.1 Productos a analizar	18
2.1.4.2 Aspectos a analizar	18
2.1.4.3 Análisis de productos	19
2.1.4.4 Análisis de resultados	19
2.1.5 Análisis de la solución actual	20
2.1.5.1 Funcionalidades de la solución actual	20
2.1.5.2 Limitaciones de la solución actual	24
2.1.6 Relación entre usuario y empleado	25
2.1.7 Requisitos candidatos	26
2.1.7.1 Requisitos no funcionales candidatos	26
2.1.7.2 Requisitos funcionales candidatos	27
2.2 Gestión de requisitos	29
2.2.1 Estimación del coste	29
2.2.2 Estimación del valor	30
2.2.3 Estimación de los requisitos funcionales candidatos	31
2.2.4. Priorización de los requisitos	34
2.2.5 Selección de requisitos	34
2.3 Documentación de requisitos	36
2.3.1 Requisitos no funcionales	36
2.3.2 Requisitos funcionales	38
3. Análisis y diseño	51
3.1 Diagrama de casos de uso	51
3.2 Modelización de la interfaz con wireframes	52
3.3 Mapa navegacional	58
3.4 Análisis de solicitud de días de permiso	60
3.5 Análisis de creación de una tarea	61
3.6 Modelo conceptual de clases	63
3.6.1 Reglas de integridad	63

3.6.2 Información derivada	65
3.7 Diseño de la base de datos	65
3.7.1 Diseño conceptual	65
3.7.2 Diseño lógico	65
3.8 Arquitectura	70
3.9 Tecnología	74
3.10 Diseño de la API REST	77
4. Conclusiones	79
5. Glosario	82
6. Referencias	84
Anexo A - Perfiles de usuario	86
Anexo B - Análisis de productos existentes	91
Anexo C - Wireframes	98
Anexo D - Interfaces de la capa de negocio	109
Anexo E - Documentación API REST	111

Índice de figuras

Figura 1.1. Diagrama de Gantt que muestra la planificación temporal del trabajo	13
Figura 2.1. Planificación de proyectos de la hoja de cálculo de Pivotree	21
Figura 2.2. Detalle de las tareas de los empleados	22
Figura 2.3. Gráfica que desglosa las horas por semana	22
Figura 2.4. Tabla que desglosa las horas por semana	23
Figura 2.5. Análisis de KPIs	24
Figura 2.6. Filtros de estadísticas	24
Figura 2.7. Ejemplo de empleado que sobrepasa su jornada laboral	25
Figura 3.1. Diagrama de casos de uso	51
Figura 3.2. Menú de navegación para usuarios con rol User	54
Figura 3.3. Menú de navegación para usuarios con rol Viewer, Editor y Admin	54
Figura 3.4. Planificación de los proyectos	55
Figura 3.5. Planificación de los proyectos con la información de un proyecto expandida	55
Figura 3.6. Planificación de los proyectos guardados como favoritos	56
Figura 3.7. Filtros de búsqueda de la planificación de los proyectos	56
Figura 3.8. Planificación de las tareas de todos los usuarios	56
Figura 3.8. Planificación de las tareas de todos los usuarios	57
Figura 3.10. Filtros de búsqueda de la planificación de las tareas	57
Figura 3.11. Crear una tarea	58
Figura 3.12. Modificar una tarea	58
Figura 3.13. Mapa navegacional de las diferentes pantallas para los usuarios User	59
Figura 3.14. Mapa navegacional de las diferentes pantallas para usuarios privilegiados	59
Figura 3.15. Diagrama de máquina de estados de una solicitud de días de permiso	60
Figura 3.16. Diagrama de actividades de una solicitud de días de permiso	61
Figura 3.17. Diagrama de actividades con los pasos para crear una tarea	62
Figura 3.18. Diagrama de clases de eResourceKeeper	63
Figura 3.19. Diagrama del esquema relacional de eResourceKeeper	69
Figura 3.20. Diagrama de la aplicación web	70
Figura 3.21. Diagrama de arquitectura de la aplicación por capas	71
Figura 3.22. Diagrama de componentes desde el punto de vista computacional	73
Figura 3.23. Diagrama de las interfaces de la capa de negocio	73
Figura 3.24. Alternativas tecnológicas evaluadas	74
Figura 3.25. Diagrama de arquitectura de la aplicación con el stack tecnológico escogido	75
Figura 3.26. Diagrama de secuencia del flujo de autenticación y autorización	76
Figura A.1. Personaje arquetípico Persona 1	86
Figura A.2. Personaje arquetípico Persona 2	87
Figura A.3. Personaje arquetípico Persona 3	88
Figura A.4. Personaje arquetípico Persona 4	89
Figura A.4. Personaje arquetípico Persona 4	90
Figura C.1. Usuarios con horas libres	98
Figura C.2. Planificación de los proyectos del usuario	98
Figura C.3. Planificación de las tareas del usuario	98
Figura C.4. Permisos solicitados por el usuario	99
Figura C.5. Formulario de solicitud de permiso	99

Figura C.6. Permisos pendientes de gestionar	99
Figura C.7. Listado de proyectos	100
Figura C.8. Detalle de un proyecto de cliente	100
Figura C.8. Detalle de un proyecto de cliente	100
Figura C.10. Modificar un proyecto de cliente	101
Figura C.11. Modificar un proyecto interno	101
Figura C.12. Crear un proyecto	101
Figura C.13. Eliminar un proyecto	102
Figura C.14. Búsqueda de proyecto por nombre con resultados	102
Figura C.15. Búsqueda de proyecto por nombre sin resultados	102
Figura C.16. Listado de usuarios	103
Figura C.17. Detalle de un usuario de tipo Employee	103
Figura C.18. Detalle de un usuario de tipo Contractor	104
Figura C.19. Modificar un usuario de tipo Employee	104
Figura C.20. Modificar un usuario de tipo Contractor	105
Figura C.21. Listado de clientes	105
Figura C.22. Detalle de un cliente	105
Figura C.23. Modificar un cliente	106
Figura C.24. Eliminar un cliente	106
Figura C.25. Listado de localizaciones	107
Figura C.26. Detalle de una localización	107
Figura C.27. Modificar una localización	108

Índice de tablas

Tabla 1.1. Planificación del trabajo	11
Tabla 2.1. Análisis de resultados del estudio de los productos de la competencia	19
Tabla 2.2. Requisitos no funcionales candidatos	26
Tabla 2.3. Requisitos funcionales candidatos	27
Tabla 2.4. Estimación de los requisitos funcionales candidatos	31
Tabla 2.5. Requisitos funcionales seleccionados	34
Tabla 2.6. Coste de los requisitos agrupados por valor	36
Tabla 2.7. Casos de uso	39
Tabla 3.1 Correspondencia de casos de uso con wireframes	53
Tabla 3.2 Correspondencia de capas de la aplicación y su función	78
Tabla B.1. Análisis de productos de la competencia	98

1. Introducción

1.1 Contexto y justificación del trabajo

En términos generales, la planificación de recursos humanos consiste en prever las necesidades de personal de una organización y fijar los pasos necesarios para cubrirlas. Se trata de una actividad de vital importancia para el funcionamiento eficaz de cualquier empresa, pues su objetivo es utilizar los recursos humanos con la mayor eficacia posible, a fin de alcanzar las metas de la organización. Una mala planificación de los recursos humanos puede suponer que la empresa no pueda hacer frente al trabajo por falta de personal, o que tenga recursos humanos sin aprovechar.

En las empresas que se dedican a los servicios profesionales¹, esta actividad está directamente relacionada con la gestión de proyectos. En este contexto, la planificación de recursos humanos consiste en determinar qué perfil profesional se necesita para llevar a cabo un proyecto, localizar la persona idónea para ese puesto, dentro o fuera de la empresa, y asignarla al proyecto durante el tiempo necesario.

Pivotree, empresa en la que trabajo desde hace cinco años, es una agencia de *ecommerce* que, para gestionar la planificación de recursos humanos en los proyectos, utiliza actualmente como herramienta una hoja de cálculo.

El uso de una hoja de cálculo presenta los siguientes problemas:

- No hay una forma fácil de controlar los permisos de acceso para cada empleado.
- Para ofrecer diferentes vistas de la información se utilizan diferentes pestañas, replicando los datos manualmente de unas a otras, por lo que el proceso es ineficaz y propenso a fallos.
- La hoja de cálculo presenta limitaciones a la hora de mostrar los datos.

Este trabajo pretende aportar una solución a todos esos problemas. En él se realizará el análisis y diseño de una nueva aplicación, que denominaremos eResourceKeeper, que servirá para planificar la asignación de recursos humanos a los proyectos de una empresa, teniendo en cuenta las necesidades concretas que tiene Pivotree actualmente. Este trabajo servirá como punto de partida para una futura implementación, tarea que queda fuera del alcance de este proyecto.

A nivel personal, mi motivación para realizar este trabajo es que he identificado una necesidad en la empresa para la que trabajo, y una oportunidad de mejora en la planificación de recursos humanos y proyectos. Además, creo que es un caso real adecuado para poner en práctica los conocimientos de ingeniería del software adquiridos en el Grado de Ingeniería Informática.

¹Cualquier empresa o departamento cuyo producto principal es un servicio que requiere personal cualificado en lugar de un producto manufacturado. Por ejemplo, servicios jurídicos, de contabilidad, publicidad y marketing, de IT, etc.

1.2 Objetivos de eResourceKeeper

El objetivo principal de eResourceKeeper es ofrecer una aplicación que sirva para planificar la asignación de recursos humanos a los proyectos en la empresa Pivotree.

eResourceKeeper debería ser una alternativa mejor que la hoja de cálculo utilizada actualmente. Para ello será necesario que provea, al menos, las mismas funcionalidades, elimine las limitaciones existentes y proporcione una mayor eficiencia.

Aunque no será hasta la fase siguiente del proyecto cuando se definirán los requisitos, las funcionalidades principales que ofrecerá eResourceKeeper son las siguientes:

- Planificación temporal de proyectos, con el fin de tener una visión general clara de la línea de tiempo de cada proyecto, las tareas y los empleados que se harán cargo de ellas.
- Gestión de la jornada laboral de los empleados, teniendo en cuenta la duración de la jornada laboral de cada trabajador, los días festivos, las vacaciones y permisos.
- Consulta de estadísticas para monitorizar la ocupación de los empleados, la capacidad prevista de los equipos y qué parte del trabajo planificado es facturable.

1.3 Objetivos del trabajo

1.3.1 Objetivo general

Realizar la especificación de requisitos, el análisis y el diseño de una aplicación que sirva para planificar la asignación de recursos humanos a los proyectos de una empresa.

1.3.2 Objetivos específicos

- Realizar un análisis de mercado de las aplicaciones de planificación de recursos humanos.
- Identificar las necesidades que tiene la empresa Pivotree en la planificación de recursos humanos.
- Definir un producto mínimo viable.
- Especificar los requisitos.
- Documentar mediante modelos los requisitos establecidos.
- Realizar el diseño técnico de la aplicación.
- Aplicar los conocimientos adquiridos en diferentes asignaturas del Grado de Ingeniería Informática.

1.4 Enfoque y metodología a seguir

Una posible solución al problema que tiene la empresa Pivotree sería utilizar un producto ya existente. Sin embargo, la funcionalidad provista por la hoja de cálculo utilizada actualmente ha ido creciendo a lo largo de los años, y se ha hecho tan a medida de la empresa, que parece poco probable que exista una herramienta que proporcione exactamente las mismas funcionalidades. Por otro lado, la empresa cuenta con personal cualificado capaz de abordar un desarrollo de este tipo. Por tanto, la opción escogida es desarrollar una

aplicación completamente nueva a medida, eResourceKeeper, para la planificación de recursos humanos, que reemplace y mejore la hoja de cálculo actualmente en uso.

Aunque el objetivo es diseñar un producto nuevo, se realizará un estudio de mercado de las soluciones existentes para tomarlas como punto de referencia a la hora de definir los requisitos de eResourceKeeper.

Según la clasificación de Wysocki [2014] nuestro proyecto encaja en lo que él denomina grupo 1, ya que se tiene clara la necesidad a cubrir y se conocen los detalles de cómo será la solución. Utilizaremos la metodología de desarrollo del ciclo de vida clásico o desarrollo en cascada [Royce, 1970], que se considera ideal para este tipo de proyectos.

El modelo en cascada define una serie de etapas que deben ejecutarse de forma consecutiva. Este proyecto se centrará exclusivamente en las etapas de requisitos y de análisis y diseño. El resto queda fuera del ámbito de este proyecto.

Aunque el modelo en cascada en teoría es estrictamente secuencial, existirá siempre cierta retroalimentación a las fases previas, con el fin de corregir errores detectados *a posteriori*. Además, se usarán algunas técnicas que tienen su origen en metodologías de desarrollo ágil, como por ejemplo, la estimación de requisitos utilizando *story points* o la técnica de priorización MoSCoW. En cada caso se ha optado por utilizar la técnica que se ha considerado más apropiada, sin tener en cuenta la metodología en la que se originó.

1.5 Planificación del trabajo

Para la realización de este trabajo se utilizarán las siguientes herramientas:

- [ProjectLibre](#), para la realización del diagrama de Gantt y la planificación temporal del proyecto.
- [Trello](#), para la organización y gestión de tareas.
- [Google Docs](#) como editor de texto.
- [Adobe XD](#), para la realización de los *wireframes* y los perfiles de usuario.
- [draw.io](#) y [Lucidchart](#), para la realización de los diagramas UML (diagramas de clases, de actividades, de secuencia...).

El proyecto está dividido en dos bloques principales que se corresponden con cada una de las fases del modelo en cascada:

- **Requisitos.** En esta fase se define qué funcionalidades debe poseer el producto a desarrollar. Las actividades que componen este trabajo son la obtención, la gestión y la documentación de requisitos.
- **Análisis y diseño.** En esta fase se define cómo va a ser el producto desde el punto de vista externo e interno para alcanzar los requisitos establecidos en la fase anterior.

La planificación viene determinada por el calendario de actividades evaluables de la asignatura del TFG. En la planificación propuesta se ha desplazado la fecha de entrega de la PAC2 para poder hacerla coincidir con el fin de la fase de requisitos.

La planificación tiene 5 hitos principales, que se corresponden con las actividades evaluables de la asignatura:

1. PAC1 Plan de trabajo
2. PAC2 Requerimientos
3. PAC3 Análisis y Diseño
4. Memoria y presentación
5. Defensa

El siguiente cronograma, que abarca desde el inicio del proyecto hasta el día de la defensa, muestra las diferentes etapas y sus tareas. El diagrama de Gantt que hay a continuación muestra, además, la dependencia entre diferentes tareas.

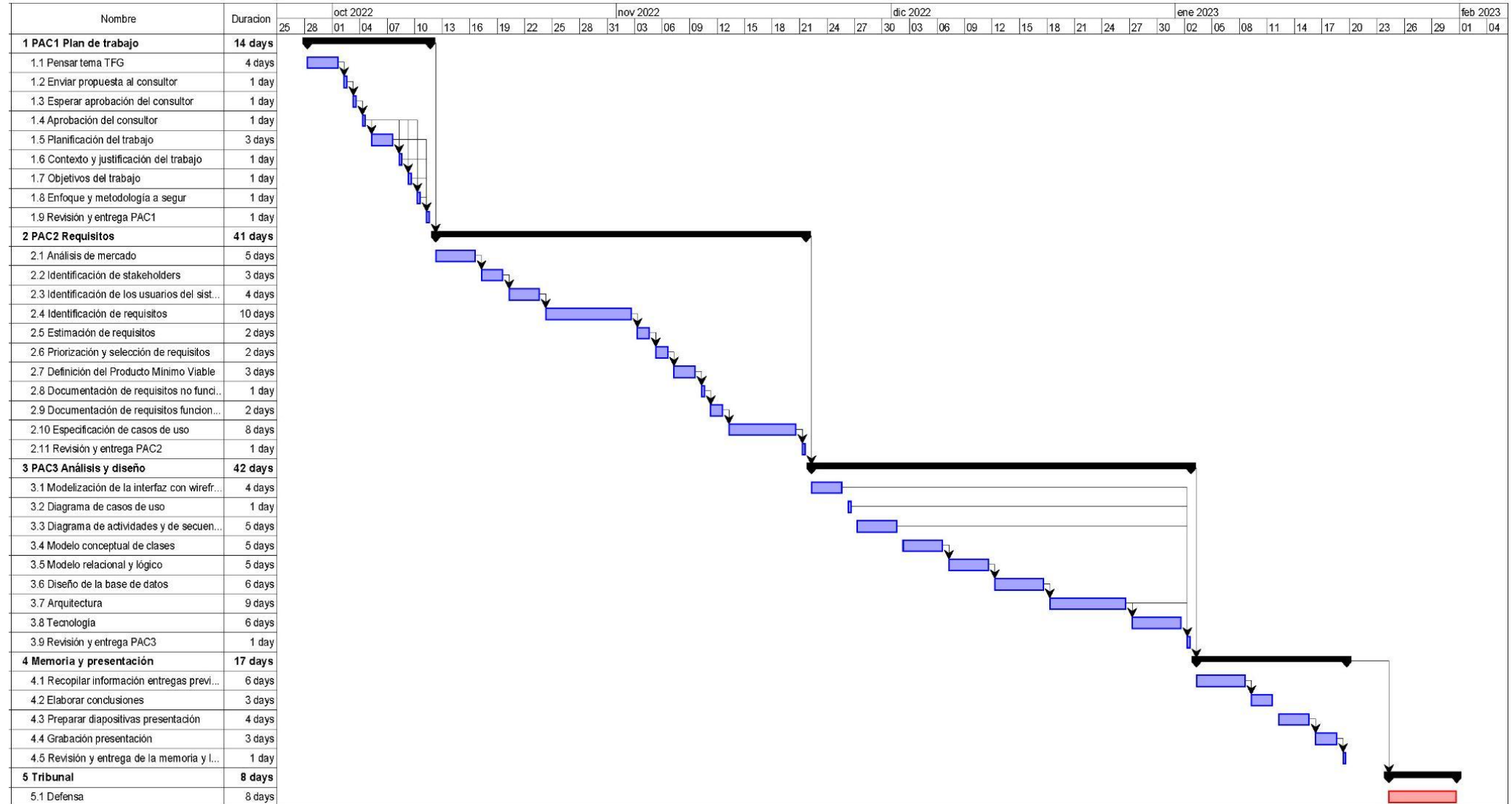
Tabla 1.1

Planificación del trabajo

ID	Actividad	Fecha inicio	Fecha fin	Duración
1	PAC1 Plan de trabajo	28 septiembre 2022	11 octubre 2022	14 días
1.1	Pensar tema TFG	28 septiembre 2022	1 octubre 2022	4 días
1.2	Enviar propuesta al consultor	2 octubre 2022	2 octubre 2022	1 día
1.3	Esperar aprobación del consultor	3 octubre 2022	3 octubre 2022	1 día
1.4	Aprobación del consultor	4 octubre 2022	4 octubre 2022	1 día
1.5	Planificación del trabajo	5 octubre 2022	7 de octubre	3 días
1.6	Contexto y justificación del trabajo	8 octubre 2022	8 octubre 2022	1 día
1.7	Objetivos del trabajo	9 octubre 2022	9 octubre 2022	1 día
1.8	Enfoque y metodología a seguir	10 octubre 2022	10 octubre 2022	1 día
1.9	Revisión y entrega PAC1	11 octubre 2022	11 octubre 2022	1 día
2	PAC2 Requisitos	12 octubre 2022	21 noviembre 2022	41 días
2.1	Análisis de mercado	12 octubre 2022	16 octubre 2022	5 días
2.2	Identificación de los <i>stakeholders</i>	17 octubre 2022	19 octubre 2022	3 días
2.3	Identificación de los usuarios	20 octubre 2022	23 octubre 2022	4 días
2.4	Identificación de requisitos	24 octubre 2022	2 noviembre 2022	10 días
2.5	Estimación de requisitos	3 noviembre 2022	4 noviembre 2022	2 días
2.6	Priorización y selección de requisitos	5 noviembre 2022	6 noviembre 2022	2 días
2.7	Definición del Producto Mínimo Viable	7 noviembre 2022	9 noviembre 2022	3 días
2.8	Documentación de requisitos no funcionales	10 noviembre 2022	10 noviembre 2022	1 día
2.9	Documentación de requisitos funcionales	11 noviembre 2022	12 noviembre 2022	2 días
2.10	Especificación de casos de uso	13 noviembre 2022	20 noviembre 2022	8 días
2.11	Revisión y entrega PAC2	21 noviembre 2022	21 noviembre 2022	1 día
3	PAC3 Análisis y diseño	22 noviembre 2022	2 enero 2023	42 días
3.1	Diagrama de casos de uso	22 noviembre 2022	22 noviembre 2022	1 día
3.2	Modelización de la interfaz con <i>wireframes</i>	23 noviembre 2022	26 noviembre 2022	4 días
3.3	Diagrama de actividades y de secuencia	27 noviembre 2022	1 diciembre 2022	5 días

3.4	Modelo conceptual de clases	2 diciembre 2022	6 diciembre 2022	5 días
3.5	Modelo relacional y lógico	7 diciembre 2022	11 diciembre 2022	5 días
3.6	Diseño de la base de datos	12 diciembre 2022	17 diciembre 2022	6 días
3.7	Arquitectura	18 diciembre 2022	26 diciembre 2022	9 días
3.8	Tecnología	27 diciembre 2022	1 enero 2023	6 días
3.9	Revisión y entrega PAC3	2 enero 2023	2 enero 2023	1 día
4	Memoria y presentación	3 enero 2023	19 enero 2023	17 días
4.1	Recopilar información entregas previas y aplicar correcciones	3 enero 2023	8 enero 2023	6 días
4.2	Elaborar conclusiones	9 enero 2023	11 enero 2023	3 días
4.3	Preparar diapositivas presentación	12 enero 2023	15 enero 2023	4 días
4.4	Grabación presentación	16 enero 2023	18 enero 2023	3 días
4.5	Revisión y entrega de la memoria y la presentación	19 enero 2023	19 enero 2023	1 día
5	Tribunal	24 enero 2023	31 enero 2023	8 días
5.1	Defensa	24 enero 2023	31 enero 2023	8 días

Figura 1.1.
Diagrama de Gantt que muestra la planificación temporal del trabajo



1.6 Breve resumen de los productos obtenidos

Al finalizar este trabajo se obtendrán los siguientes productos:

- La memoria del trabajo con la especificación de requisitos, el análisis y el diseño del producto y las conclusiones.
- La presentación sobre el trabajo realizado.
- Un vídeo explicativo sobre el trabajo realizado.

1.7 Breve descripción de los otros capítulos de la memoria

En los siguientes capítulos de la memoria se verán los siguientes contenidos:

- **Requisitos.** Este capítulo se centra en definir los requisitos de eResourceKeeper. En primer lugar, se llevará a cabo la obtención de los requisitos candidatos mediante diferentes técnicas. Una vez obtenidos, se seleccionarán aquellos que van a formar parte del producto, estimando el coste de cada requisito y el valor que aporta. Por último, se documentarán los requisitos del proyecto.
- **Análisis y diseño.** Este capítulo tiene como objetivo realizar el análisis y diseño de eResourceKeeper. El análisis nos permitirá especificar en detalle los requisitos de eResourceKeeper al documentar mediante diferentes modelos qué hace el sistema, mientras que el diseño nos permitirá definir cómo debe implementarse (qué componentes forman parte de él, cómo se relacionan entre ellos, etc.).

2. Requisitos

2.1 Obtención de requisitos

En esta primera etapa del proyecto se obtendrán los requisitos candidatos de la aplicación, que serán todos aquellos que, idealmente, debería satisfacer eResourceKeeper. En etapas posteriores se decidirá cuales se incorporan.

La obtención de los requisitos pasa por diferentes fases, que se describen a continuación.

2.1.1 Identificación de los *stakeholders*

Una de las primeras tareas en la obtención de requisitos consiste en identificar a los *stakeholders*, con el fin de determinar cuáles son sus necesidades y restricciones.

Aunque el objetivo de eResourcekeeper es planificar la asignación de recursos humanos de cualquier empresa, el proyecto se realizará teniendo en cuenta específicamente los requisitos que tiene la empresa Pivotree. Por tanto, la identificación de *stakeholders* se hará para este ámbito en concreto.

Todos los usuarios de la aplicación son *stakeholders*, puesto que todos la usan y, por lo tanto, tienen interés en ella. Sin embargo, no todos los *stakeholders* serán usuarios, dado que puede haber entidades que, aunque no utilicen la aplicación, se vean afectadas por ella y que, por lo tanto, también sean parte interesada.

A continuación se enumeran los principales stakeholders del proyecto.

- **Capacity manager.** Es la persona actualmente encargada de mantener la hoja de cálculo. El nuevo producto afectará a su forma de trabajar. Será el usuario que se encargue de gestionar la aplicación y mantener los datos actualizados.
- **Project manager, Delivery manager y Resource manager.** Son los principales usuarios de la hoja de cálculo actual. Necesitan acceder a la información para conocer la planificación prevista y poder anticiparse a futuras necesidades. La nueva aplicación afectará a su forma de trabajar.
- **Equipo directivo.** No será usuario de eResourceKeeper pero, en última instancia, es quien va a financiar el proyecto y, por tanto, tiene un impacto en él. Además, tiene interés en el desarrollo de la aplicación pues ésta supondrá una mayor eficiencia en un proceso fundamental para la empresa.
- **Patrocinador del proyecto.** No será usuario de la aplicación pero es la persona que ha promovido la iniciativa en la empresa y por tanto, tiene un interés en que el proyecto llegue a buen puerto y que sea fácil de vender al equipo directivo para que lo financie.
- **Departamento de RRHH.** En principio, no será usuario de la aplicación pero tiene interés en ella con el fin de mejorar la gestión de recursos humanos en la empresa.

- **Empleados.** Actualmente los trabajadores no tienen acceso a la hoja de cálculo y por tanto, no son usuarios. Esto se debe a que, en la hoja de cálculo no es posible controlar los permisos de acceso y hay cierta información que la empresa no desea que vean. Dado que en eResourceKeeper sí que será posible establecer diferentes permisos de acceso, los empleados pasarán a ser usuarios de la aplicación y por tanto, *stakeholders*.
- **Administradores de sistemas.** Serán los encargados de administrar el servidor donde se aloje la nueva aplicación y pueden tener algunos requisitos o restricciones que sea necesario tener en cuenta.

Una vez identificados los *stakeholders* del proyecto, el siguiente paso es identificar los requisitos de cada uno de ellos. Una de las técnicas más utilizadas es la realización de entrevistas y cuestionarios. En nuestro caso, esta técnica no se ha utilizado debido al carácter académico del proyecto.

2.1.2 Identificación de los actores/roles

Según la especificación de UML “Un Actor especifica un rol desempeñado por un usuario o cualquier otro sistema que interactúa con el sujeto” (OMG, 2017: 647).

Así pues, podemos a todos los efectos prácticos considerar rol y actor como sinónimos. Generalmente actor se utiliza preferentemente en la de obtención de requisitos y elaboración de los casos de uso, mientras que rol se emplea sobre todo en las fases de diseño e implementación. Nosotros seguiremos esa convención.

La identificación de los actores es uno de los primeros pasos a realizar para delimitar la aplicación y definir su alcance. Además, su correcta identificación es crucial de cara a aplicar algunas técnicas para la obtención de requisitos.

Es importante establecer la diferencia entre *stakeholder* y actor. Mientras que los *stakeholders* participan en la aplicación mediante sus intereses, los actores participan en ella con su comportamiento.

También es importante establecer la diferencia entre usuario y actor. Cuando un usuario interactúa con la aplicación lo hace asumiendo en cada momento un papel determinado, que es lo que denominamos actor. Diferentes usuarios pueden acceder como un mismo actor. Por otro lado, en algunos casos un mismo usuario puede acceder a la aplicación, en diferentes momentos, como distintos actores.

En eResourceKeeper vamos a establecer los siguientes actores:

- **User.** Tiene permisos de lectura únicamente sobre las tareas que tiene asignadas y los proyectos en los que participa, salvo los proyectos tentativos, a los que no tendrá acceso. También puede utilizar la aplicación para solicitar días de permiso.

- **Viewer.** Tiene los mismos permisos que un *User*. Además, tiene permisos de lectura sobre toda la aplicación. No tiene permisos de escritura y, por tanto, no puede hacer ningún cambio en las tareas de los usuarios, ni en la planificación de proyectos. Puede aprobar o denegar los días de permiso solicitados por los usuarios que tiene a su cargo.
- **Editor.** Puede hacer lo mismo que un *Viewer* y, además, tiene permisos de escritura sobre toda la aplicación, excepto sobre las entidades auxiliares (clientes, puesto, localizaciones, tipo de empleado, etc), los usuarios de la aplicación o los ajustes generales.
- **Admin.** Tiene acceso completo a todas las funcionalidades. Puede hacer lo mismo que un *Editor* y, además, tiene permisos de escritura sobre las entidades auxiliares, los usuarios y los ajustes generales.

2.1.3 Modelado de perfiles de usuario

El modelado de perfiles de usuario es una técnica que consiste en identificar tipos de personajes para cada actor que interactúa con la aplicación y definir los requisitos que cada uno tendrá. Es decir, en lugar de buscar los requisitos de los usuarios a nivel individual, se agrupan según características comunes. Esta técnica se basa en la premisa de que todos los usuarios con un mismo perfil tendrán requisitos parecidos.

Además, definir perfiles de usuario ayuda a que todo el equipo tenga presente para quién se está diseñando el producto y puede ser útil en determinados momentos para tomar decisiones y justificarlas para un perfil determinado.

Se han elaborado cinco perfiles de usuario correspondientes a sujetos que encajan en los cuatro actores mencionados, utilizando el concepto de 'usuario persona' propuesto por Cooper [2004]. Esta técnica utiliza personajes arquetípicos y permite aportar información concreta sobre las características de los diferentes tipos de usuario: su nombre, su aspecto, sus necesidades, motivaciones, habilidades, limitaciones, conocimientos previos, etc.

En el [anexo A](#) se muestran cinco personajes que encajan con los diferentes perfiles que contempla la aplicación. Para cada uno de ellos se muestran los datos de identificación, datos demográficos, necesidades y objetivos del nuevo producto y los *pain points*, es decir, los problemas con los que se enfrentan actualmente y que la aplicación debería solucionar. Además, se muestra la tolerancia al cambio de cada individuo, un punto importante a tener en cuenta dado que se trata de migrar de la solución existente a una nueva. También se especifica el uso que cada uno hace de las funcionalidades clave de la aplicación.

Las fotografías utilizadas para los diferentes personajes son imágenes generadas mediante inteligencia artificial, combinando fotos reales de personas en la web <https://thispersondoesnotexist.com/>.

2.1.4 Análisis de mercado

Una fuente de información muy útil a la hora de obtener los requisitos para el diseño de una aplicación es analizar las soluciones de la competencia. Esta técnica puede ser de utilidad para conocer:

- Las funcionalidades existentes en el mercado, que pueden ayudar a descubrir requisitos no sospechados. Hay necesidades que cuesta identificar de primeras pero si se tienen delante resultan mucho más fáciles de reconocer.
- Las expectativas que pueden tener los usuarios
- Buenas y malas prácticas
- Estándares a nivel de diseño de interfaz

En este apartado se llevará a cabo un análisis de diferentes aplicaciones de gestión de recursos humanos existentes en el mercado.

2.1.4.1 Productos a analizar

Para escoger los productos a analizar, de entre todos los existentes en el mercado, se ha llevado a cabo una primera investigación de las soluciones disponibles. Para ello, se han consultado varios artículos donde se comparan diferentes productos y sus funcionalidades principales. Tras dicho análisis preliminar, se han seleccionado los siguientes productos:

- **Float** <https://www.float.com/>
- **Runn** <https://www.runn.io/>
- **Teamdeck** <https://teamdeck.io/pricing/>
- **ResourceGuru** <https://resourceguruapp.com/>

Llegados a este punto merece la pena comentar que los productos existentes en el mercado, tanto los que hemos elegido para analizar como otros, ofrecen sus servicios en la nube y, por tanto, están diseñados para un entorno multiempresa, es decir, una sola instancia de la aplicación, ejecutándose en un servidor, permite dar servicio a múltiples empresas. Aunque eResourceKeeper aspira a ofrecer el servicio a una sola empresa, ya que ese es el ámbito de utilización de la hoja de cálculo actual, diseñaremos la aplicación para que en un futuro pueda usarse en un entorno multiempresa.

Contemplar un entorno multiempresa implicaría requisitos adicionales como, por ejemplo, añadir un actor *SuperAdmin* para controlar el sistema en su conjunto o establecer un proceso de registro de empresas. Estos requisitos añadirían complejidad al desarrollo, aumentarían su coste y no estarían justificados a la vista de los objetivos actuales del proyecto. Por ello, para esta primera versión de la aplicación prescindiremos del actor *SuperAdmin*. La base de datos de momento solo tendrá una empresa, que será Pivotree.

2.1.4.2 Aspectos a analizar

Se realizará una exploración general de cada aplicación. Además, se examinarán en detalle los siguientes aspectos:

1. **Creación de una cuenta de usuario.** Cuál es el proceso de creación de una cuenta.
2. **Búsqueda.** Qué funcionalidades ofrece la herramienta a la hora de realizar búsquedas de diferentes contenidos.
3. **Gestión de proyectos (alta, baja, edición).** Cuál es el proceso de gestión de un proyecto.
4. **Gestión de empleados (alta, baja, edición).** Cuál es el proceso de gestión de un empleado.
5. **Consulta de planificación de proyectos.** Cómo se muestra la información relativa a las tareas asignadas a un empleado, necesidades no cubiertas, planificación de hitos, etc.
6. **Consulta de tareas de empleados.** Cómo se muestran las tareas asignadas a los empleados, y sus vacaciones.
7. **Control de disponibilidad de empleados.** Cómo se registran en la aplicación los días de vacaciones, días de permiso, los festivos y la duración de la jornada laboral.
8. **Estadísticas e informes.** Qué estadísticas ofrece la aplicación, cómo se representan y lo útiles que pueden resultar para el usuario.
9. **Otros.** Aspectos que no entran dentro de ninguna de las categorías anteriores.

2.1.4.3 Análisis de productos

El [anexo B](#) muestra una tabla comparativa de los cuatro productos seleccionados.

2.1.4.4 Análisis de resultados

Una vez finalizado el estudio de los productos seleccionados, es importante analizar aquellos aspectos que se consideran clave y que deberían tenerse en cuenta en el diseño de eResourceKeeper. A continuación se enumeran tanto los aspectos o *insights* que funcionan bien y se quieren incorporar, como aquellos que no funcionan o se consideran negativos y se desean evitar.

Tabla 2.1

Análisis de resultados del estudio de los productos de la competencia

<i>Insights positivos</i>	<i>Insights a evitar</i>
Que los empleados tengan o no acceso a la aplicación según convenga.	Que ningún empleado tenga acceso a la aplicación.
Que se puedan realizar búsquedas mediante filtros, para encontrar rápidamente lo que se busca y obtener listados más acotados.	No distinguir visualmente de forma clara los proyectos y tareas facturables y no facturables.
Que el usuario pueda guardar búsquedas, para poder acceder cómodamente a aquellos contenidos que consulta habitualmente.	Mostrar en la planificación de proyectos solo las tareas, sin mostrar los empleados que se hacen cargo de ellas.
Permitir añadir proyectos y empleados de forma tentativa, para anticipar futuras necesidades.	Que no sea posible establecer la función que un empleado desempeña en un proyecto, cuando esta es diferente al puesto que ocupa en la empresa.
Mostrar en la planificación de proyectos las tareas que tiene el proyecto y el empleado que la tiene asignada.	El abuso de botones únicamente formados por iconos, ya que pueden resultar difíciles de entender para usuarios con dificultades cognitivas.
Permitir añadir y editar tareas arrastrando en el calendario.	Presentar las estadísticas solo mediante tablas.

Guardar proyectos como favoritos para destacar ciertos proyectos.	
Poder añadir fases/hitos a un proyecto	
Poder intercambiar un empleado por otro en la planificación de un proyecto.	
Destacar visualmente cuando un empleado está por encima de su jornada laboral.	
En la vista de planificación de proyectos, mostrar el total de horas planificadas para cada empleado.	
En la vista de las tareas de empleados, mostrar cuántas horas tiene ya asignadas y cuántas quedan pendientes de asignar.	
Poder asignar diferentes tipos de permisos (vacaciones, maternidad, etc) a los empleados.	
Poder configurar la duración de la jornada completa de forma global y no tener que hacerlo por empleado.	
Poder configurar la duración de la jornada parcial de cada empleado, en los casos en que sea necesario.	
Poder contar con diferentes calendarios festivos por ubicaciones, dado que la empresa Pivotree trabaja con equipos distribuidos geográficamente.	
Poder ver todos las ausencias planificadas por permisos o vacaciones, para buscar posibles reemplazos y detectar ausencias solapadas.	
Presentar las estadísticas como una combinación de gráficas y tablas.	
Permitir la exportación/importación de datos.	
Permitir la configuración de campos personalizados.	

2.1.5 Análisis de la solución actual

Dado que se parte de una solución existente (la hoja de cálculo), efectuar un análisis sobre las funcionalidades y limitaciones que a día de hoy presenta dicha solución es quizás la forma más útil de obtener requisitos. En este apartado se llevará a cabo dicho análisis.

2.1.5.1 Funcionalidades de la solución actual

- Dar de **alta** un **proyecto** con la siguiente información: nombre del proyecto, estado (activo/inactivo/tentativo), plataforma (oracle, SAP, VTEX, ...) y cliente. Todos los campos son obligatorios.
- Dar de **baja** un **proyecto**.

- Dar de **alta** un **empleado** con la siguiente información: nombre, estado (activo/inactivo), centro de costes, puesto (desarrollador *back-end*, desarrollador *front-end*, arquitecto, *business analyst*, ...), fecha de inicio, fecha de fin, proveedor (la empresa que proporciona al trabajador *freelance*), localización, manager y duración de la jornada laboral. Todos los campos, excepto la fecha de fin, son obligatorios.
- Los empleados no se dan de baja. Cuando un empleado deja la empresa pasa a estar inactivo.
- Las líneas temporales muestran semanas, es decir, cada celda representa una semana. La forma de indicar la dedicación de un empleado es mediante el número de horas por semana.
- Existe una pestaña que muestra la **planificación de todos los proyectos**. En ella se puede ver una línea temporal con los diferentes empleados que forman el equipo del proyecto. Para cada empleado se muestra la tarea que llevará a cabo en el proyecto, si su trabajo es facturable y si su asignación al proyecto es *hard* o *soft*, es decir, si se puede intercambiar por otro empleado o no. Además, se muestra la asignación del empleado al proyecto a lo largo del tiempo, indicando el número de horas a la semana que el empleado trabaja en el proyecto.

Figura 2.1
Planificación de proyectos de la hoja de cálculo

Client	Project Name	Name	Role	Billable	31/03/22	1/04/22	14/04/22	21/04/22	28/04/22	5/05/22	12/05/22	19/05/22	26/05/22
Locality One	L1 Oracle Commerce Services												
Locality One	L1 Oracle Commerce Services	Melissa King	Commerce BE Dev	Client Billable	40	40	40	40	40	40	40	40	40
Subsidiary Inc.	ULL - PS-ATG Development												
Subsidiary Inc.	ULL - PS-ATG Development	Kunal Desai, Sarah	Commerce BE Dev	Client Billable	40	40	40	40	40	40	40	40	40
Subsidiary Inc.	ULL - PS-ATG Development	Felix Wang	Commerce BE Dev	Client Billable	40	0	0	40	40	40	40	0	0
Subsidiary Inc.	ULL - PS-ATG Development	Cheng Kai	Commerce BE Dev	Client Billable	40	40	40	40	40	40	40	24	16
Subsidiary Inc.	ULL - PS-ATG Development	Aditya, Tom	Commerce PM	Client Non Billable	8	8	8	8	8	8	8	8	8
Subsidiary Inc.	ULL - PS-ATG Development	Caroline Jean	Commerce Architect	Client Billable	40	40	40	40	40	40	40	24	0
Subsidiary Inc.	ULL - PS-ATG Development	Kandarp Navarathnan	Commerce BE Dev	Client Billable	40	40	40	40	40	40	40	40	40
Subsidiary Inc.	ULL - PS-ATG Development	Kunal Arif	Commerce BE Dev	Client Billable	24	40	32	40	40	40	40	40	40
Subsidiary Inc.	ULL - PS-ATG Development	Prashant Kulkarni (C)	Commerce BE Dev	Client Billable	40	40	40	40	40	40	40	40	40

[Ver imagen a tamaño completo](#)

- Existe una pestaña que muestra diferentes líneas temporales de aspectos que se consideran claves para la planificación:
 - **Resumen de las horas asignadas a cada empleado.** Línea temporal con todos los empleados, agrupados por el puesto que ocupan en la empresa, y las horas planificadas que tienen para cada semana. [Ver imagen.](#)
 - **Vacaciones.** Línea temporal con los empleados que tienen vacaciones planificadas. Para cada empleado muestra las horas de vacaciones planificadas para cada semana. [Ver imagen.](#)
 - **Formación.** Línea temporal con los empleados que tienen horas de formación planificadas. Para cada empleado muestra la formación que va a hacer y las horas planificadas para cada semana. [Ver imagen.](#)
 - **Empleados freelance.** Línea temporal con los diferentes empleados freelance y el detalle de sus tareas: proyecto en el que están trabajando, vacaciones, etc. [Ver imagen.](#)

- **Empleados en el banquillo.** Línea temporal con todos los empleados, agrupados por el puesto que ocupan en la empresa, y las horas que tienen pendientes de asignar para cada semana. [Ver imagen.](#)
 - **Resumen de clientes.** Línea temporal que muestra para cada cliente las horas de trabajo planificadas para cada semana. [Ver imagen.](#)
 - **Tareas por cubrir.** Línea temporal que muestra las tareas que faltan por asignar, agrupadas por proyecto. Para cada tipo de puesto por cubrir se muestran las horas previstas para cada semana. [Ver imagen.](#)
- Existe una pestaña que desglosa las **tareas de cada empleado.** En ella se puede ver una línea temporal con las diferentes tareas que tiene cada empleado (tareas de proyecto, vacaciones, formación, horas en el banquillo, ...) y el número de horas a la semana para cada una de ellas.

Figura 2.2

Detalle de las tareas de los empleados

Name	Status	Cost Centre	Role	Manager	Client	Project Name	Billable	Availability	Week 11 Oct-22	Week 12 Nov-22	Week 13 Dec-22	Week 14 Jan-23	Week 15 Feb-23	Week 16 Mar-23	Week 17 Apr-23	Week 18 May-23				
Employee Name	Active	VTEX	Commerce FE Dev	Management, Spain	soft	Pivotree	1	0	0	4	4	4	16	40	32	0	8	40	40	40
Employee Name	Active	VTEX	Commerce FE Dev	Management, Spain	hard	Spartanware World	1	8	24	36	36	36								
Employee Name	Active	VTEX	Commerce FE Dev	Management, Spain	hard	Pivotree Commercial	1	16												
Employee Name	Active	VTEX	Commerce FE Dev	Management, Spain	hard	Pivotree	1	32				24	8	40	32					
Employee Name	Active	VTEX	Commerce FE Dev	Management, Spain	hard	Pivotree	1	32												

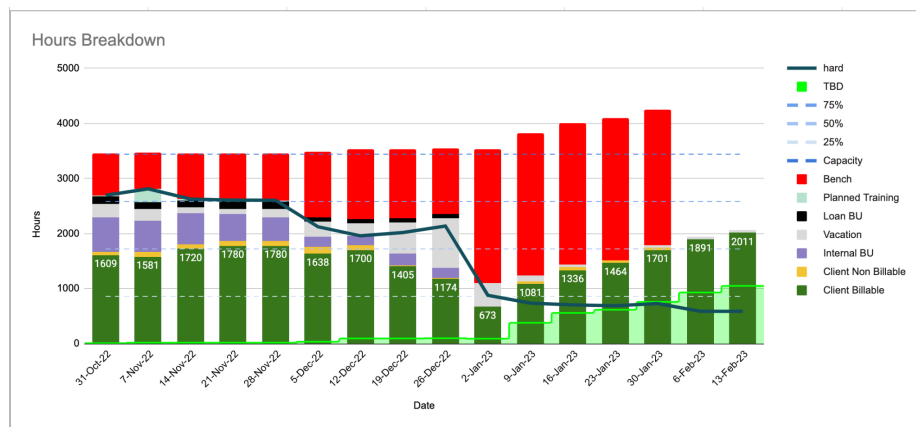
[Ver imagen a tamaño completo](#)

- Existe una pestaña que muestra diferentes **estadísticas generales.**

Por un lado, hay un gráfico combinado que muestra el desglose de horas semanales.

Figura 2.3

Gráfica que desglosa las horas por semana



[Ver imagen a tamaño completo](#)

Para cada semana se muestra, mediante una gráfica de barras, el total de horas por semana y se desglosa de la siguiente manera:

- Número de horas de cliente facturables (verde)
- Número de horas de cliente no facturables (amarillo)
- Número de horas dedicadas a proyectos internos (morado)
- Número de horas planificadas de vacaciones y permisos (gris)

- Número de horas de empleados cedidos a otra unidad de negocio² de la empresa (negro)
- Número de horas planificadas de formación (verde claro)
- Número de horas sin asignar (rojo)

Además, mediante una gráfica de líneas se muestra:

- Número de horas planificadas inamovibles (*hard*).
- Número de horas planificadas sin asignar a ningún empleado (TBD).

Esta representación de los datos permite rápidamente identificar cuando se está en una situación desfavorable; una gráfica con mucho rojo indica que hay muchos empleados parados sin aprovechar, una gráfica con poco verde indica que no se está facturando suficiente. En cuanto a las horas planificadas sin asignar (TBD), lo ideal es que en las fechas próximas sean pocas y conforme se avanza en el tiempo puedan aumentar ligeramente.

Por otro lado, además del gráfico hay una tabla que muestra la misma información. Esta representación alternativa es útil para conocer con más exactitud el número de horas, cosa que en el gráfico puede resultar difícil de leer.

Figura 2.4

Tabla que desglosa las horas por semana

J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
	31-Oct-22	7-Nov-22	14-Nov-22	21-Nov-22	28-Nov-22	5-Dec-22	12-Dec-22	19-Dec-22	26-Dec-22	2-Jan-23	9-Jan-23	16-Jan-23	23-Jan-23	30-Jan-23	6-Feb-23	13-Feb-23	
Capacity	3600	3600	3600	3600	3600	3600	3600	3600	3600	3600	3600	3600	3600	3600	3600	3600	
25%	900	900	900	900	900	900	900	900	900	900	900	900	900	900	900	900	
50%	1750	1750	1750	1750	1750	1750	1750	1750	1750	1750	1750	1750	1750	1750	1750	1750	
75%	2550	2550	2550	2550	2550	2550	2550	2550	2550	2550	2550	2550	2550	2550	2550	2550	
TBD	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
hard	2600	2600	2600	2600	2600	2600	2600	2600	2600	2600	2600	2600	2600	2600	2600	2600	
soft	700	700	700	700	700	700	700	700	700	700	700	700	700	700	700	700	
Bench	700	700	700	700	700	700	700	700	700	700	700	700	700	700	700	700	
Billable	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Client Billable	2600	2600	2600	2600	2600	2600	2600	2600	2600	2600	2600	2600	2600	2600	2600	2600	
Client Non Billable	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Internal BU	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Loan BU	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Planned Training	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Vacation	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Billable		7300			6237			6255									
Non-Billable		6471			7863			1343									
		1365			1432			1998									

[Ver imagen a tamaño completo.](#)

Por último, se muestra una análisis de diferentes KPIs (*Key Performance Indicator*):

- Pronóstico de utilización (*Forecast Utilization*): porcentaje de horas planificadas.
- Banquillo (*Bench*): porcentaje de horas de empleados sin asignar.

² Unidades organizativas en las que se divide una empresa que tienen una estrategia autónoma, pero no totalmente independiente del resto de la empresa; todas las estrategias de las distintas unidades de negocio se encuadran dentro de la estrategia global de la empresa.

- Asignaciones inamovibles (*Hard Booking*): porcentaje de horas asignadas que no se pueden modificar.
- Trabajo facturable (*Chargeability*): porcentaje de trabajo facturable.
- Asignaciones flexibles (*Soft Booking*): porcentaje de horas asignadas que se pueden modificar y asignar a otro empleado.

Estos KPIs se obtienen para un rango de fechas. Este rango se puede definir utilizando las casillas “Start week” y “End week”.

Figura 2.5

Análisis de KPIs

KPI Analysis	Forecasted Utilization	Bench	Hard Booking %	Chargeability %	Soft Booking%
	77.0%	80.0%	77.0%	80.0%	77.0%
Start Week	1	End Week	4	Helios review rolling 4 weeks KPI data by editing YELLOW highlighted columns	

[Ver imagen a tamaño completo](#)

- Además de la pestaña que muestra las estadísticas generales, existen diferentes pestañas que muestran las mismas **estadísticas por puesto**. Cada una de estas pestañas muestra la misma información que se ha detallado en el punto anterior, pero para un puesto en concreto. Hay una pestaña de estadísticas para cada uno de los siguientes puestos: business analysts, quality assurance, business consultants, front-end developers, back-end developers, developer leads y project managers.
- Tanto la pestaña que muestra las estadísticas por puesto como la que muestra las asignaciones de cada empleado permiten **filtrar los datos** mediante diferentes criterios.

Figura 2.6

Filtros de estadísticas

Name	Status	Cost Centre	Role	Manager	Client	Project Name	Billable
			Commerce FE Dev	Montañana, Laura			

[Ver imagen a tamaño completo](#)

- Nombre. Permite visualizar los datos para un empleado en concreto.
- Estado. Si se filtra por “Active” permite visualizar los datos de empleados activos. Si se filtra por “TBD” se muestran solo los datos de horas por asignar de un empleado.
- Centro de costes. Permite ver los datos de un centro de costes concreto.
- Puesto. Permite ver los datos para un puesto determinado.
- Mánager. Permite que cada mánager filtre por su nombre para ver los datos de los empleados que gestiona.
- Cliente. Permite ver los datos relacionados con un cliente determinado.
- Proyecto. Permite ver los datos relacionados con un proyecto determinado.
- Facturable. Permite filtrar el tipo de horas que se muestran: facturables, no facturables, dedicadas a proyectos internos, vacaciones y permisos, cedidas a otra *business unit*, formación o sin asignar.

2.1.5.2 Limitaciones de la solución actual

- La definición de la jornada laboral no se hace para un rango de fechas. Se aplica de forma global e inmediata.
- En la pestaña de planificación de los proyectos se muestran todos los proyectos. No hay una forma sencilla de filtrar y ordenar la información que se muestra.
- Cuando se filtran o se reordenan los datos a mostrar la vista cambia para todos los usuarios de la hoja de cálculo.
- A la hora de asignar una tarea a un empleado, no se tiene en cuenta si ha llegado al límite de su jornada laboral. Se le puede asignar a un empleado más horas de las que le corresponde y no se avisa.
- No se indica cuando se sobrepasa la jornada laboral de un empleado. Esto puede verse en la tabla resumen de las horas asignadas a cada empleado, pero solo si se conoce la duración de su jornada laboral y se detecta que el número de horas que tiene asignadas es superior. Por ejemplo, si sabemos que la jornada laboral de Jason es de 40 horas semanales nos daremos cuenta que está por encima de su capacidad la semana del 7 de noviembre al consultar el listado de la figura 2.7.

Figura 2.7

Ejemplo de empleado que sobrepasa su jornada laboral

		Employee Summary											
Role	Name	SUM of 31-Oct-22	SUM of 7-Nov-22	SUM of 14-Nov-22	SUM of 21-Nov-22	SUM of 28-Nov-22	SUM of 5-Dec-22	SUM of 12-Dec-22	SUM of 19-Dec-22	SUM of 26-Dec-22	SUM of 2-Jan-23	SUM of 9-Jan-23	SUM of 16-Jan-23
<input checked="" type="checkbox"/> Commerce Architect	Bonnes, Adam	40	0	40	40	40	40	40	8	16	0	0	0
<input type="checkbox"/> Commerce Architect	Cauffman, Jason	40	52	40	40	40	40	40	24	0	40	40	40

- Al mostrar únicamente semanas en la línea temporal no es posible especificar en detalle la dedicación por días. Esto hace que se pierda información que podría ser de utilidad.
- La replicación de los datos es manual. Por ejemplo, por un lado se introduce el detalle de las tareas de un empleado y por otro, hay que sumar a mano e introducir el total en la tabla de resumen de las horas asignadas a cada empleado.

2.1.6 Relación entre usuario y empleado

Llegados a este punto vamos a comentar una cuestión que resulta fundamental en el diseño de la aplicación, y por ende en la especificación de los requisitos. Se trata de decidir la relación entre usuario de la aplicación y empleado. Como sabemos bien eResourceKeeper va a gestionar recursos humanos, es decir empleados. Por otro lado, el acceso a la aplicación se hará mediante cuentas de usuario.

En una solución plenamente flexible cabría pensar, en principio, que existan empleados que no tengan acceso a la aplicación, y por tanto no tengan usuario, bien porque no la vayan a utilizar o bien porque no se quiera que accedan a la misma. Por otro lado, se puede dar también el caso de que haya personas que deban poder acceder a eResourceKeeper, pero cuyo trabajo no se planifique mediante la aplicación, de modo que serían usuarios pero no empleados.

Al ser los usuarios y los empleados entidades independientes, y en principio no coincidentes, sería preciso establecer procesos, también independientes, para el alta, baja y modificación de unos y otros, lo cual complica el diseño de la aplicación.

Una solución sencilla a este problema sería establecer que todo usuario es, en principio, empleado y todo empleado es, en principio, usuario. Cuando, por el motivo que sea, se quiera que un empleado no pueda acceder a la aplicación, bastará con desactivar el usuario correspondiente. Cuando queramos que un usuario de la aplicación no aparezca como empleado bastará con asignarle una jornada laboral nula.

Al establecer una correspondencia biunívoca entre usuario y empleado se simplifica el diseño de la aplicación, pues no necesitamos contemplar los procesos de alta, baja y modificación para cada una de las dos entidades por separado.

2.1.7 Requisitos candidatos

En este apartado se llevará a cabo una revisión de todos los requisitos que se han obtenido mediante diferentes técnicas, con el fin de obtener una lista de requisitos candidatos.

2.1.7.1 Requisitos no funcionales candidatos

Usando como base la plantilla de especificación de requisitos Volere [Robertson y Robertson, 2012] se han obtenido los siguientes requisitos no funcionales candidatos (RNFC):

Tabla 2.2

Requisitos no funcionales candidatos

Grupo 1: Requisitos de presentación (<i>Look and Feel</i>)	
ID	Descripción
RNFC01.1	La aplicación debe cumplir con la guía de estilos corporativa de Pivotree.
Grupo 2: Requisitos de usabilidad y humanidad	
ID	Descripción
RNFC02.1	La aplicación debe poder ser usada sin formación previa.
RNFC02.2	Pasado el periodo de adaptación, los usuarios deben preferir usar eResourceKeeper frente a la hoja de cálculo.
RNFC02.3	La aplicación conservará, a nivel de usuario, las preferencias de visualización de los datos.
RNFC02.4	La aplicación estará solo disponible en inglés pero debe ser fácil de internacionalizar si se desea en un futuro.
Grupo 3: Requisitos de cumplimiento	
ID	Descripción
RNFC03.1	La aplicación debe estar disponible como mínimo de lunes a viernes 24 horas al día, para dar servicio a todas las zonas horarias en las que trabaja la empresa.
Grupo 4: Requisitos operacionales y de entorno	

ID	Descripción
RNFC04.1	La aplicación deberá ser accesible desde distintos navegadores con independencia del sistema operativo y del dispositivo (PC, tablet, smartphone...).
RNFC04.2	La aplicación funcionará, al menos, en las últimas dos versiones de los siguientes navegadores: Chrome, Firefox, Edge y Safari.
Grupo 5: Requisitos de mantenimiento y soporte	
ID	Descripción
RNFC05.1	La aplicación tendrá un coste reducido de mantenimiento.
Grupo 6: Requisitos de seguridad	
ID	Descripción
RNFC06.1	La aplicación debe prevenir la introducción de datos incorrectos.
RNFC06.2	La aplicación protegerá la información confidencial de acuerdo con las leyes de privacidad relevantes y la política de seguridad de la información de Pivotree.
RNFC06.3	La aplicación utilizará HTTPS con el fin de usar un canal cifrado y que la información no pueda ser interceptada.
RNFC06.4	El acceso a la aplicación se controlará mediante autenticación usuario/contraseña.
Grupo 7: Requisitos culturales y políticos	
ID	Descripción
RNFC07.1	La aplicación guardará un registro de los días festivos oficiales de los siguientes países: España, Estados Unidos, Canadá, India, Brasil y Argentina.
RNFC07.2	Las fechas se mostrarán en formato MM/DD/AAAA.
Grupo 8: Requisitos legales	
ID	Descripción
RNFC08.1	La aplicación debe cumplir el estándar WCAG 2.1 [W3C, 2018] con un nivel de conformidad AA.

2.1.7.2 Requisitos funcionales candidatos

A continuación, se enumeran los requisitos funcionales candidatos (RFC). Salvo que se indique otra cosa, las tareas “Gestionar ...” incluyen las operaciones de alta, baja, modificación y consulta.

Los requisitos del ‘Grupo 1: Autenticación de usuarios’, tienen su origen en el requisito no funcional RNFC06.4. La autenticación de usuarios no sería en principio un requisito funcional, porque no es una funcionalidad de la aplicación esperada por ninguno de los *stakeholders*, sino una restricción de seguridad que debe cumplir eResourceKeeper. Sin embargo, dado que nos interesa documentar este requisito mediante casos de uso para poder describir el comportamiento de la aplicación, hemos optado por incluirlo en la lista de requisitos candidatos funcionales.

Tabla 2.3

Requisitos funcionales candidatos

Grupo 1: Autenticación de usuarios			
ID	Descripción	ID	Descripción
RFC01.1	Identificarse en la aplicación (<i>login</i>).	RFC01.3	Recuperar contraseña
RFC01.2	Salir de la aplicación (<i>logout</i>).	RFC01.4	Que un usuario pueda actualizar su contraseña
Grupo 2: Gestionar los usuarios			
ID	Descripción	ID	Descripción
RFC02.1	Gestionar los usuarios.	RFC02.3	Ordenar los usuarios por ID, nombre o puesto.

RFC02.2	Definir en detalle la jornada parcial de un usuario.	RFC02.4	Buscar un usuario por su nombre.
Grupo 3: Gestionar los proyectos			
ID	Descripción	ID	Descripción
RFC03.1	Gestionar los proyectos.	RFC03.3	Buscar un proyecto por su nombre.
RFC03.2	Ordenar los proyectos por nombre o cliente.		
Grupo 4: Gestionar las entidades auxiliares			
ID	Descripción	ID	Descripción
RFC04.1	Gestionar los clientes.	RFC04.6	Asignar un calendario de festivos a una localización.
RFC04.2	Gestionar las plataformas.	RFC04.7	Asignar a mano días festivos a una localización.
RFC04.3	Gestionar los centros de costes.	RFC04.8	Gestionar los tipos de permisos.
RFC04.4	Gestionar los puestos.	RFC04.9	Gestionar los proveedores.
RFC04.5	Gestionar las localizaciones.	RFC04.10	Gestionar los tipos de empleado.
Grupo 5: Gestionar las tareas y la jornada laboral de los usuarios			
ID	Descripción	ID	Descripción
RFC05.1	Gestionar las tareas de un usuario (alta, baja y modificación).	RFC05.5	Consultar las horas que tienen por asignar todos los usuarios (usuarios en el banquillo).
RFC05.2	Consultar las tareas de todos los usuarios.	RFC05.6	Consultar para un usuario el total de horas pendientes de asignar para el rango de fechas establecido.
RFC05.3	Ordenar las tareas de todos los usuarios por nombre, puesto u horas por asignar	RFC05.7	Indicar cuando un empleado ha superado su jornada laboral.
RFC05.4	Que un usuario pueda consultar sus tareas.		
Grupo 6: Buscar las tareas de los usuarios			
ID	Descripción	ID	Descripción
RFC06.1	Buscar las tareas de usuarios por nombre.	RFC06.7	Buscar las tareas de usuarios por proyecto.
RFC06.2	Buscar las tareas de usuarios por manager.	RFC06.8	Buscar las tareas de usuarios por cliente.
RFC06.3	Buscar las tareas de usuarios por puesto.	RFC06.9	Buscar las tareas de usuarios por tipo de tarea.
RFC06.4	Buscar las tareas de usuarios por estado.	RFC06.10	Buscar las tareas de usuarios por tipo de facturación.
RFC06.5	Buscar las tareas de usuarios por tipo de empleado.	RFC06.11	Guardar configuraciones de búsquedas sobre las tareas de los usuarios.
RFC06.6	Buscar las tareas de usuarios por centro de costes.	RFC06.12	Consultar una búsqueda guardada de tareas de los usuarios.
Grupo 7: Gestionar los días de permiso			
ID	Descripción	ID	Descripción
RFC07.1	Solicitar días de permiso (vacaciones, maternidad, baja por enfermedad).	RFC07.3	Consultar los días de permiso que tiene pendientes de aprobar/denegar para su equipo.
RFC07.2	Que un usuario pueda consultar los días de permiso solicitados.	RFC07.4	Aprobar/denegar una solicitud de días de permiso.
Grupo 8: Consultar la planificación los proyectos			
ID	Descripción	ID	Descripción
RFC08.1	Consultar la planificación de todos los proyectos activos.	RFC08.5	Gestionar las fases de un proyecto (alta, baja y modificación).
RFC08.2	Consultar la planificación de todos los proyectos inactivos.	RFC08.6	Mostrar las fases de un proyecto en la planificación.
RFC08.3	Ordenar la planificación de proyectos por nombre, cliente o fecha de alta.	RFC08.7	Consultar para un proyecto el número total de horas asignadas a cada usuario.
RFC08.4	Que un usuario pueda consultar la planificación de todos los proyectos activos y confirmados en los que participa.	RFC08.8	Intercambiar un usuario por otro en la planificación de un proyecto.
Grupo 9: Buscar planificaciones de proyecto			
ID	Descripción	ID	Descripción

RFC09.1	Buscar la planificación de proyectos por nombre.	RFC09.9	Buscar la planificación de proyectos con tareas tentativas.
RFC09.2	Buscar la planificación de proyectos por estado.	RFC09.10	Buscar la planificación de proyectos por puesto.
RFC09.3	Buscar la planificación de proyectos por <i>project manager</i> .	RFC09.11	Buscar la planificación de proyectos por plataforma.
RFC09.4	Buscar la planificación de proyectos por <i>delivery manager</i> .	RFC09.12	Guardar configuraciones de búsquedas sobre la planificación de proyectos.
RFC09.5	Buscar la planificación de proyectos por cliente.	RFC09.13	Consultar una búsqueda guardada de planificación de proyectos.
RFC09.6	Buscar la planificación de proyectos por tipo de facturación.	RFC09.14	Añadir un proyecto a favoritos.
RFC09.7	Buscar la planificación de proyectos tentativos.	RFC09.15	Consultar la planificación de proyectos favoritos.
RFC09.8	Buscar la planificación de proyectos por tipo de empleado.		
Grupo 10: Personalizar la línea de tiempo			
ID	Descripción	ID	Descripción
RFC10.1	Establecer un rango de fechas para mostrar la línea de tiempo.	RFC10.2	Seleccionar la granularidad de la línea de tiempo: días o semanas.
Grupo 11: Consultar estadísticas			
ID	Descripción	ID	Descripción
RFC11.1	Consultar las estadísticas de desglose de horas.	RFC11.8	Buscar las estadísticas por centro de costes.
RFC11.2	Establecer un rango de fechas para las estadísticas de desglose de horas.	RFC11.9	Buscar las estadísticas por proyecto.
RFC11.3	Buscar las estadísticas por nombre del empleado.	RFC11.10	Buscar las estadísticas por cliente.
RFC11.4	Buscar las estadísticas por manager del empleado.	RFC11.11	Buscar las estadísticas por tipo de tarea.
RFC11.5	Buscar las estadísticas por puesto del empleado.	RFC11.12	Buscar las estadísticas por tipo de facturación.
RFC11.6	Buscar las estadísticas por estado del empleado.	RFC11.13	Guardar configuraciones de búsquedas sobre las estadísticas.
RFC11.7	Buscar las estadísticas por tipo de empleado.	RFC11.14	Consultar una búsqueda guardada de estadísticas.
Grupo 12: Importar y exportar datos			
ID	Descripción	ID	Descripción
RFC12.1	Importar usuarios en csv.	RFC12.4	Exportar la planificación de proyectos en csv.
RFC12.2	Importar proyectos en csv.	RFC12.5	Exportar las estadísticas en csv.
RFC12.3	Exportar las tareas de usuarios en csv.		

2.2 Gestión de requisitos

Una vez obtenidos los requisitos candidatos, debemos seleccionar aquellos que se van a implementar en la aplicación. Para ello fijaremos un orden de prioridades basándonos en dos criterios:

- El **coste** de su implementación
- El **valor** que cada requisito tiene para los *stakeholders*

Una vez evaluados estos dos criterios podremos hacer la priorización y con ella la selección de los requisitos.

2.2.1 Estimación del coste

El **coste** de un requisito es una medida de la complejidad de su implementación, y aunque existen múltiples técnicas para su estimación, al tratarse de un proyecto académico e individual algunas, como por ejemplo las colaborativas, no pueden aplicarse.

Seguiremos el modelo propuesto por Cohn [2006] basado en el uso de *story points*. Los *story points* son una unidad de medida relativa que intenta estimar, de forma aproximada, el coste de desarrollo. Se trata de una escala proporcional lineal, de modo que un requisito cuyo coste estimado es de 2 *story points* tendrá un coste de desarrollo doble que otro con un coste de 1 *story point*. El modelo de Cohn no intenta estimar el tiempo de desarrollo, que dependerá de la velocidad del equipo desarrollador, pero se supone que para un mismo equipo el tiempo será proporcional al número de *story points*.

Puesto que las estimaciones generalmente tienen un elevado margen de error, Cohn propone restringir los valores posibles en la escala de *story points* a los enteros de la serie de Fibonacci (1, 2, 3, 5, 8,...), ya que de ese modo la resolución disminuye a medida que aumenta la magnitud del valor estimado. Cohn sugiere también empezar evaluando un requisito de complejidad media, asignándole un valor de 5 *story points*, de forma que a partir de él se vayan estimando por comparación el resto de requisitos.

Siguiendo estas recomendaciones hemos establecido para el requisito "Identificarse en la aplicación (*login*)" (RFC01.1) un coste de 5 *story points*. A partir de este valor procedemos a estimar el resto de requisitos.

Por lo que respecta al coste de los requisitos no funcionales, la mayoría son globales, es decir, afectan a todos los requisitos funcionales o a gran parte de ellos. Por consiguiente, su coste se tendrá en cuenta a la hora de estimar los requisitos funcionales. Solo hay un par de requisitos no funcionales que son locales, es decir, que afectan a un número reducido de requisitos funcionales, concretamente:

- RNFC02.3 afecta a RFC05.3, RFC08.3, RFC10.1 y RFC10.2
- RNFC07.1 afecta a RFC04.6

El coste de estos requisitos no funcionales se ha incluido en la estimación de los requisitos funcionales correspondientes.

2.2.2 Estimación del valor

En una situación ideal el **valor** de cada requisito debería hacerse mediante una encuesta a los *stakeholders* pero, debido al carácter académico del proyecto y la escasez de recursos, hemos optado por realizar una valoración subjetiva personal basándonos en la experiencia y conocimientos sobre el funcionamiento de la empresa y la actual hoja de cálculo.

Definiremos el valor que tiene cada requisito utilizando la técnica MoSCoW desarrollada por Clegg [Clegg y Barker, 1994], que utiliza cuatro niveles de prioridad, que ordenados de mayor a menor quedan como sigue:

- **Musthave:** son esenciales, sin estos requisitos el producto sería un fracaso.
- **Shouldhave:** son importantes pero no críticos, es posible prescindir de ellos temporalmente.
- **Couldhave:** también llamados *nice to have*, son requisitos no prioritarios pero que mejoran el resultado final del producto. Por lo general, se incluirán si el tiempo y los recursos lo permiten.
- **Won't have:** requisitos que no es apropiado incluir de momento. Se pueden descartar sin que se vea afectado el éxito del producto.

Los requisitos no funcionales los consideraremos todos una parte esencial de la aplicación, por lo que desde el punto de vista de su valor se considera que todos están en el nivel *Musthave*.

2.2.3 Estimación de los requisitos funcionales candidatos

A continuación se detalla la estimación del coste y el valor de cada requisito candidato:

Tabla 2.4

Estimación de los requisitos funcionales candidatos

Grupo 1: Acceder a la aplicación			
ID	Descripción	SP	Valor
RFC01.1	Identificarse en la aplicación (<i>login</i>).	5	<i>Musthave</i>
RFC01.2	Salir de la aplicación (<i>logout</i>).	2	<i>Musthave</i>
RFC01.3	Recuperar contraseña	3	<i>Shouldhave</i>
RFC01.4	Que un usuario pueda actualizar su contraseña	2	<i>Musthave</i>
Grupo 2: Gestionar los usuarios			
ID	Descripción	SP	Valor
RFC02.1	Gestionar los usuarios.	8	<i>Musthave</i>
RFC02.2	Definir en detalle la jornada parcial de un usuario.	2	<i>Shouldhave</i>
RFC02.3	Ordenar los usuarios por ID, nombre o puesto.	1	<i>Couldhave</i>
RFC02.4	Buscar un usuario por su nombre.	3	<i>Shouldhave</i>
Grupo 3: Gestionar los proyectos			
ID	Descripción	SP	Valor
RFC03.1	Gestionar los proyectos.	8	<i>Musthave</i>
RFC03.2	Ordenar los proyectos por nombre o cliente.	1	<i>Couldhave</i>
RFC03.3	Buscar un proyecto por su nombre.	3	<i>Shouldhave</i>
Grupo 4: Gestionar las entidades auxiliares			
ID	Descripción	SP	Valor
RFC04.1	Gestionar los clientes.	5	<i>Musthave</i>
RFC04.2	Gestionar las plataformas.	5	<i>Won't have</i>
RFC04.3	Gestionar los centros de costes.	5	<i>Won't have</i>
RFC04.4	Gestionar los puestos.	5	<i>Won't have</i>

RFC04.5	Gestionar las localizaciones.	5	<i>Musthave</i>
RFC04.6	Asignar un calendario de festivos a una localización.	8	<i>Couldhave</i>
RFC04.7	Asignar a mano días festivos a una localización.	2	<i>Musthave</i>
RFC04.8	Gestionar los tipos de permisos.	5	<i>Won't have</i>
RFC04.9	Gestionar los proveedores.	5	<i>Shouldhave</i>
RFC04.10	Gestionar los tipos de empleado.	5	<i>Won't have</i>
Grupo 5: Gestionar las tareas y la jornada laboral de los usuarios			
ID	Descripción	SP	Valor
RFC05.1	Gestionar las tareas de un usuario (alta, baja y modificación).	8	<i>Musthave</i>
RFC05.2	Consultar las tareas de los usuarios.	13	<i>Musthave</i>
RFC05.3	Ordenar las tareas por nombre, puesto u horas por asignar	2	<i>Couldhave</i>
RFC05.4	Que un usuario pueda consultar sus tareas.	2	<i>Shouldhave</i>
RFC05.5	Consultar las horas que tienen por asignar todos los usuarios (usuarios en el banquillo).	5	<i>Musthave</i>
RFC05.6	Consultar para un usuario el total de horas pendientes de asignar para el rango de fechas establecido.	2	<i>Couldhave</i>
RFC05.7	Indicar cuando un empleado está por encima de su jornada laboral.	3	<i>Shouldhave</i>
Grupo 6: Buscar las tareas de los usuarios			
ID	Descripción	SP	Valor
RFC06.1	Buscar las tareas de usuarios por nombre.	3	<i>Musthave</i>
RFC06.2	Buscar las tareas de usuarios por manager.	3	<i>Musthave</i>
RFC06.3	Buscar las tareas de usuarios por puesto.	3	<i>Musthave</i>
RFC06.4	Buscar las tareas de usuarios por estado.	3	<i>Won't have</i>
RFC06.5	Buscar las tareas de usuarios por tipo de empleado.	3	<i>Couldhave</i>
RFC06.6	Buscar las tareas de usuarios por centro de costes.	3	<i>Won't have</i>
RFC06.7	Buscar las tareas de usuarios por proyecto.	3	<i>Couldhave</i>
RFC06.8	Buscar las tareas de usuarios por cliente.	3	<i>Couldhave</i>
RFC06.9	Buscar las tareas de usuarios por tipo de tarea.	3	<i>Musthave</i>
RFC06.10	Buscar las tareas de usuarios por tipo de facturación.	3	<i>Couldhave</i>
RFC06.11	Guardar configuraciones de búsquedas sobre las tareas de usuarios.	5	<i>Couldhave</i>
RFC06.12	Consultar una búsqueda guardada de tareas de usuarios.	2	<i>Couldhave</i>
Grupo 7: Gestionar los días de permiso			
ID	Descripción	SP	Valor
RFC07.1	Solicitar días de permiso (vacaciones, maternidad, baja por enfermedad).	2	<i>Shouldhave</i>
RFC07.2	Que un usuario pueda consultar los días de permiso solicitados.	2	<i>Shouldhave</i>
RFC07.3	Consultar los días de permiso que tiene pendientes de aprobar/denegar para su equipo.	2	<i>Shouldhave</i>
RFC07.4	Aprobar/denegar una solicitud de días de permiso.	1	<i>Shouldhave</i>
Grupo 8: Consultar la planificación los proyectos			
ID	Descripción	SP	Valor
RFC08.1	Consultar la planificación de todos los proyectos activos.	13	<i>Musthave</i>
RFC08.2	Consultar la planificación de todos los proyectos inactivos.	2	<i>Shouldhave</i>
RFC08.3	Ordenar la planificación de proyectos por nombre, cliente o fecha de alta.	1	<i>Couldhave</i>
RFC08.4	Que un usuario pueda consultar la planificación de todos los proyectos activos y confirmados en los que participa.	2	<i>Shouldhave</i>
RFC08.5	Gestionar las fases de un proyecto (alta, baja y modificación).	5	<i>Couldhave</i>
RFC08.6	Mostrar las fases de un proyecto en la planificación.	3	<i>Couldhave</i>
RFC08.7	Consultar para un proyecto el número total de horas asignadas a cada usuario.	2	<i>Couldhave</i>

RFC08.8	Intercambiar un usuario por otro en la planificación de un proyecto.	5	Won't have
Grupo 9: Buscar planificaciones de proyecto			
ID	Descripción	SP	Valor
RFC09.1	Buscar la planificación de proyectos por nombre.	3	Musthave
RFC09.2	Buscar la planificación de proyectos por estado.	3	Couldhave
RFC09.3	Buscar la planificación de proyectos por <i>project manager</i> .	3	Shouldhave
RFC09.4	Buscar la planificación de proyectos por <i>delivery manager</i> .	3	Shouldhave
RFC09.5	Buscar la planificación de proyectos por cliente.	3	Musthave
RFC09.6	Buscar la planificación de proyectos por tipo de facturación.	3	Musthave
RFC09.7	Buscar la planificación de proyectos tentativos.	3	Musthave
RFC09.8	Buscar la planificación de proyectos por tipo de empleado.	3	Couldhave
RFC09.9	Buscar la planificación de proyectos con tareas tentativas.	3	Musthave
RFC09.10	Buscar la planificación de proyectos por puesto.	3	Won't have
RFC09.11	Buscar la planificación de proyectos por plataforma.	3	Musthave
RFC09.12	Guardar configuraciones de búsquedas sobre la planificación de proyectos.	5	Couldhave
RFC09.13	Consultar una búsqueda guardada de planificación de proyectos.	2	Couldhave
RFC09.14	Añadir un proyecto a favoritos.	2	Couldhave
RFC09.15	Consultar la planificación de proyectos favoritos.	2	Couldhave
Grupo 10: Personalizar la línea de tiempo			
ID	Descripción	SP	Valor
RFC10.1	Establecer un rango de fechas para mostrar la línea de tiempo.	3	Shouldhave
RFC10.2	Seleccionar la granularidad de la línea de tiempo: días o semanas.	5	Couldhave
Grupo 11: Consultar estadísticas			
ID	Descripción	SP	Valor
RFC11.1	Consultar las estadísticas de desglose de horas.	13	Shouldhave
RFC11.2	Establecer un rango de fechas para las estadísticas de desglose de horas.	3	Shouldhave
RFC11.3	Buscar las estadísticas por nombre del empleado.	3	Couldhave
RFC11.4	Buscar las estadísticas por manager del empleado.	3	Shouldhave
RFC11.5	Buscar las estadísticas por puesto del empleado.	3	Shouldhave
RFC11.6	Buscar las estadísticas por estado del empleado.	3	Won't have
RFC11.7	Buscar las estadísticas por tipo de empleado.	3	Won't have
RFC11.8	Buscar las estadísticas por centro de costes.	3	Won't have
RFC11.9	Buscar las estadísticas por proyecto.	3	Shouldhave
RFC11.10	Buscar las estadísticas por cliente.	3	Shouldhave
RFC11.11	Buscar las estadísticas por tipo de tarea.	3	Couldhave
RFC11.12	Buscar las estadísticas por tipo de facturación.	3	Couldhave
RFC11.13	Guardar configuraciones de búsquedas sobre las estadísticas.	5	Couldhave
RFC11.14	Consultar una búsqueda guardada de estadísticas.	2	Couldhave
Grupo 12: Importar y exportar datos			
ID	Descripción	SP	Valor
RFC12.1	Importar usuarios en csv.	5	Won't have
RFC12.2	Importar proyectos en csv.	5	Won't have
RFC12.3	Exportar las tareas de usuarios en csv.	5	Won't have
RFC12.4	Exportar la planificación de proyectos en csv.	5	Won't have
RFC12.5	Exportar las estadísticas en csv.	5	Won't have

A la hora de estimar el valor que aportan los requisitos del Grupo 4 (Gestionar las entidades auxiliares) se ha tenido en cuenta la frecuencia con la que se actualiza cada tipo de entidad. Se ha considerado que la gestión de las plataformas (RFC04.2), centro de costes (RFC04.3), puestos (RFC04.4) y tipos de permisos (RFC04.8) tienen una prioridad *Won't have*, dado que no se actualizan prácticamente nunca y por ello, no es crítico que se puedan gestionar por un *Admin*, de momento. Por su parte, la gestión de clientes (RFC04.1) y localizaciones (RFC04.5) es esencial, pues estas entidades cambian con frecuencia.

Para estimar el valor que aporta la búsqueda de las tareas de los usuarios y de las planificaciones de proyecto (Grupo 6 y Grupo 9) se ha valorado por separado cada uno de los atributos y se ha tenido en cuenta el uso que se hace de los filtros existentes en la solución actual.

Dentro del Grupo 11 (Consultar estadísticas) no hay requisitos *Musthave* ya que, aunque algunos pensamos que son importantes creemos que es posible prescindir temporalmente de ellos.

2.2.4. Priorización de los requisitos

Una vez evaluado el valor y el coste de cada requisito procedemos a priorizarlos con el fin de seleccionar aquellos que formarán parte del producto.

Optaremos por incluir todos los requisitos con prioridad *Musthave*, independientemente de su coste, pues se trata de funcionalidades esenciales sin las que el producto no es viable. Del mismo modo, los requisitos con prioridad *Won't have* se descartarán, aunque su coste sea muy bajo, porque se consideran prescindibles.

Para decidir si se incorporan los requisitos con prioridad *Shouldhave* y *Couldhave* se tendrá en cuenta su coste. El criterio a seguir en estos casos será el siguiente:

- Se incorporarán los requisitos de prioridad *Shouldhave* que tengan un coste ≤ 3 SP.
- Se incorporarán los requisitos de prioridad *Couldhave* que tengan un coste ≤ 2 SP.

2.2.5 Selección de requisitos

Aplicando los criterios que acabamos de describir a la lista de requisitos funcionales candidatos obtenemos la lista de requisitos funcionales seleccionados (RFS).

Los requisitos RFC06.12, RFC08.6, RFC09.13, RFC11.4, RFC11.5 y RFC11.14 no han sido seleccionados, a pesar de cumplir la condición de coste y valor establecida, porque cada uno de ellos depende de otro requisito que ha sido descartado en la selección (RFC06.11, RFC08.5, RFC09.12, RFC11.1, RFC11.1 y RFC11.13, respectivamente).

Aunque anteriormente hemos establecido que todos los requisitos no funcionales tienen un valor *Musthave* y por tanto, habría incluir todos, al existir una relación de dependencia entre

el requisito no funcional RNFC07.1 y el requisito funcional candidato RFC04.6 que ha quedado descartado nos vemos obligados a descartar dicho requisito no funcional.

Tabla 2.5

Requisitos funcionales seleccionados

Grupo 1: Acceder a la aplicación			
ID	Descripción	SP	Valor
RFS01.1	Identificarse en la aplicación (<i>login</i>).	5	<i>Musthave</i>
RFS01.2	Salir de la aplicación (<i>logout</i>).	2	<i>Musthave</i>
RFS01.3	Recuperar contraseña	3	<i>Shouldhave</i>
RFC01.4	Que un usuario pueda actualizar su contraseña	2	<i>Musthave</i>
Grupo 2: Gestionar los usuarios			
ID	Descripción	SP	Valor
RFS02.1	Gestionar los usuarios (alta, baja, modificación y consulta).	8	<i>Musthave</i>
RFS02.2	Definir en detalle la jornada parcial de un usuario.	2	<i>Shouldhave</i>
RFS02.3	Ordenar los usuarios por ID, nombre o puesto.	1	<i>Couldhave</i>
RFS02.4	Buscar un usuario por su nombre.	3	<i>Shouldhave</i>
Grupo 3: Gestionar los proyectos			
ID	Descripción	SP	Valor
RFS03.1	Gestionar los proyectos (alta, baja, modificación y consulta).	8	<i>Musthave</i>
RFS03.2	Ordenar los proyectos por nombre o cliente.	1	<i>Couldhave</i>
RFS03.3	Buscar un proyecto por su nombre.	3	<i>Shouldhave</i>
Grupo 4: Gestionar las entidades auxiliares			
ID	Descripción	SP	Valor
RFS04.1	Gestionar los clientes (alta, baja, consulta y modificación).	5	<i>Musthave</i>
RFS04.2	Gestionar las localizaciones (alta, baja, consulta y modificación).	5	<i>Musthave</i>
RFS04.3	Asignar a mano días festivos a una localización.	2	<i>Musthave</i>
Grupo 5: Gestionar las tareas y la jornada laboral de los usuarios			
ID	Descripción	SP	Valor
RFS05.1	Gestionar las tareas de un usuario (alta, baja y modificación).	8	<i>Musthave</i>
RFS05.2	Consultar las tareas de los usuarios.	13	<i>Musthave</i>
RFS05.3	Ordenar las tareas por nombre, puesto u horas por asignar	2	<i>Couldhave</i>
RFS05.4	Que un usuario pueda consultar sus tareas.	2	<i>Shouldhave</i>
RFS05.5	Consultar las horas que tienen por asignar todos los usuarios (usuarios en el banquillo).	5	<i>Musthave</i>
RFS05.6	Consultar para un usuario el total de horas pendientes de asignar para el rango de fechas establecido.	2	<i>Couldhave</i>
RFS05.7	Indicar cuando un usuario está por encima de su jornada laboral.	3	<i>Shouldhave</i>
Grupo 6: Buscar las tareas de los usuarios			
ID	Descripción	SP	Valor
RFS06.1	Buscar las tareas de usuarios por nombre.	3	<i>Musthave</i>
RFS06.2	Buscar las tareas de usuarios por manager.	3	<i>Musthave</i>
RFS06.3	Buscar las tareas de usuarios por puesto.	3	<i>Musthave</i>
RFS06.4	Buscar las tareas de usuarios por tipo de tarea.	3	<i>Musthave</i>
Grupo 7: Gestionar los días de permiso			
ID	Descripción	SP	Valor
RFS07.1	Solicitar días de permiso (vacaciones, maternidad, baja por enfermedad).	2	<i>Shouldhave</i>

RFS07.2	Que un usuario pueda consultar los días de permisos solicitados.	2	<i>Shouldhave</i>
RFS07.3	Consultar los días de permisos que tiene pendientes de aprobar/denegar para su equipo.	2	<i>Shouldhave</i>
RFS07.4	Aprobar/denegar una solicitud de días de permiso.	1	<i>Shouldhave</i>
Grupo 8: Consultar la planificación los proyectos			
ID	Descripción	SP	Valor
RFS08.1	Consultar la planificación de todos los proyectos activos.	13	<i>Musthave</i>
RFS08.2	Consultar la planificación de todos los proyectos inactivos.	2	<i>Shouldhave</i>
RFS08.3	Ordenar la planificación de proyectos por nombre, cliente o fecha de alta.	1	<i>Couldhave</i>
RFS08.4	Que un usuario pueda consultar la planificación de todos los proyectos activos y confirmados en los que participa.	2	<i>Shouldhave</i>
RFS08.5	Consultar para un proyecto el número total de horas asignadas a cada usuario.	2	<i>Couldhave</i>
Grupo 9: Buscar planificaciones de proyecto			
ID	Descripción	SP	Valor
RFS09.1	Buscar la planificación de proyectos por nombre.	3	<i>Musthave</i>
RFS09.2	Buscar la planificación de proyectos por cliente.	3	<i>Musthave</i>
RFS09.3	Buscar la planificación de proyectos por tipo de facturación.	3	<i>Musthave</i>
RFS09.4	Buscar la planificación de proyectos tentativos.	3	<i>Musthave</i>
RFS09.5	Buscar la planificación de proyectos con tareas tentativas.	3	<i>Musthave</i>
RFS09.6	Buscar la planificación de proyectos por plataforma.	3	<i>Musthave</i>
RFS09.7	Buscar la planificación de proyectos por <i>project manager</i> .	3	<i>Shouldhave</i>
RFS09.8	Buscar la planificación de proyectos por <i>delivery manager</i> .	3	<i>Shouldhave</i>
RFS09.9	Añadir un proyecto a favoritos.	2	<i>Couldhave</i>
RFS09.10	Consultar la planificación de proyectos favoritos.	2	<i>Couldhave</i>
Grupo 10: Personalizar la línea de tiempo			
ID	Descripción	SP	Valor
RFS10.1	Establecer un rango de fechas para mostrar la línea de tiempo.	3	<i>Shouldhave</i>

Si agrupamos los requisitos seleccionados por valor y sumamos los *story points* de cada grupo podemos extraer algunas estadísticas interesantes, como se muestra a continuación:

Tabla 2.6
Coste de los requisitos agrupados por valor

Valor	Coste (SP)	Coste (%)
<i>Musthave</i>	104	58 %
<i>Shouldhave</i>	64	36 %
<i>Couldhave</i>	11	6 %
TOTAL	179	100 %

Tal y como se recomienda en DSDM, la proporción de requisitos *Musthave* no supera el 60 % del coste total [Agile Business Consortium, 2014].

2.3 Documentación de requisitos

En esta etapa del proyecto documentaremos los requisitos de eResourceKeeper.

2.3.1 Requisitos no funcionales

Los requisitos no funcionales globales expresan restricciones de toda la aplicación. Cada vez que se implemente un requisito funcional será necesario comprobar esta lista de requisitos no funcionales y asegurarnos de que se cumplen.

Requisito	RNF01.1 Cumplir guía de estilos corporativa de Pivotree
Descripción	La aplicación debe cumplir con la guía de estilos corporativa de Pivotree.
Tipo	Presentación (<i>Look and Feel</i>)
Stakeholders	Equipo directivo
Criterios de aceptación	El departamento de marketing de Pivotree certificará que la aplicación cumple con la guía de estilos corporativa.
Requisito	RNF02.1 No necesitar formación previa
Descripción	La aplicación debe poder utilizarse sin necesidad de recibir formación previa.
Tipo	Usabilidad y humanidad
Stakeholders	<i>Capacity Manager, Project Manager, Delivery Manager, Resource Manager</i> y empleados.
Criterios de aceptación	Se realizará una encuesta para valorar la facilidad de uso de la aplicación. Los resultados deben mostrar que al menos el 75 % de los usuarios usa el producto sin dificultad después de un período de adaptación de un mes.
Requisito	RNF02.2 Preferir eResourceKeeper frente a la hoja de cálculo
Descripción	Pasado el periodo de adaptación, los usuarios deben preferir usar eResourceKeeper frente a la hoja de cálculo.
Tipo	Usabilidad y humanidad
Stakeholders	Equipo directivo
Criterios de aceptación	Se realizará una encuesta para valorar la satisfacción de los usuarios. Los resultados deben mostrar que al menos el 85 % de los usuarios prefieren usar eResourceKeeper frente a la hoja de cálculo después de un período de adaptación de dos meses.
Requisito	RNF02.3 Conservar las preferencias de visualización de los datos
Descripción	La aplicación conservará, a nivel de usuario, las preferencias de visualización de los datos.
Tipo	Usabilidad y humanidad
Stakeholders	<i>Capacity Manager, Project Manager, Delivery Manager, Resource Manager</i> y empleados.
Criterios de aceptación	Se harán pruebas para comprobar que, cuando el usuario personaliza una vista esta se mantiene en sesiones posteriores.
Requisito	RNF02.4 Fácil de internacionalizar
Descripción	La aplicación estará solo disponible en inglés pero debe ser fácil de internacionalizar si se desea en un futuro.
Tipo	Usabilidad y humanidad
Stakeholders	Patrocinador del proyecto
Criterios de aceptación	Se realizará un muestreo del código de la aplicación para comprobar que se hayan seguido los consejos sobre internacionalización establecidos por el W3C [Ishida, 2015].
Requisito	RNF03.1 Disponibilidad 24h al día de lunes a viernes
Descripción	La aplicación debe estar disponible como mínimo de lunes a viernes 24 horas al día, para dar servicio a todas las zonas horarias en las que trabaja la empresa.
Tipo	Requisitos de cumplimiento
Stakeholders	Equipo directivo y Departamento de RRHH
Criterios de aceptación	Se hará un control de calidad para comprobar que la aplicación tiene una disponibilidad superior al 97 % en el período comprendido entre las 0 h del lunes y las 24 h del viernes.
Requisito	RNF04.1 Accesible desde diferentes sistemas operativos y tipos de dispositivo
Descripción	La aplicación deberá ser accesible desde distintos navegadores web con independencia del sistema operativo y del dispositivo (PC, <i>tablet, smartphone</i> ...).

Tipo	Requisitos operacionales y de entorno
Stakeholders	Capacity Manager, Project Manager, Delivery Manager, Resource Manager y empleados.
Criterios de aceptación	Se harán pruebas para comprobar que la aplicación, es accesible mediante un navegador web desde sistemas Windows, Linux, Android, macOS e iOS. Se comprobará que todas las funcionalidades esenciales están disponibles desde PC, tablet y smartphone y que el contenido se adapta a la resolución de la pantalla.
Requisito	RNF04.2 Navegadores web soportados
Descripción	La aplicación funcionará, al menos, en las últimas dos versiones de los siguientes navegadores: Chrome, Firefox, Edge y Safari.
Tipo	Requisitos operacionales y de entorno
Stakeholders	Capacity Manager, Project Manager, Delivery Manager, Resource Manager y empleados.
Criterios de aceptación	Se harán pruebas para comprobar que todas las funcionalidades están disponibles en las últimas dos versiones de los navegadores Chrome, Firefox, Edge y Safari.
Requisito	RNF05.1 Coste reducido de mantenimiento
Descripción	La aplicación tendrá un coste reducido de mantenimiento.
Tipo	Requisitos de mantenimiento y soporte
Stakeholders	Equipo directivo
Criterios de aceptación	Se comprobará que el diseño de la aplicación permite un mantenimiento sencillo y que se dispone de documentación adecuada.
Requisito	RNF06.1 Prevenir introducción de datos incorrectos
Descripción	La aplicación debe prevenir la introducción de datos incorrectos
Tipo	Requisitos de seguridad
Stakeholders	Capacity Manager, Project Manager y Delivery Manager
Criterios de aceptación	Se realizarán pruebas para comprobar que en los formularios se informa al usuario de los campos obligatorios y del formato de los datos a introducir en cada campo, y que estos se validan una vez introducidos.
Requisito	RNF06.2 Proteger información confidencial
Descripción	La aplicación protegerá la información confidencial de acuerdo con las leyes de privacidad relevantes y la política de seguridad de la información de Pivotree.
Tipo	Requisitos de seguridad
Stakeholders	Equipo directivo y departamento de RRHH
Criterios de aceptación	Se llevará a cabo una auditoría para comprobar que el tratamiento de los datos que realiza la aplicación es conforme con la LOPD-GDD española [BOE, 2018] y sus equivalentes en Estados Unidos, Canadá, India, Brasil y Argentina, así como con la política de seguridad de la información de Pivotree.
Requisito	RNF06.3 Uso de HTTPS
Descripción	La aplicación utilizará HTTPS con el fin de usar un canal cifrado y que la información no pueda ser interceptada.
Tipo	Requisitos de seguridad
Stakeholders	Administradores de sistemas
Criterios de aceptación	Se comprobará, haciendo uso de un analizador de paquetes, que el intercambio de datos entre el cliente y el servidor se lleva a cabo dentro de una sesión HTTPS.
Requisito	RNF06.4 Autenticación obligatoria
Descripción	El acceso a la aplicación se controlará mediante autenticación usuario/contraseña.
Tipo	Requisitos de seguridad
Stakeholders	Equipo directivo y departamento de RRHH
Criterios de aceptación	Se comprobará que ninguna funcionalidad de la aplicación, salvo la recuperación de contraseña, sea accesible sin previa autenticación.
Requisito	RNF07.1 Formato de fechas
Descripción	Las fechas se mostrarán en formato MM/DD/AAAA.
Tipo	Requisitos culturales y políticos

Stakeholders	Capacity Manager
Criterios de aceptación	Se comprobará que las fechas se muestran en formato MM/DD/AAAA y que haya un <i>tooltip</i> indicándolo.
Requisito	RNF08.1 Cumplir con WCAG 2.1 AA
Descripción	La aplicación debe cumplir el estándar WCAG 2.1 con un nivel de conformidad AA.
Tipo	Requisitos legales
Stakeholders	Patrocinador del proyecto
Criterios de aceptación	Se realizará una auditoría de accesibilidad para asegurar que se cumple el estándar WCAG 2.1 [W3C, 2018] con el nivel de conformidad AA.

2.3.2 Requisitos funcionales

Para especificar los requisitos funcionales utilizaremos casos de uso siguiendo el formato que Cockburn [2000] describe como “fully dressed form”. La siguiente tabla enumera todos los casos de uso que hemos identificado a partir de los requisitos funcionales seleccionados. Para cada uno se especifican los actores que harán uso de él (*User* = U, *Viewer* = V, *Editor* = E, *Admin* = A) y los requisitos que lo motivan.

Dado que vamos a tener muchos casos de uso similares, para hacer la descripción menos repetitiva y facilitar la comprensión emplearemos en algunos de ellos lo que se conoce como casos de uso parametrizados [Cockburn, 2000], que consiste en que, ante un conjunto de casos de uso similares, primero definimos uno que sirva de plantilla con una serie de parámetros y, a continuación, detallamos cada caso de uso concreto indicando los valores que adoptan dichos parámetros.

En la tabla siguiente se muestran todos los casos de uso que vamos a considerar en nuestro proyecto, agrupando los de tipo parametrizado por la plantilla de la que derivan, como lo indica la numeración.

Tabla 2.7

Casos de uso

ID	Caso de uso	Tipo	Actores	Requisitos
CU01	Consultar <u>entidad</u>	Plantilla CU	n/a	n/a
CU01.1	Consultar usuarios	CU param.	V, E, A	RFS02.1, RFS02.3
CU01.2	Consultar proyectos	CU param.	V, E, A	RFS03.1, RFS03.2
CU01.3	Consultar clientes	CU param.	V, E, A	RFS04.1
CU01.4	Consultar localizaciones	CU param.	V, E, A	RFS04.2
CU02	Crear/modificar una <u>entidad</u>	Plantilla CU	n/a	n/a
CU02.1	Crear/modificar un usuario	CU param.	A	RFS02.1, RFS02.2
CU02.2	Crear/modificar un proyecto	CU param.	E, A	RFS03.1
CU02.3	Crear/modificar un cliente	CU param.	A	RFS04.1
CU02.4	Crear/modificar una localización	CU param.	A	RFS04.2, RFS04.3
CU03	Eliminar una <u>entidad</u>	Plantilla CU	n/a	n/a
CU03.1	Eliminar un usuario	CU param.	A	RFS02.1
CU03.2	Eliminar un proyecto	CU param.	E, A	RFS03.1
CU03.3	Eliminar un cliente	CU param.	A	RFS04.1
CU03.4	Eliminar una localización	CU param.	A	RFS04.2
CU04	Buscar una <u>entidad</u> por nombre	Plantilla CU	n/a	n/a
CU04.1	Buscar un usuario por nombre	CU param.	V, E, A	RFS02.4

CU04.2	Buscar un proyecto por nombre	CU param.	V, E, A	RFS03.3
CU05	Crear/modificar una tarea	CU normal	E, A	RFS06.1
CU06	Eliminar una tarea	CU normal	E, A	RFS06.1
CU07	Consultar sus tareas	CU normal	U, V, E, A	RFS05.4, RFS10.1, RFS05.6, RFS05.7
CU08	Consultar las tareas de todos los usuarios	CU normal	V, E, A	RFS05.2, RFS05.3, RFS10.1, RFS05.6, RFS05.7
CU09	Buscar las tareas de los usuarios por atributo	CU normal	V, E, A	RFS06.1, RFS06.2, RFS06.3, RFS06.4
CU10	Consultar las horas que tienen por asignar los usuarios	CU normal	V, E, A	RFS05.5
CU11	Consultar la planificación de los proyectos en los que participa	CU normal	U, V, E, A	RFS08.4, RFS10.1
CU12	Consultar la planificación de todos los proyectos	CU normal	V, E, A	RFS08.1, RFS08.2, RFS08.3, RFS10.1, RFS08.5
CU13	Buscar la planificación de proyectos por atributo	CU normal	V, E, A	RFS09.1, RFS09.2, RFS09.3, RFS09.4, RFS09.5, RFS09.6, RFS09.7, RFS09.8
CU14	Añadir un proyecto a favoritos	CU normal	V, E, A	RFS09.9
CU15	Consultar la planificación de proyectos favoritos	CU normal	V, E, A	RFS09.10
CU16	Consultar los días de permiso solicitados	CU normal	U, V, E, A	RFS07.2
CU17	Solicitar días de permiso	CU normal	U, V, E, A	RFS07.1
CU18	Gestionar días de permiso	CU normal	V, E, A	RFS07.3, RFS07.4
CU19	Identificarse en la aplicación (<i>login</i>)	CU normal	U, V, E, A	RFC01.1
CU20	Salir de la aplicación (<i>logout</i>)	CU normal	U, V, E, A	RFC01.2
CU21	Recuperar contraseña	CU normal	U, V, E, A	RFC01.3
CU22	Cambiar contraseña	CU normal	U, V, E, A	RFC01.4

Tenemos cuatro casos de uso de consultas que son muy similares: consulta de usuarios, consulta de proyectos, consulta de clientes y consulta de localizaciones. Para estos creamos la plantilla 'consultar entidades' (CU01), que aparece en el siguiente recuadro (los parámetros aparecen subrayados):

Plantilla: CU01 Consultar entidad

Actor principal: Admin

Stakeholders e intereses:

Capacity Manager, Project Manager, Delivery Manager y Resource Manager: quieren consultar las *entidades*

Precondición: El usuario debe haberse identificado en la aplicación

Garantías en caso de éxito: La aplicación mostrará al usuario la información de entidad

Escenario principal de éxito:

1. El usuario accede al listado de entidad dadas de alta en la aplicación.
2. La aplicación valida que el usuario tiene permiso para consultar la entidad y muestra el listado de entidad, utilizando la ordenación por defecto (fecha de la última modificación en orden descendente).
3. Opcionalmente, el usuario cambia el criterio de ordenación (campos de ordenación) y vuelve al paso 2.

Extensiones:

2a. El usuario no tiene permiso para consultar entidad.

2a1. La aplicación redirige al usuario a la página de inicio y muestra un mensaje indicándolo.

Escenarios alternativos:

- a. Ver el detalle de una entidad
 1. El usuario indica qué entidad quiere consultar.
 2. La aplicación valida que el usuario tiene permiso para consultar la entidad.
 3. La aplicación muestra la información de la entidad.

Actores con permiso: *Viewer, Editor o Admin.*

Parámetros:

- **entidad:** puede tener cuatro valores: 'usuarios', 'proyectos', 'clientes' o 'localizaciones'. Cada valor configura un caso de uso diferente.
- **campos de ordenación:** sus posibles valores dependen de **entidad**.

A continuación especificamos los cuatro casos de uso indicando los valores de los parámetros para cada uno. Todo dato no especificado en el caso de uso hereda el valor que tenía en la plantilla:

Caso de uso: CU01.1 Consultar usuarios

Escenario principal de éxito:

1. El usuario consulta los usuarios usando la plantilla **CU01 Consultar entidad**, en la que:
 - **entidad:** *usuarios*.
 - **campos de ordenación:** ID, nombre o puesto del usuario en orden ascendente y descendente.

Caso de uso: CU01.2 Consultar proyectos

Escenario principal de éxito:

1. El usuario consulta los proyectos usando la plantilla **CU01 Consultar entidad**, en la que:
 - **entidad:** *proyectos*.
 - **campos de ordenación:** nombre o cliente en orden ascendente y descendente.

Caso de uso: CU01.3 Consultar clientes

Escenario principal de éxito:

1. El usuario consulta los clientes usando la plantilla **CU01 Consultar entidad**, en la que:
 - **entidad:** *clientes*.
 - **campos de ordenación:** nombre en orden ascendente y descendente.

Caso de uso: CU01.4 Consultar localizaciones

Escenario principal de éxito:

1. El usuario consulta las localizaciones usando la plantilla **CU01 Consultar entidad**, en la que:
 - **entidad:** *localizaciones*.
 - **campos de ordenación:** nombre en orden ascendente y descendente.

Procedemos del mismo modo con el resto de las plantillas:

Plantilla: CU02 Crear/modificar una **entidad**

Precondición: El usuario debe haberse identificado en la aplicación

Garantías en caso de éxito: La aplicación crea/modifica la **entidad**

Escenario principal de éxito:

1. El usuario **consulta entidad (CU01)** e indica que quiere crear/modificar una **entidad**.
2. La aplicación valida que el usuario tiene **permiso** para crear/modificar **entidad** y muestra el formulario con los **campos a rellenar**.
3. El usuario rellena los **campos a rellenar** y envía los datos.
4. La aplicación crea la **entidad** y muestra un mensaje de confirmación de que **entidad** ha sido creada/modificada.

Extensiones:

2a. El usuario no tiene **permiso** para crear/modificar **entidad**.

2a1. La aplicación redirige al usuario a la página de inicio y muestra un mensaje indicándolo.

3a. El usuario deja algún **campo obligatorio** vacío.

3a1. La aplicación indica que no se puede dejar el **campo obligatorio** vacío y volvemos al punto 2.

3b. El usuario rellena un campo con un formato incorrecto.

3b1. La aplicación indica el formato del campo para que el usuario pueda corregir el error y volvemos al punto 2.

Parámetros:

- **entidad:** puede valer 'usuario', 'proyecto', 'tarea', 'cliente' o 'localización'. Cada valor configura un caso de uso diferente.
- **campos a rellenar:** sus posibles valores dependen de **entidad**.
- **campo obligatorio:** sus posibles valores dependen de **entidad**.
- **permiso:** puede valer 'Editor' o 'Admin'. Indica qué actor está autorizado a ejecutar el caso de uso.

Caso de uso: CU02.1 Crear/modificar un usuario

Actor principal: Admin

Stakeholders e intereses:

Capacity Manager: quiere dar de alta un usuario.

Escenario principal de éxito:

1. El usuario crea un usuario usando la plantilla **CU02 Crear una entidad**, en la que:

- **entidad:** usuario
- **permiso:** Admin.

Campos a rellenar obligatorios	Campos a rellenar opcionales
- Email - Estado - Rol - Nombre - Apellidos - Puesto - Tipo de empleado - Proveedor (solo disponible para empleados 'Contractor') - Centro de costes - Mánager (solo obligatorio para empleados de tipo 'Employee' cuando el puesto seleccionado requiere que tenga un mánager) - Localización - Fecha de inicio - Tipo de jornada - N° de horas de la jornada parcial por día (solo obligatorio si el tipo de jornada tiene el valor "jornada parcial")	- Contraseña - Fecha fin

Caso de uso: CU02.2 Crear/modificar un proyecto

Actor principal: Admin

Stakeholders e intereses:

Capacity Manager, Project Manager y Delivery Manager: quieren crear un proyecto.

Escenario principal de éxito:

1. El usuario crea un proyecto usando la plantilla **CU02 Crear/modificar una entidad**, en la que:

- **entidad:** proyecto
- **permiso:** Editor y Admin.

Campos a rellenar obligatorios	Campos a rellenar opcionales
- Nombre - Tipo - Cliente (solo disponible para proyectos de tipo 'Client') - Plataforma (solo obligatorio para proyectos de tipo 'Client') - Delivery manager - Project manager - Estado - Reserva - Tipo facturación (solo para proyectos de tipo 'Client')	Ninguno

Caso de uso: CU02.3 Crear/modificar un cliente

Actor principal: Admin

Stakeholders e intereses:

Capacity Manager: quiere crear un cliente.

Equipo directivo: quiere que se pueda fácilmente dar de alta nuevos clientes.

Escenario principal de éxito:

1. El usuario crea un cliente usando la plantilla **CU02 Crear/modificar una entidad**, en la que:

- **entidad:** cliente
- **permiso:** Admin.

Campos a rellenar obligatorios	Campos a rellenar opcionales
- Nombre - Estado	- Logotipo

Caso de uso: CU02.4 Crear/modificar una localización

Actor principal: Admin

Stakeholders e intereses:

Capacity Manager: quiere crear una localización

Departamento de RRHH: quiere que se pueda fácilmente dar de alta nuevas localizaciones.

Escenario principal de éxito:

1. El usuario crea una localización usando la plantilla **CU02 Crear/modificar una entidad**, en la que:

- **entidad:** localización
- **permiso:** Admin.

Campos a rellenar obligatorios	Campos a rellenar opcionales
- Nombre - País - Zona horaria	- Día festivo. (Cuando se añade un día festivo es obligatorio rellenar la fecha y la descripción.)

Plantilla: CU03 Eliminar una entidad

Actor principal: Admin

Precondición: El usuario debe haberse identificado en la aplicación

Garantías en caso de éxito: La aplicación eliminará la entidad

Escenario principal de éxito:

1. El usuario consulta entidad (CU01) y selecciona la entidad que quiere eliminar.
2. La aplicación valida que el usuario tiene permiso para eliminar entidad, verifica que se cumple el requisito para eliminar entidad y pide la confirmación del usuario.
3. El usuario confirma que quiere eliminar la entidad.
4. La aplicación elimina la entidad y muestra un mensaje de confirmación de que la entidad ha sido eliminada.

Extensiones:

2a. El usuario no tiene permiso para eliminar entidad.

2a1. La aplicación redirige al usuario a la página de inicio y muestra un mensaje indicándolo.

2b. No se cumple el requisito para eliminar entidad.

2b1. La aplicación muestra un mensaje indicándolo.

3a. El usuario no confirma que quiere eliminar la entidad.

3a1. La aplicación no elimina la entidad.

4a. La entidad no existe.

4a1. La aplicación muestra un mensaje de error informado de que la entidad no existe.

Parámetros:

- **entidad:** puede valer 'usuario', 'proyecto', 'tarea', 'cliente' o 'localización'. Cada valor configura un caso de uso diferente.
- **permiso:** puede valer 'Editor' o 'Admin'. Indica qué actor está autorizado a ejecutar el caso de uso.

Caso de uso: CU03.1 Eliminar un usuario

Stakeholders e intereses:

Capacity Manager: quiere eliminar un usuario

Equipo directivo y departamento RRHH: quiere que se pueda eliminar un usuario para que se cumpla la política de seguridad de la información de Pivotree.

Escenario principal de éxito:

1. El usuario elimina un usuario usando la plantilla **CU03 Eliminar una entidad**, en la que:
 - entidad: usuario
 - permiso: *Admin*.

Caso de uso: CU03.2 Eliminar un proyecto

Escenario principal de éxito:

Stakeholders e intereses:

Capacity Manager, Project Manager y Delivery Manager: quieren eliminar un proyecto

Escenario principal de éxito:

1. El usuario elimina un proyecto usando la plantilla **CU03 Eliminar una entidad**, en la que:
 - entidad: proyecto
 - permiso: *Editor y Admin*.

Caso de uso: CU03.3 Eliminar un cliente

Stakeholders e intereses:

Capacity Manager: quiere eliminar un cliente

Escenario principal de éxito:

1. El usuario elimina un cliente usando la plantilla **CU03 Eliminar una entidad**, en la que:
 - entidad: cliente
 - permiso: *Admin*

Caso de uso: CU03.4 Eliminar una localización

Stakeholders e intereses:

Capacity Manager: quiere eliminar una localización

Escenario principal de éxito:

1. El usuario elimina una localización usando la plantilla **CU03 Eliminar una entidad**, en la que:
 - entidad: localización
 - permiso: *Admin*

Plantilla: CU04 Buscar una entidad por nombre

Actor principal: *Admin*

Stakeholders e intereses:

Capacity Manager, Project Manager, Delivery Manager y Resource Manager: quieren encontrar una entidad

Precondición: El usuario debe haberse identificado en la aplicación

Garantías en caso de éxito: La aplicación encuentra la entidad

Escenario principal de éxito:

1. El usuario **consulta entidad (CU01)**.
2. El usuario introduce un valor en el campo de búsqueda de entidad y envía la búsqueda.
3. La aplicación muestra un listado de entidad coincidentes (todas, si no se ha introducido ningún valor en el paso 2).
4. El paso 2 se repite hasta que el usuario decide salir.

Actores con permiso: *Viewer, Editor o Admin*.

Parámetros:

- entidad: puede valer 'usuario' o 'proyecto'. Cada valor configura un caso de uso diferente.

Caso de uso: CU04.1 Buscar un usuario por nombre

Escenario principal de éxito:

1. El usuario busca un usuario usando la plantilla **CU04 Buscar una entidad por nombre**, en la que:

- **entidad:** usuario

Caso de uso: CU04.2 Buscar un proyecto por nombre

Escenario principal de éxito:

1. El usuario busca un proyecto usando la plantilla **CU04 Buscar una entidad por nombre**, en la que:

- **entidad:** proyecto

Plantilla: CU05 Crear/modificar una tarea

Actor principal: Admin

Stakeholders e intereses:

Capacity Manager, Project Manager y Delivery Manager: quieren crear/modificar una tarea.

Precondición: El usuario debe haberse identificado en la aplicación

Garantías en caso de éxito: La aplicación crea/modifica la entidad

Escenario principal de éxito:

1. El usuario indica que quiere crear/modificar una tarea.
2. La aplicación valida que el usuario tiene permiso para crear/modificar una tarea y muestra el formulario con los campos a rellenar.
3. El usuario rellena los campos y envía los datos.
4. La aplicación crea/modifica la tarea y muestra un mensaje de confirmación.

Extensiones:

2a. El usuario no tiene permiso para crear/modificar una tarea.

2a1. La aplicación redirige al usuario a la página de inicio y muestra un mensaje indicándolo.

3a. El usuario deja algún campo obligatorio vacío.

3a1. La aplicación indica que no se puede dejar el campo obligatorio vacío y volvemos al punto 2.

3b. El usuario rellena un campo con un formato incorrecto.

3b1. La aplicación indica el formato del campo para que el usuario pueda corregir el error y volvemos al punto 2.

3c. El usuario ha programado la tarea en un día no laborable.

3c1. La aplicación pide confirmación al usuario de que quiere crear la tarea en un día no laborable.

3c2i. El usuario no confirma que quiere crear la tarea en un día no laborable y volvemos al punto 2.

3c2ii. El usuario confirma que quiere crear la tarea en un día no laborable.

3d. El usuario ha programado una tarea que hace que se exceda la jornada laboral del usuario.

3d1. La aplicación pide confirmación al usuario de que quiere crear la tarea aunque exceda la jornada laboral.

3d2i. El usuario no confirma que quiere exceder la jornada laboral y volvemos al punto 2.

3d2ii. El usuario confirma que quiere exceder la jornada laboral.

Campos:

Campos a rellenar obligatorios	Campos a rellenar opcionales
<ul style="list-style-type: none"> - Nombre - Descripción - Reserva - Tipo facturación (solo para proyectos de tipo 'Client') - Proyecto - Empleado (solo si la reserva está confirmada) - Fecha de inicio - Fecha de fin - Horas al día 	<ul style="list-style-type: none"> - Descripción

Plantilla: CU06 Eliminar una tarea

Actor principal: Admin

Stakeholders e intereses:

Capacity Manager, Project Manager y Delivery Manager: quieren eliminar una tarea si hay cambios en la planificación

Precondición: El usuario debe haberse identificado en la aplicación

Garantías en caso de éxito: La aplicación crea/modifica la entidad

Escenario principal de éxito:

1. El usuario indica la tarea que quiere eliminar.
2. La aplicación valida que el usuario tiene permiso para eliminar tarea y pide la confirmación del usuario.
3. El usuario confirma que quiere eliminar la tarea.
4. La aplicación elimina la tarea y muestra un mensaje de confirmación de que la entidad ha sido eliminada.

Extensiones:

- 2a. El usuario no tiene permiso para eliminar tarea.
 - 2a1. La aplicación redirige al usuario a la página de inicio y muestra un mensaje indicándolo.
- 3a. El usuario no confirma que quiere eliminar la tarea.
 - 3a1. La aplicación no elimina la tarea.
- 4a. La tarea no existe.
 - 4a1. La aplicación muestra un mensaje de error informado de que la tarea no existe.

Caso de uso: CU07 Consultar sus tareas

Actor principal: *User*

Stakeholders e intereses:

Empleado y *Resource Manager*: quiere consultar sus tareas

Precondición: El usuario debe haberse identificado en la aplicación

Garantías en caso de éxito: La aplicación mostrará al usuario sus tareas

Escenario principal de éxito:

1. El usuario solicita ver sus tareas
2. La aplicación muestra el listado de tareas del usuario y el total de horas pendientes de asignar. Para cada tarea indica el número de horas asignadas por día. Si algún día el empleado está por encima de su jornada laboral la aplicación lo resalta visualmente.
3. Opcionalmente, el usuario cambia el rango de fechas para el que se muestran las tareas y vuelve al paso 2.

Escenarios alternativos:

- 2a. El usuario no tiene ninguna tarea
 - 2a1. La aplicación muestra un mensaje informando que el usuario no tiene ninguna tarea.

Caso de uso: CU8 Consultar las tareas de todos los usuarios

Actor principal: *Viewer*

Stakeholders e intereses:

Capacity Manager, Project Manager, Delivery Manager y Resource Manager: quieren consultar las tareas de los usuarios

Precondición: El usuario debe haberse identificado en la aplicación

Garantías en caso de éxito: La aplicación mostrará al usuario las tareas de todos los usuarios

Escenario principal de éxito:

1. El usuario solicita ver las tareas de todos los usuarios
2. La aplicación valida que el usuario tiene permiso para consultar las tareas de todos los usuarios y muestra el listado de tareas, agrupadas por usuario. Para cada usuario muestra el total de horas pendientes de asignar. Para cada tarea indica el número de horas asignadas por día. Si un empleado está por encima de su jornada laboral algún día la aplicación lo resalta visualmente.
3. Opcionalmente, el usuario cambia el criterio de ordenación (el nombre, puesto o número de horas por asignar en orden ascendente y descendente) y vuelve al paso 2.
4. Opcionalmente, el usuario cambia el rango de fechas para el que se muestran las tareas y vuelve al paso 2.

Extensiones:

- 2a. El usuario no tiene permiso para consultar tareas de todos los usuarios.
 - 2a1. La aplicación redirige al usuario a la página de inicio y muestra un mensaje indicándolo.

Plantilla: CU09 Buscar las tareas de los usuarios por atributo

Actor principal: *Viewer*

Stakeholders e intereses:

Capacity Manager, Project Manager, Delivery Manager y Resource Manager: quieren poder acotar el listado de tareas

Precondición: El usuario debe haberse identificado en la aplicación

Garantías en caso de éxito: La aplicación encuentra las tareas que coinciden con los atributos

Escenario principal de éxito:

1. El usuario **consulta las tareas de todos los usuarios (CU08)** e indica que quiere realizar una búsqueda.
2. El usuario selecciona/deselecciona uno de los valores disponibles para el atributo ('nombre de usuario', 'manager', 'puesto' o 'tipo de tarea').
3. La aplicación muestra un listado de las tareas coincidentes para el valor de los atributos seleccionados.
4. El paso 2 se repite hasta que el usuario decide salir.

Caso de uso: CU10 Consultar las horas que tienen por asignar los usuarios

Actor principal: Admin

Stakeholders e intereses:

Capacity Manager: quiere poder ver los usuarios que tienen horas disponibles.

Precondición: El usuario debe haberse identificado en la aplicación

Garantías en caso de éxito: La aplicación mostrará los usuarios que tienen alguna hora sin asignar y cuántas horas que tienen disponible por día.

Escenario principal de éxito:

1. El usuario solicita ver los usuarios en el banquillo.
2. La aplicación valida que el usuario tiene permiso para la consulta y muestra los usuarios que tienen alguna hora sin asignar para el rango de fechas establecido. Para cada usuario muestra el número de horas que tiene libres cada día y el total de horas por asignar.
3. Opcionalmente, el usuario cambia el criterio de ordenación del listado (nombre, puesto o número de horas por asignar en orden ascendente y descendente) y vuelve al paso 2.
4. Opcionalmente, el usuario cambia el rango de fechas para el que se muestra la información y vuelve al paso 2.

Extensiones:

2a. El usuario no tiene permiso para consultar los usuarios en el banquillo.

2a1. La aplicación redirige al usuario a la página de inicio y muestra un mensaje indicándolo.

Caso de uso: CU11 Consultar la planificación de los proyectos en los que participa

Actor principal: User

Stakeholders e intereses:

Empleado y Resource Manager: quiere consultar la planificación de los proyectos en los que participa

Precondición: El usuario debe haberse identificado en la aplicación

Garantías en caso de éxito: La aplicación mostrará la planificación de todos los proyectos en los que participa el usuario

Escenario principal de éxito:

1. El usuario solicita ver la planificación de los proyectos en los que él participa
2. La aplicación muestra la planificación de los proyectos en los que participa el usuario para el rango de fechas establecido. Para cada proyecto muestra los usuarios que participan en él. Para cada usuario muestra las tareas que tiene planificadas con una descripción, número de horas al día, fecha de inicio, fecha de fin y el total de horas planificadas.
3. Opcionalmente, el usuario cambia el rango de fechas para el que se muestra la planificación de proyectos y vuelve al paso 2.

Escenarios alternativos:

2a. El usuario no participa en ningún proyecto para el rango de fechas establecido

2a1. La aplicación muestra un mensaje informando que el usuario no tiene ningún proyecto.

Caso de uso: CU12 Consultar la planificación de todos los proyectos

Actor principal: Viewer

Stakeholders e intereses:

Capacity Manager, Project Manager, Delivery Manager y Resource Manager: quieren consultar la planificación de todos los proyectos

Precondición: El usuario debe haberse identificado en la aplicación

Garantías en caso de éxito: La aplicación mostrará la planificación de todos los proyectos

Escenario principal de éxito:

1. El usuario solicita ver la planificación de todos los proyectos

2. La aplicación valida que el usuario tiene permiso para consultar la planificación de todos los proyectos y muestra una lista, agrupada por proyectos. Para cada proyecto muestra los usuarios que participan en él. Para cada usuario muestra las tareas que tiene planificadas con una descripción, número de horas al día, fecha de inicio, fecha de fin y el total de horas planificadas.
3. Opcionalmente, el usuario cambia el criterio de ordenación de la planificación de proyectos (el nombre, cliente o fecha de alta en orden ascendente y descendente) y vuelve al paso 2.
4. Opcionalmente, el usuario cambia el rango de fechas para el que se muestra la planificación de proyectos y vuelve al paso 2.
5. Opcionalmente, el usuario indica que quiere ver la planificación de los proyectos inactivos y vuelve al paso 2 (por defecto solo se muestran los proyectos activos).

Extensiones:

2a. El usuario no tiene permiso para consultar la planificación de proyectos.

2a1. La aplicación redirige al usuario a la página de inicio y muestra un mensaje indicándolo.

Plantilla: CU013 Buscar la planificación de proyectos por atributo

Actor principal: Viewer

Stakeholders e intereses:

Capacity Manager, Project Manager, Delivery Manager y Resource Manager: quiere acotar el número de proyectos para los que consulta su planificación

Precondición: El usuario debe haberse identificado en la aplicación

Garantías en caso de éxito: La aplicación encuentra la planificación de proyecto que coinciden con los atributos

Escenario principal de éxito:

1. El usuario **consulta la planificación de todos los proyectos (CU12)** e indica que quiere realizar una búsqueda.
2. El usuario selecciona/deselecciona uno de los valores disponibles para el atributo ('nombre del proyecto', 'por cliente', 'por tipo de facturación', 'tentativos/confirmados', 'con/sin tareas tentativas', 'por plataforma', 'por project manager' o 'por delivery manager').
3. La aplicación muestra un listado de los proyectos coincidentes para el valor de los atributos seleccionados.
4. El paso 2 se repite hasta que el usuario decide salir.

Caso de uso: CU14 Añadir un proyecto a favoritos

Actor principal: Viewer

Stakeholders e intereses:

Capacity Manager, Project Manager, Delivery Manager y Resource Manager: quiere poder guardar los proyectos que consulta más frecuentemente como favorito

Precondición: El usuario debe haberse identificado en la aplicación

Garantías en caso de éxito: La aplicación guarda el proyecto como favorito

Escenario principal de éxito:

1. El usuario **consulta la planificación de todos los proyectos (CU12)**.
2. El usuario selecciona un proyecto e indica que quiere guardarlo como favorito.
3. La aplicación guarda el proyecto seleccionado como favorito en el perfil de ese usuario.

Caso de uso: CU15 Consultar la planificación de proyectos favoritos

Actor principal: Viewer

Stakeholders e intereses:

Capacity Manager, Project Manager, Delivery Manager y Resource Manager: quiere consultar los proyectos que tiene guardados como favoritos

Precondición: El usuario debe haberse identificado en la aplicación

Garantías en caso de éxito: La aplicación muestra los proyectos guardados como favoritos

Escenario principal de éxito:

1. El usuario indica que quiere ver los proyectos guardados como favoritos.
2. La aplicación muestra el listado de proyectos favoritos del usuario y su planificación.

Escenarios alternativos:

2a. El usuario no tiene ningún proyecto guardado como favorito.

2a1. La aplicación muestra un mensaje informando que el usuario no tiene ningún proyecto guardado como favorito.

Caso de uso: CU16 Consultar los días de permiso solicitados

Actor principal: *User*

Stakeholders e intereses:

Empleado: quiere consultar los días de permiso que ha solicitado

Precondición: El usuario debe haberse identificado en la aplicación

Garantías en caso de éxito: La aplicación muestra los días de permiso solicitados

Escenario principal de éxito:

1. El usuario accede al listado de días de permiso solicitados.
2. La aplicación muestra el listado de días de permiso solicitados por el usuario (tanto en futuro como en pasado) ordenados de más reciente a más antiguo. Para cada permiso muestra la fecha de inicio y de fin, el tipo de permiso y el estado.

Escenarios alternativos:

2a. El usuario no ha solicitado días de permiso.

2a1. La aplicación muestra un mensaje informando que el usuario no tiene ninguna solicitud de días de permiso.

Caso de uso: CU17 Solicitar días de permiso

Actor principal: *User*

Stakeholders e intereses:

Empleado: quiere solicitar días de permiso

Precondición: El usuario debe haberse identificado en la aplicación

Garantías en caso de éxito: La aplicación registrará la solicitud

Escenario principal de éxito:

1. El usuario indica que quiere solicitar días de permiso.
2. La aplicación muestra el formulario para solicitar días de permiso con los campos a rellenar.
3. El usuario rellena los campos del formulario (tipo de permiso, fecha de inicio, fecha de fin y comentario) y envía los datos.
4. La aplicación crea una solicitud de días de permiso con estado 'pendiente' y muestra un mensaje de confirmación.

Extensiones:

3a. El usuario deja algún campo vacío.

3a1. La aplicación indica que no se puede dejar ningún campo vacío y volvemos al punto 2.

3b. El usuario rellena algún campo con un formato incorrecto.

3a1. La aplicación indica el campo y el error para que el usuario pueda corregirlo, y volvemos al punto 2.

Caso de uso: CU18 Gestionar días de permiso

Actor principal: *Viewer*

Stakeholders e intereses:

Capacity Manager y Departamento de RRHH: quiere centralizar la planificación de proyectos y solicitud de días de permiso en una única aplicación

Resource Manager: quiere consultar los días de permiso que tiene pendientes de aprobar y aprobarlos o denegarlos, según proceda.

Precondición: El usuario debe haberse identificado en la aplicación

Garantías en caso de éxito: La aplicación muestra los días de permiso pendientes de aprobar y registra su aprobación o denegación.

Escenario principal de éxito:

1. El usuario accede al listado de días de permiso pendientes de aprobar de los usuarios que dependen de él.
2. La aplicación muestra el listado solicitado.

Escenarios alternativos:

a. Aprobar una solicitud de días de permiso

1. El usuario aprueba la solicitud.

2. La aplicación valida que el usuario tiene permisos para aprobar la solicitud, registra el día de permiso como 'aprobado' y muestra un mensaje de confirmación al usuario.

b. Denegar una solicitud de días de permiso

1. El usuario deniega la solicitud.

2. La aplicación pide al usuario que introduzca el motivo de la denegación.

3. El usuario introduce el motivo y confirma que la quiere denegar.
4. La aplicación valida que el usuario tiene permiso para denegar la solicitud, la registra como 'denegada' y muestra un mensaje de confirmación al usuario.

Caso de uso: CU19 Identificarse en la aplicación (login)

Actor principal: *Admin*

Stakeholders e intereses:

Capacity Manager, Project Manager, Delivery Manager, Resource Manager y empleado: quiere utilizar la aplicación

Precondición: El usuario está registrado en la aplicación

Garantías en caso de éxito: El usuario inicia sesión

Escenario principal de éxito:

1. El usuario accede a la aplicación.
2. La aplicación muestra el formulario para solicitar las credenciales de acceso (email y contraseña).
3. El usuario introduce su email y contraseña y envía el formulario.
4. El sistema valida las credenciales y redirige al usuario a la página que intentaba consultar.

Escenarios alternativos:

3a. El usuario introduce una combinación usuario/contraseña inválida.

3a1. La aplicación muestra un mensaje informando que no se ha podido hacer login y volvemos al punto 2.

Caso de uso: CU20 Salir de la aplicación (logout)

Actor principal: *Admin*

Stakeholders e intereses:

Capacity Manager, Project Manager, Delivery Manager, Resource Manager y empleado: quieren finalizar la sesión en curso.

Precondición: El usuario debe haberse identificado en la aplicación

Garantías en caso de éxito: Se cierra la sesión del usuario

Escenario principal de éxito:

1. El usuario indica que quiere cerrar la sesión en curso.
2. La aplicación cierra la sesión del usuario y redirige al usuario a la página de login.

Caso de uso: CU21 Recuperar contraseña

Actor principal: *Admin*

Stakeholders e intereses:

Capacity Manager, Project Manager, Delivery Manager, Resource Manager y empleado: ha olvidado su contraseña y la quiere resetear

Garantías en caso de éxito: El usuario recibe un email que contiene un link con un tiempo de validez limitado que puede usar para resetear la contraseña de su cuenta.

Escenario principal de éxito:

1. El usuario accede a la aplicación e indica que ha olvidado su contraseña.
2. La aplicación muestra un formulario donde solicita al usuario que introduzca su email.
3. El usuario rellena el formulario y lo envía.
4. La aplicación muestra un mensaje de confirmación informando al usuario de que se le mandará un email para resetear su contraseña y valida que exista un usuario con ese email, en cuyo caso envía un correo de recuperación de contraseña al usuario.

Escenarios alternativos:

3a. El usuario deja el email vacío.

3a1. La aplicación indica que no se puede dejar el campo vacío y volvemos al punto 2.

3b. El usuario rellena el email con un formato incorrecto.

3b1. La aplicación indica el formato correcto para que el usuario pueda corregir el error y volvemos al punto 2.

4a. La aplicación no encuentra un usuario registrado con ese email y no envía ningún correo.

Caso de uso: CU22 Cambiar contraseña

Actor principal: *Admin*

Stakeholders e intereses:

Capacity Manager, Project Manager, Delivery Manager, Resource Manager y empleado: quiere cambiar su contraseña

Precondición: El usuario debe haberse identificado en la aplicación

Garantías en caso de éxito: El usuario cambia la contraseña

Escenario principal de éxito:

1. El usuario indica que quiere cambiar la contraseña de su cuenta.
2. La aplicación muestra un formulario para cambiar la contraseña de la cuenta.
3. El usuario rellena los campos del formulario (contraseña actual, nueva contraseña y confirmación de la nueva contraseña) y envía los datos.
4. La aplicación actualiza la contraseña de la cuenta del usuario y muestra un mensaje de confirmación.

Escenarios alternativos:

3a. El usuario deja alguno de los campos del formulario vacío.

3a1. La aplicación indica que el campo es obligatorio y volvemos al punto 2.

3b. El usuario rellena un campo con un valor o formato incorrecto.

3b1. La aplicación indica un mensaje de error descriptivo para que el usuario pueda corregir el error y volvemos al punto 2.

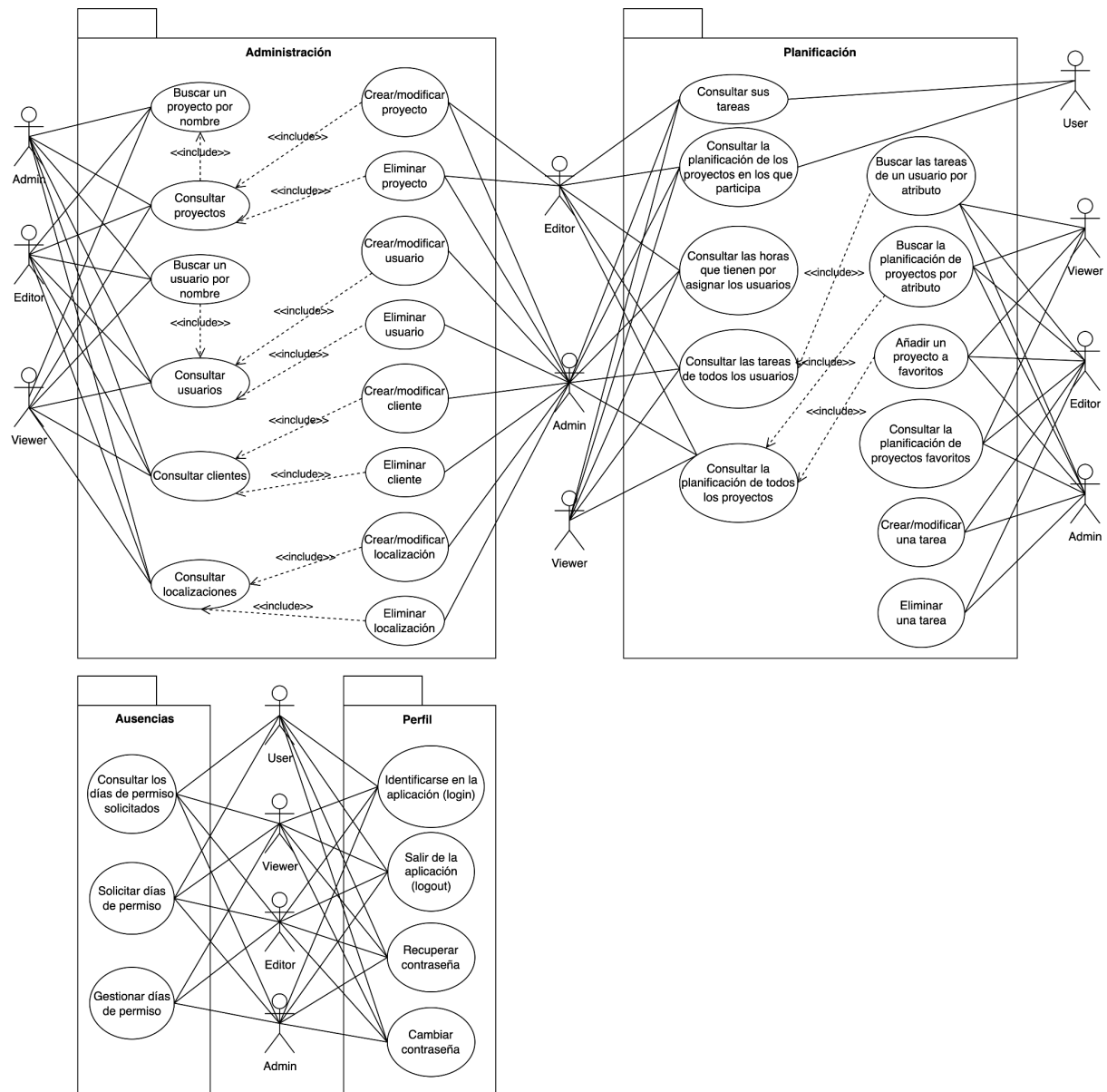
3. Análisis y diseño

3.1 Diagrama de casos de uso

Basándonos en los casos de uso identificados, y como primer paso de la fase de análisis, se ha elaborado el diagrama de casos de uso de eResourceKeeper que sirve para modelar cómo los actores interactúan con la aplicación a muy alto nivel, al relacionar visualmente los actores y los casos de uso.

Para mayor claridad, se han agrupado los casos de uso en cuatro paquetes, teniendo en cuenta la funcionalidad con la que están relacionados: Administración, Planificación, Ausencias y Perfil. Para mejorar la legibilidad cada actor aparece varias veces.

Figura 3.1
Diagrama de casos de uso



[Ver imagen a tamaño completo](#)

3.2 Modelización de la interfaz con *wireframes*

A partir de los requisitos documentados mediante casos de uso, pasamos a realizar los *wireframes*, que son una técnica muy utilizada para representar visualmente el funcionamiento de una aplicación en las fases preliminares de su desarrollo [Brown, 2010]. De este modo podremos detallar la interfaz de usuario de las diferentes páginas de la aplicación.

La elaboración de los *wireframes*, además de servir para concretar cómo se van a alcanzar los requisitos definidos en la fase anterior y ayudar a entender cómo funciona la aplicación en su conjunto, ha sido de gran utilidad para detectar y corregir errores en los requisitos. Por un lado, se han corregido requisitos que estaban mal definidos, por otro, se han detectado requisitos que se habían pasado por alto. Este es un claro ejemplo de la importancia de la retroalimentación para corregir posibles errores.

Siguiendo la metodología antes utilizada, los nuevos requisitos que se han identificado deberían incorporarse a la lista de requisitos candidatos, estimarlos y priorizarlos para saber si se encuentran entre los requisitos seleccionados, en cuyo caso deberían documentarse mediante casos de uso. Sin embargo, debido al elevado coste de dicha revisión y al carácter académico de este proyecto, descartaremos esos requisitos y nos limitaremos, únicamente, a presentar el *wireframe* correspondiente, a modo de documentación.

Los *wireframes* se han realizado para la resolución de 1920 x 1080 píxeles, una de las resoluciones de escritorio más comunes actualmente. Aunque la aplicación será *responsive*, se ha optado por dar prioridad a la versión de escritorio, dado que va a ser la de uso más frecuente. Además, hemos añadido anotaciones a los *wireframes* con el fin de describir o detallar el funcionamiento de algunos elementos.

Para revisar en detalle todos los *wireframes* y poder leer las anotaciones con claridad se recomienda visitar la siguiente página:

<https://lauramontaana168448.invisionapp.com/console/share/PQAF76XKTBV>.

A continuación, se presentan únicamente los *wireframes* de las páginas principales de la aplicación. En el [anexo C](#) se encuentra el resto de *wireframes* elaborados.

La siguiente tabla muestra los *wireframes* que corresponden a cada caso de uso. Hay casos de uso que manejan *wireframes* muy parecidos, ya que, la interfaz y la lógica es común, y lo único que cambia son los datos a mostrar. Para esos casos de uso, se ha elaborado únicamente un *wireframe*. Esto se da, por ejemplo, en los casos de uso derivados de la plantilla CU03 (eliminación de entidades) o en los de la CU04 (búsqueda de una entidad por nombre), en los que solo se ha desarrollado el *wireframe* para uno de los casos de uso del grupo. Para los casos de uso CU06, CU19, CU20 y CU21 no se ha elaborado un *wireframe* por tratarse de pantallas muy simples.

Tabla 3.1*Correspondencia de casos de uso con wireframes*

ID	Caso de uso	Wireframe
CU01.1	Consultar usuarios	WF09.1 People list WF09.2 Employee details WF09.3 Contractor details
CU01.2	Consultar proyectos	WF08.1 Projects list WF08.2 Client project details WF08.3 Internal project details
CU01.3	Consultar clientes	WF10.1 Clients list WF10.2 Client details
CU01.4	Consultar localizaciones	WF11.1 Locations lists WF11.2 Locations details
CU02.1	Crear/modificar un usuario	WF09.4 Employee edit WF09.5 Contractor edit
CU02.2	Crear/modificar un proyecto	WF08.4 Client project edit WF08.5 Internal project edit WF08.6 Create project
CU02.3	Crear/modificar un cliente	WF10.3 Client edit
CU02.4	Crear/modificar una localización	WF11.3 Location edit
CU03.1	Eliminar un usuario	WF08.7 Delete project WF10.4 Client delete
CU03.2	Eliminar un proyecto	WF08.7 Delete project WF10.4 Cliente delete
CU03.3	Eliminar un cliente	WF08.7 Delete project WF10.4 Cliente delete
CU03.4	Eliminar una localización	WF08.7 Delete project WF10.4 Cliente delete
CU04.1	Buscar un usuario por nombre	WF08.8 Search project by name WF08.9 Search project no results
CU04.2	Buscar un proyecto por nombre	WF08.8 Search project by name WF08.9 Search project no results
CU05	Crear/modificar una tarea	WF03.4 Add new task WF03.5 Edit task
CU06	Eliminar una tarea	No desarrollado
CU07	Consultar sus tareas	WF05 My schedule
CU08	Consultar las tareas de todos los usuarios	WF03.1 People's schedule WF03.2 People's schedule expanded
CU09	Buscar las tareas de los usuarios por atributo	WF03.3 People's schedule search
CU10	Consultar las horas que tienen por asignar los usuarios	WF04.1 Bench
CU11	Consultar la planificación de los proyectos en los que participa	WF02 My projects
CU12	Consultar la planificación de todos los proyectos	WF01.1 Projects plan WF01.2 Projects plan expanded
CU13	Buscar la planificación de proyectos por atributo	WF01.3 Projects plan search
CU14	Añadir un proyecto a favoritos	WF01.1 Projects plan WF01.2 Projects plan expanded
CU15	Consultar la planificación de proyectos favoritos	WF01.4 Projects plan favourites
CU16	Consultar los días de permiso solicitados	WF07.1 My time away
CU17	Solicitar días de permiso	WF07.2 Create time away request
CU18	Gestionar días de permiso	WF06 Team time away
CU19	Identificarse en la aplicación (login)	No desarrollado
CU20	Salir de la aplicación (logout)	No desarrollado
CU21	Recuperar contraseña	No desarrollado
CU22	Cambiar contraseña	WF12 Update password

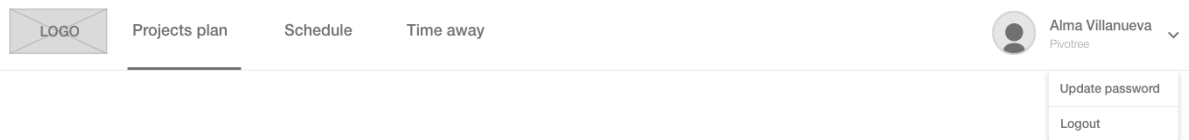
Sistemas de navegación

Uno de los propósitos principales de un *wireframe* es definir los sistemas de navegación de un sitio web. A continuación se muestra la cabecera de la aplicación que contiene, por un lado, el menú de navegación principal y, por otro, el menú de la cuenta del usuario. Existen dos versiones de la cabecera: una para los usuarios de con rol *User* y otra para el resto de usuarios.

El menú principal de un usuario no privilegiado (rol *User*) tiene únicamente 3 enlaces: la página con planificación de los proyectos en los que participa (*Projects plan*), la página con planificación de sus tareas (*Schedule*) y la página de permisos (*Time away*), desde la cual puede crear una nueva solicitud o consultar el estado de las solicitudes pendientes.

Figura 3.2

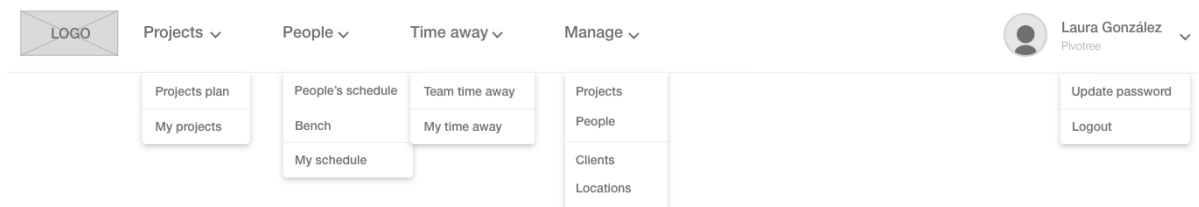
Menú de navegación para usuarios con rol *User*



Por su parte, los usuarios privilegiados (roles *Viewer*, *Editor* y *Admin*) tienen un menú principal más complejo que, para facilitar su uso, se ha organizado en 4 secciones:

Figura 3.3

Menú de navegación para usuarios con rol *Viewer*, *Editor* y *Admin*



- **Projects.** Contiene los enlaces a las páginas que muestran la planificación de proyectos:
 - [Projects plan](#): la página con la planificación de todos los proyectos
 - [My projects](#): la página con planificación de los proyectos en los que participa el usuario que hace la consulta.
- **People.** Contiene enlaces a las páginas que muestran información sobre la asignación de tareas de los usuarios:
 - [People's schedule](#): la página con la planificación de tareas de todos los usuarios
 - [Bench](#): la página con los usuarios que tienen horas libres
 - [My schedule](#): la página con la planificación de tareas del usuario que hace la consulta.
- **Time away.** Contiene enlaces a las páginas relacionadas con la solicitud de permisos:
 - [Team time away](#): la página desde la que un mánager gestiona los permisos solicitados por los miembros de su equipo
 - [My time away](#): la página desde la que el usuario que hace la consulta ve sus propios permisos.

- **Manage.** Contiene enlaces a las páginas desde las que se gestionan las entidades que son administrables:
 - **Projects:** página con el listado de proyectos. Desde el listado se puede acceder al detalle o al formulario de edición y dar de alta un proyecto.
 - **People:** página con el listado de usuarios. Desde el listado se puede acceder al detalle o al formulario de edición y dar de alta un usuario.
 - **Clients:** página con el listado de clientes. Desde el listado se puede acceder al detalle o al formulario de edición y dar de alta un cliente.
 - **Locations:** página con el listado de localizaciones. Desde el listado se puede acceder al detalle o al formulario de edición y dar de alta una localización.

Projects plan - Página con la planificación de todos los proyectos

Figura 3.4
Planificación de los proyectos

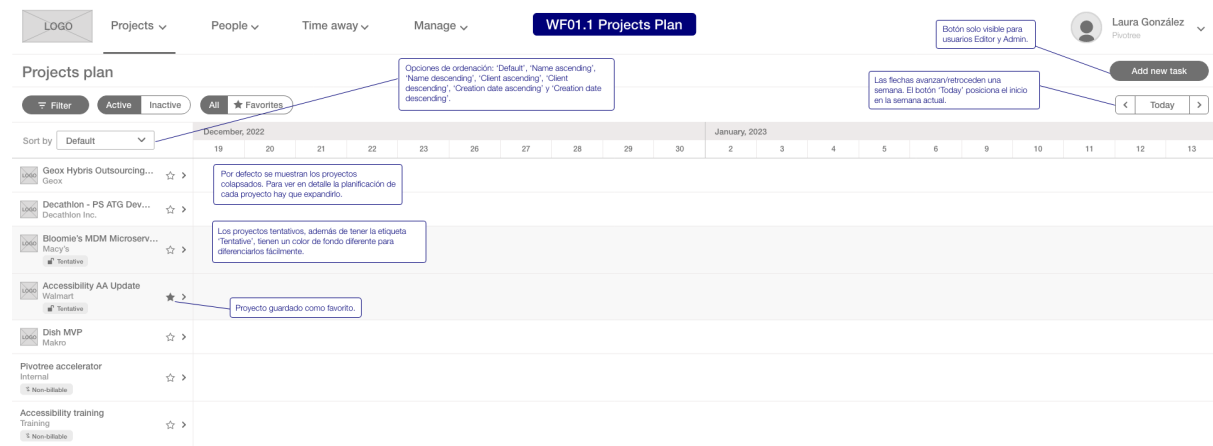


Figura 3.5
Planificación de los proyectos con la información de un proyecto expandida

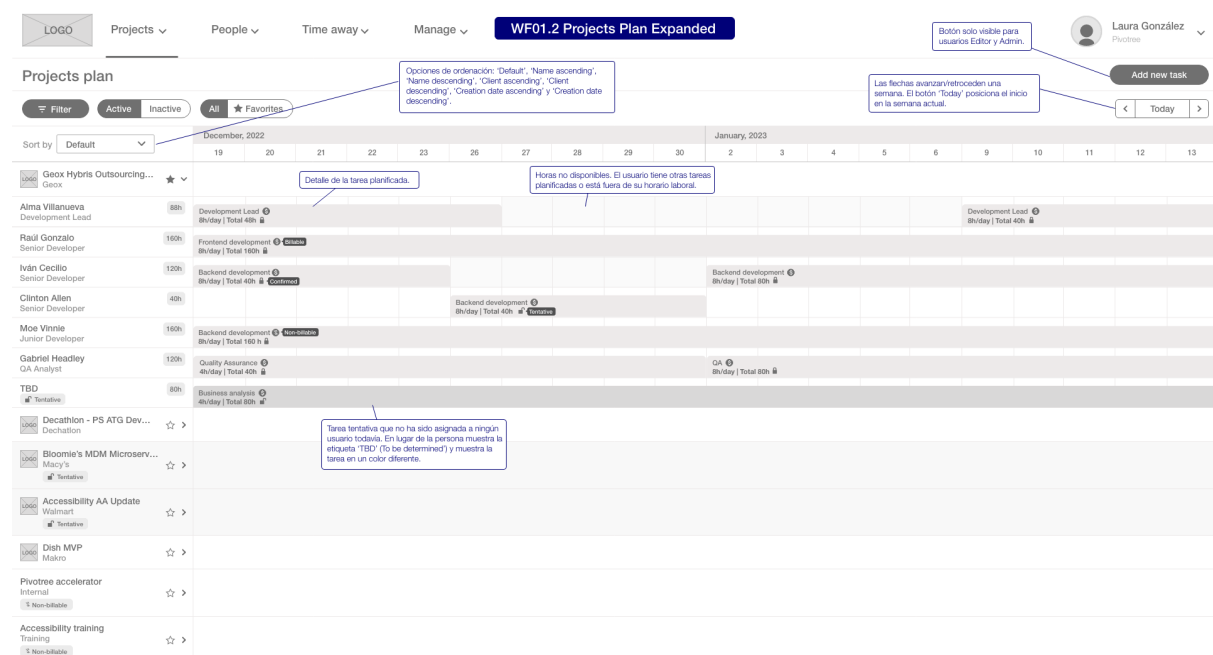


Figura 3.6
Planificación de los proyectos guardados como favoritos

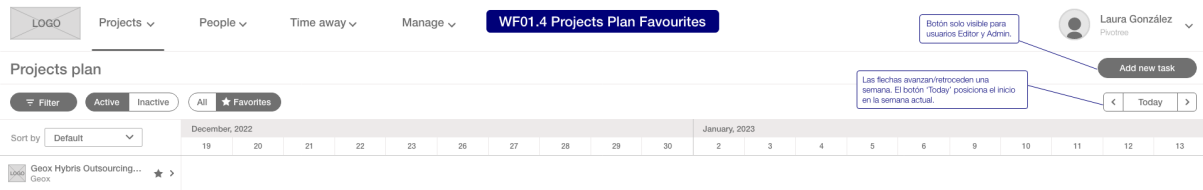
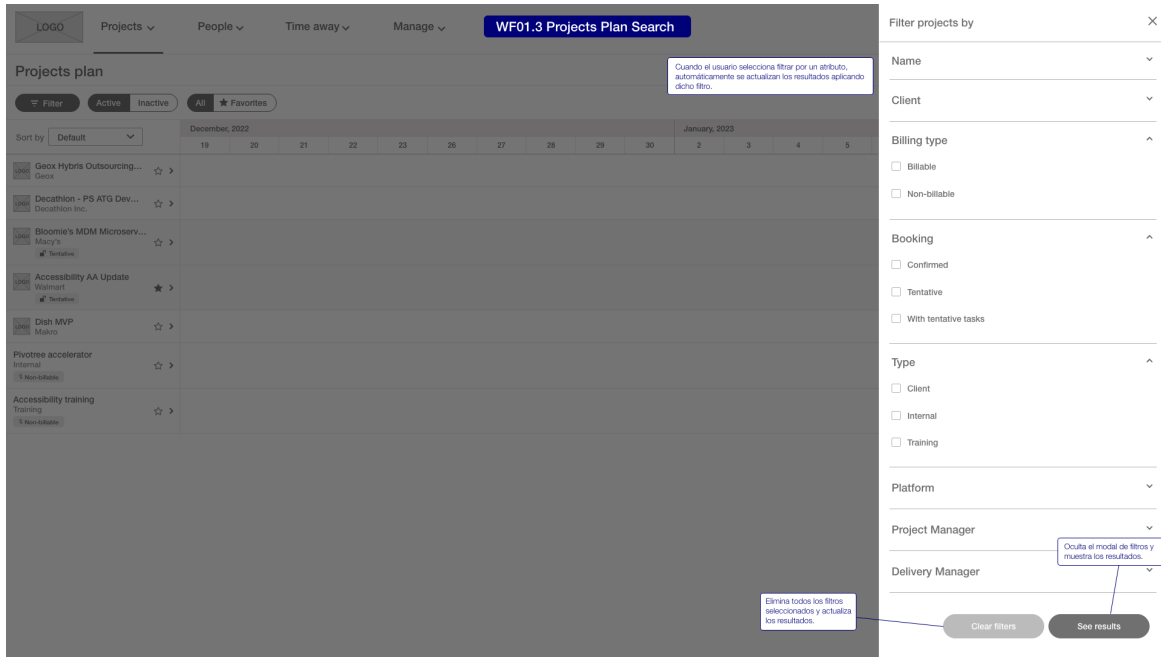


Figura 3.7
Filtros de búsqueda de la planificación de los proyectos



People's schedule - Página con la planificación de tareas de todos los usuarios

Figura 3.8
Planificación de las tareas de todos los usuarios

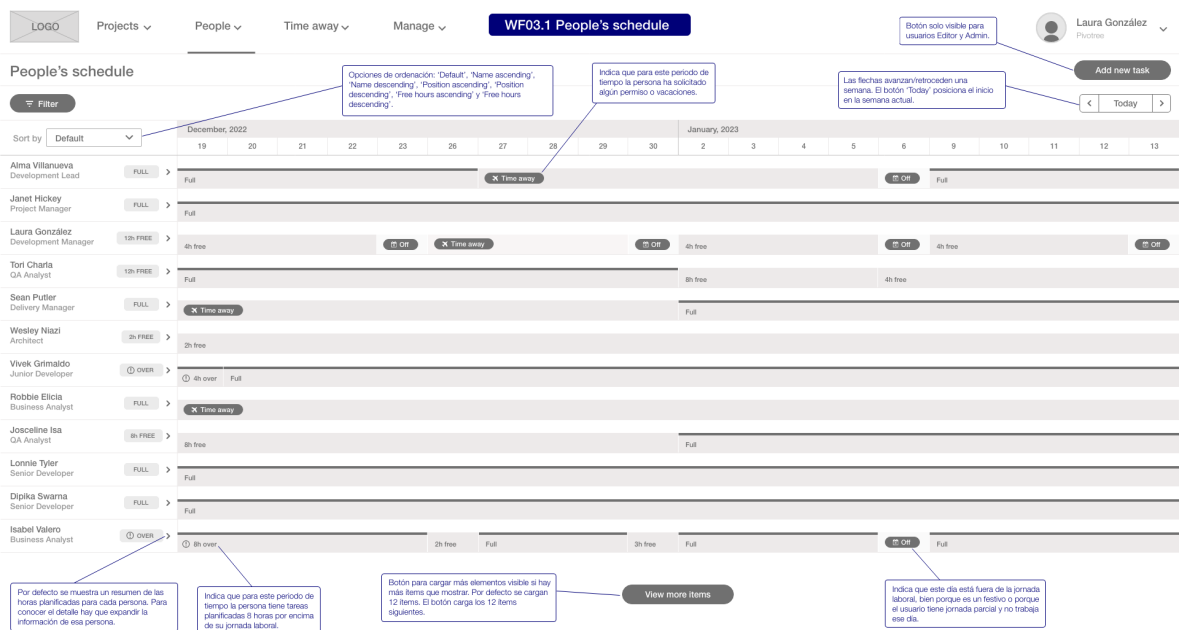


Figura 3.9
Planificación de las tareas con la información expandida

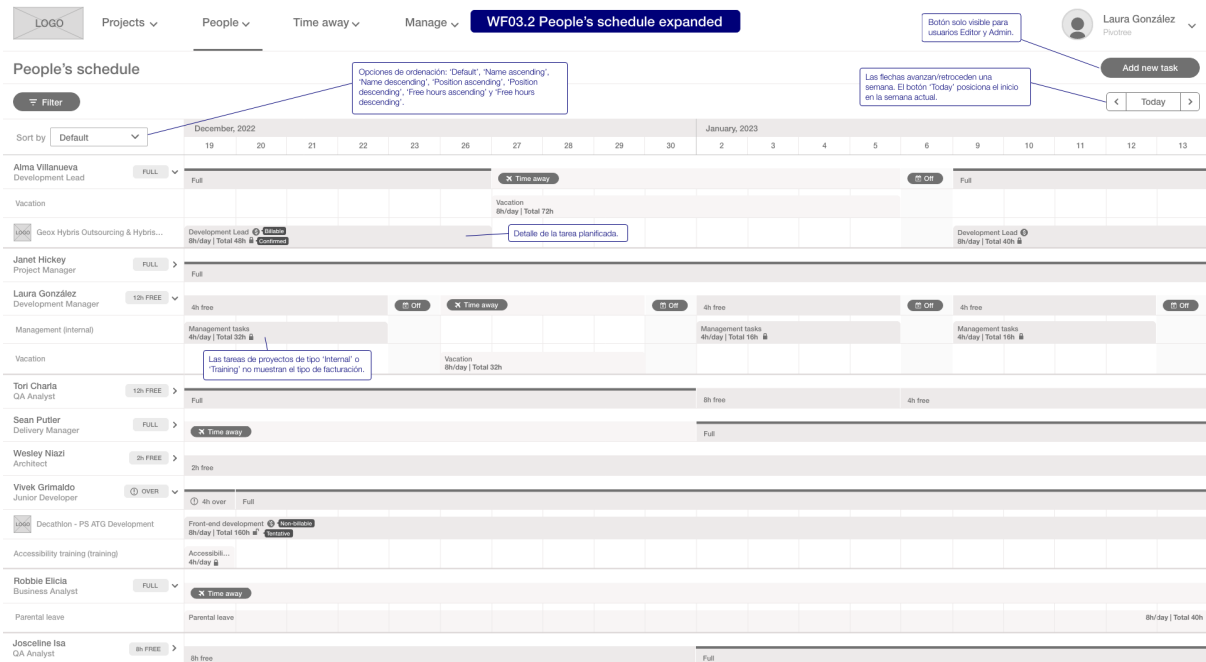
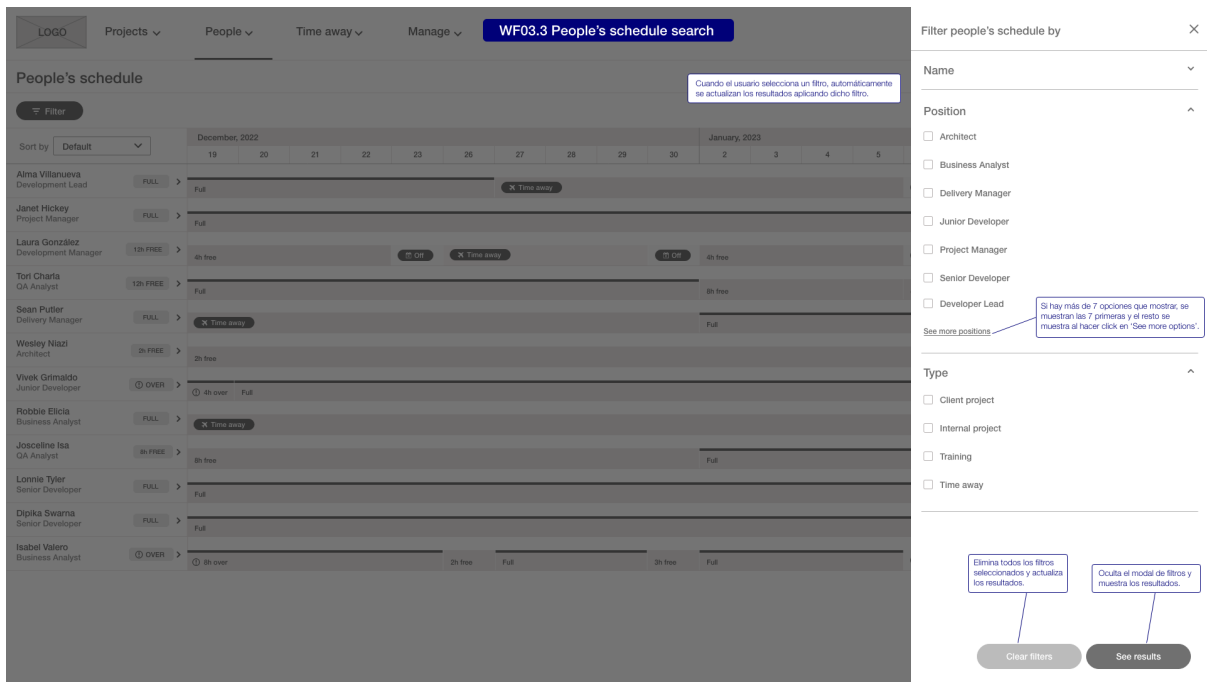


Figura 3.10
Filtros de búsqueda de la planificación de las tareas



Gestión de tareas - Modal para añadir o editar una tarea

Las tareas se pueden crear desde las páginas *Projects plan* y *People's schedule*. El botón "Add new task" muestra el formulario para crear una tarea en un modal. El formulario "Edit task" se abre al hacer click en la tarea, desde cualquiera de las vistas en las que se muestra. Desde el formulario de edición se puede eliminar la tarea.

Figura 3.11
Crear una tarea

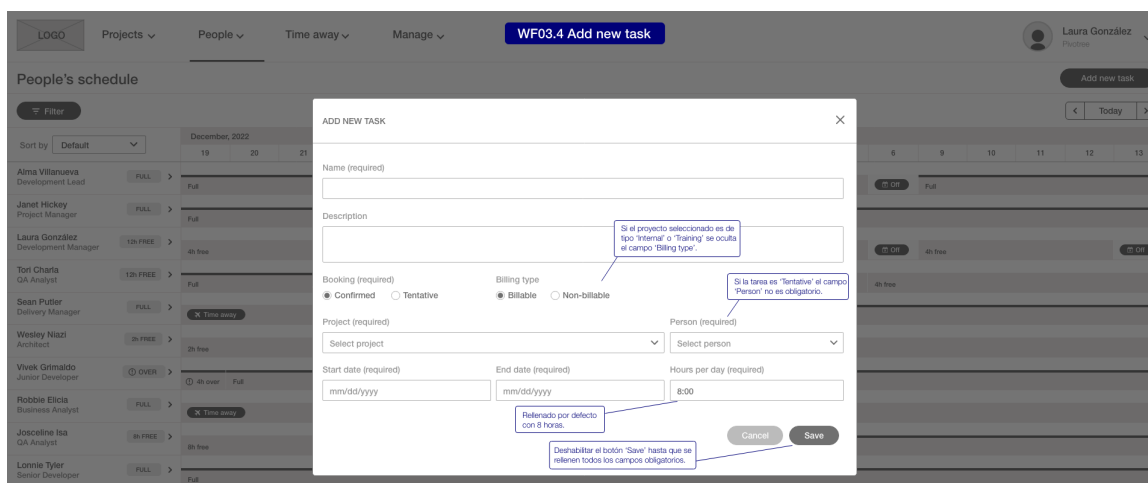
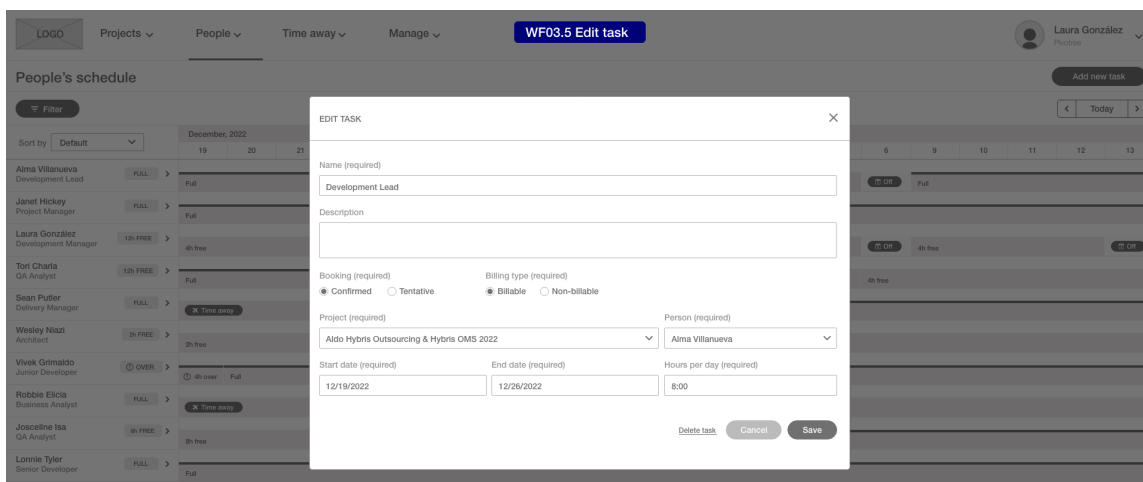


Figura 3.12
Modificar una tarea



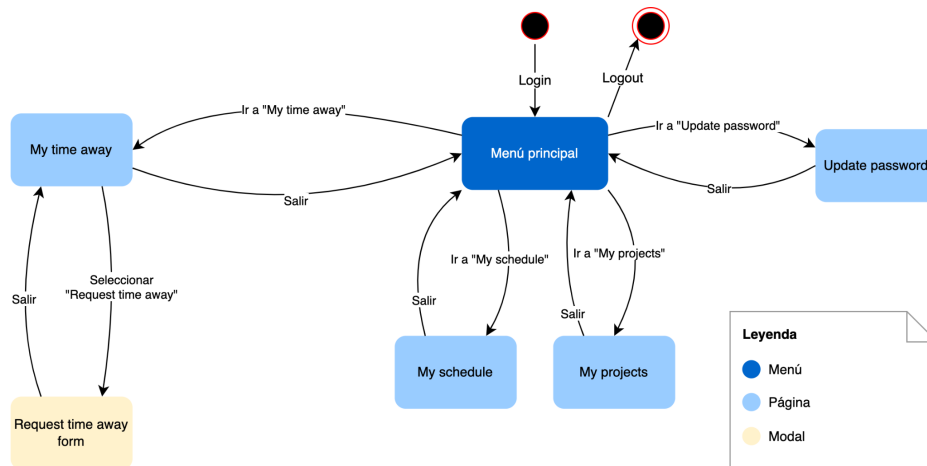
3.3 Mapa navegacional

Basándonos en los *wireframes* elaborados, se ha desarrollado el siguiente mapa navegacional con el fin de definir el orden y la relación que existe entre las diferentes pantallas que se han diseñado.

Para elaborar este modelo hemos utilizado un diagrama de estados, siguiendo las indicaciones de Horrocks [1999], donde cada pantalla de la aplicación se define como un estado y los eventos que causan el cambio de estado son eventos de usuario (click en un enlace, pulsar un botón, ...). Además de modelar qué eventos llevan de una pantalla a otra, hemos añadido condiciones para especificar qué roles de usuario tienen permisos para acceder en cada caso.

El siguiente diagrama muestra el mapa navegacional de la aplicación para un usuario no privilegiado (rol *User*). El estado inicial es el menú principal, desde el cual el usuario puede ir a diferentes páginas. Desde algunas de estas páginas el usuario puede a su vez ir a otra pantalla/modal.

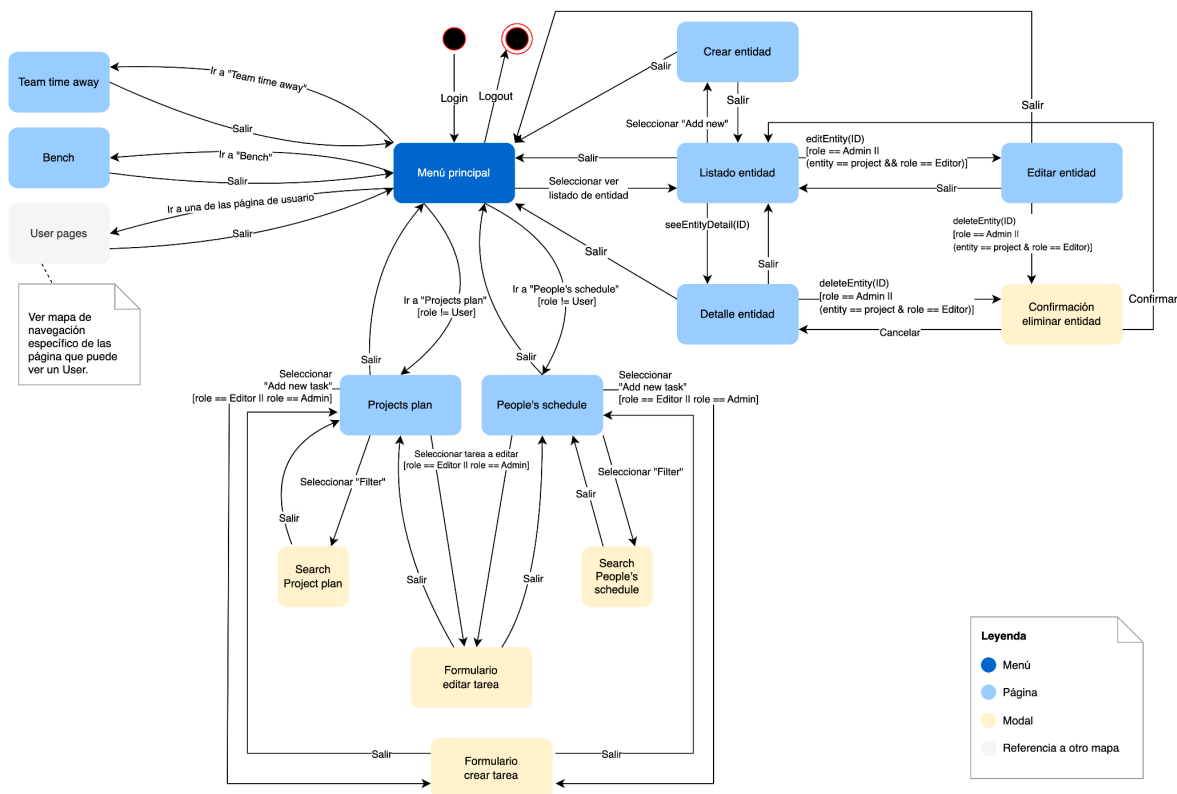
Figura 3.13
 Mapa navegacional de las diferentes pantallas para los usuarios User



[Ver imagen a tamaño completo](#)

El siguiente diagrama muestra el mapa navegacional de la aplicación para los usuarios privilegiados (roles *Viewer*, *Editor* y *Admin*). Con el fin de mejorar la legibilidad las páginas a las que también puede acceder un usuario no privilegiado, detalladas en el diagrama anterior, se han omitido y se representan con el nodo 'User pages'. Las páginas de gestión de entidades (proyectos, usuarios, cliente y localizaciones), aparecen modeladas de forma conjunta, dado que el flujo es idéntico para todas ellas.

Figura 3.14
 Mapa navegacional de las diferentes pantallas para usuarios Viewer, Editor y Admin



[Ver imagen a tamaño completo](#)

3.4 Análisis de solicitud de días de permiso

A continuación se muestra un diagrama de máquina de estados para definir y visualizar los estados por los que pasa una solicitud de días de permiso una vez es creada por el usuario. Este diagrama sirve, además, para ilustrar qué eventos pueden cambiar el estado de una solicitud. Los eventos pueden ser de dos tipos:

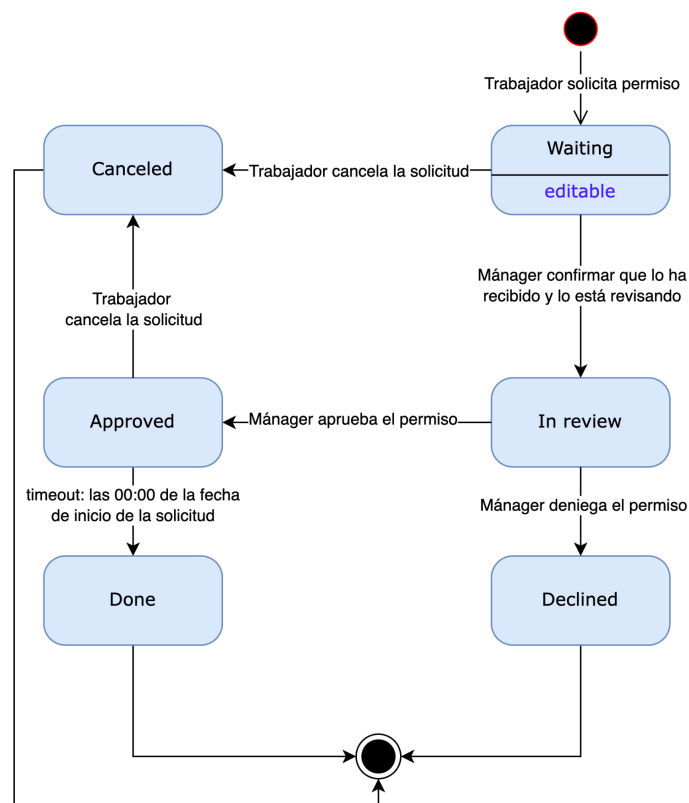
- Eventos manuales: el cambio de estado se produce mediante una acción del usuario.
- Eventos automáticos: el cambio de estado se produce al llegar una fecha concreta.

El círculo con un punto en el interior indica el fin del proceso. Como puede verse en el diagrama, una solicitud puede terminar en los estados *Declined*, *Done* y *Cancelled*.

Algunos estados muestran el atributo 'editable' para indicar que, en ese estado, la solicitud es modificable por el usuario.

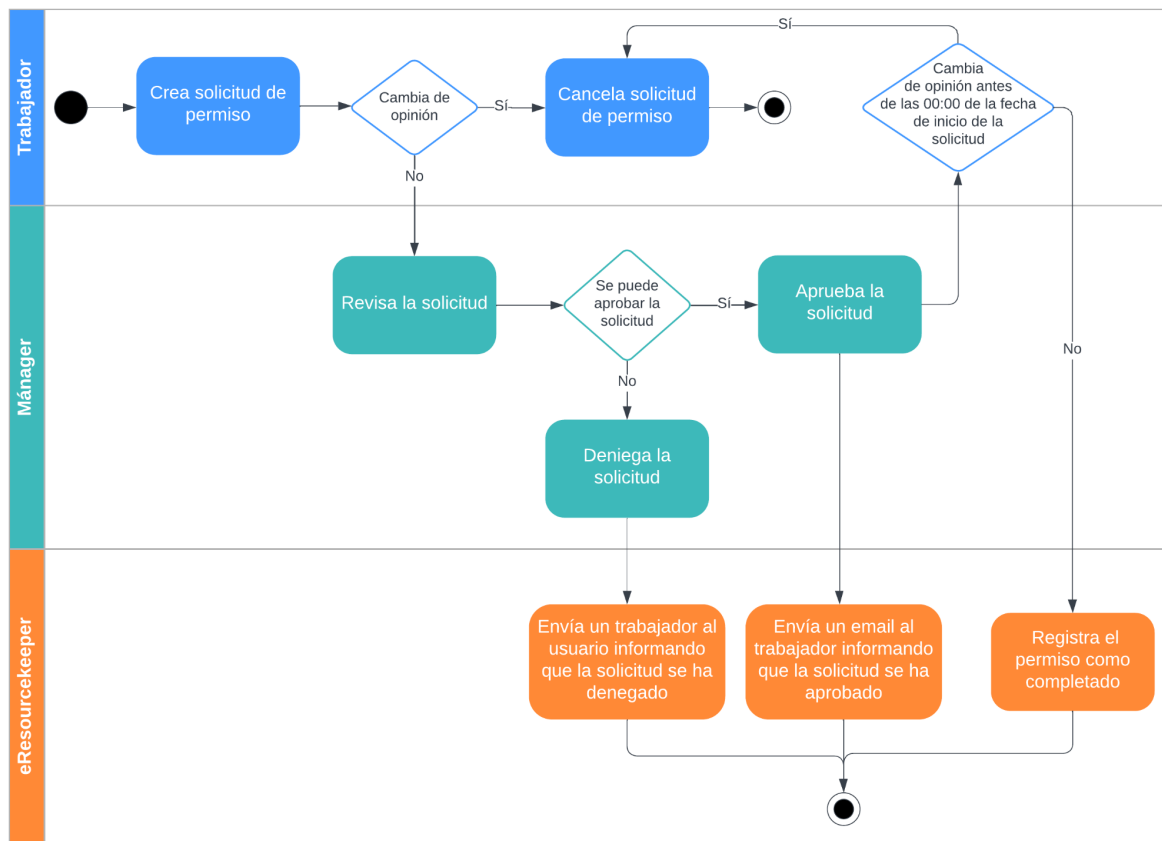
Figura 3.15

Diagrama de máquina de estados de una solicitud de días de permiso



El siguiente diagrama de actividades con carriles (*swimlanes*) indica qué actividades hace cada uno de los actores del proceso.

Figura 3.16
 Diagrama de actividades de una solicitud de días de permiso

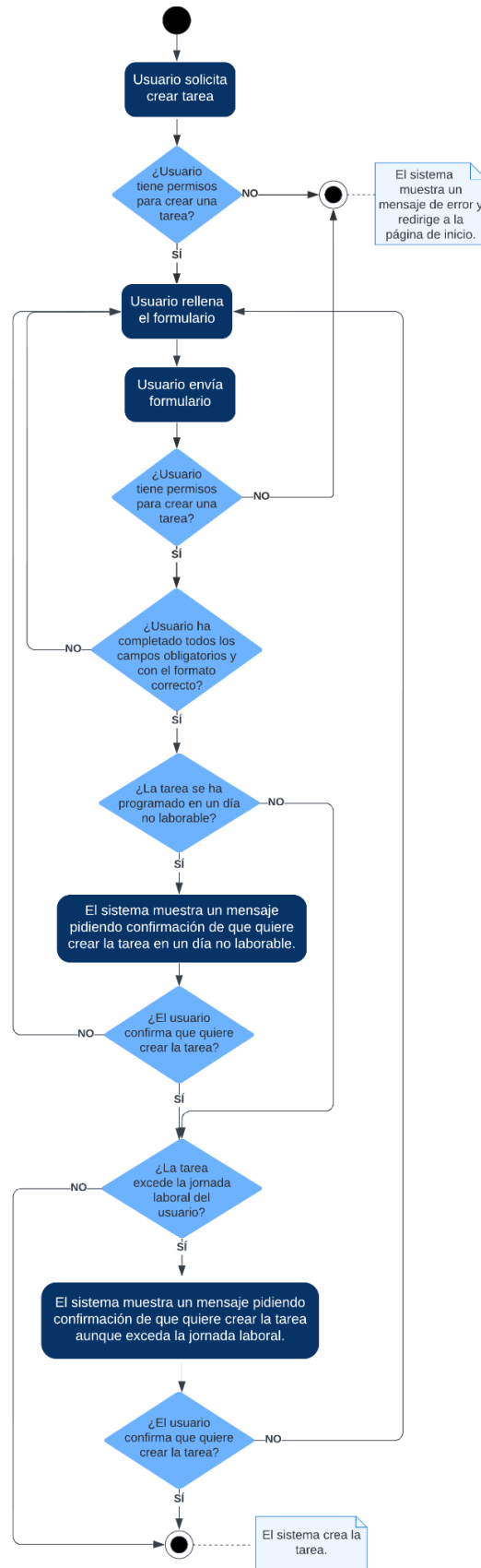


3.5 Análisis de creación de una tarea

A continuación se muestra el diagrama de actividades para el caso de uso 'CU02.3 Crear una tarea', con el fin facilitar la comprensión del proceso. Este tipo de diagrama nos permite ilustrar el flujo de trabajo entre los usuarios y la aplicación y describir con mayor claridad los pasos documentados en el caso de uso.

Figura 3.17

Diagrama de actividades con los pasos para crear una tarea

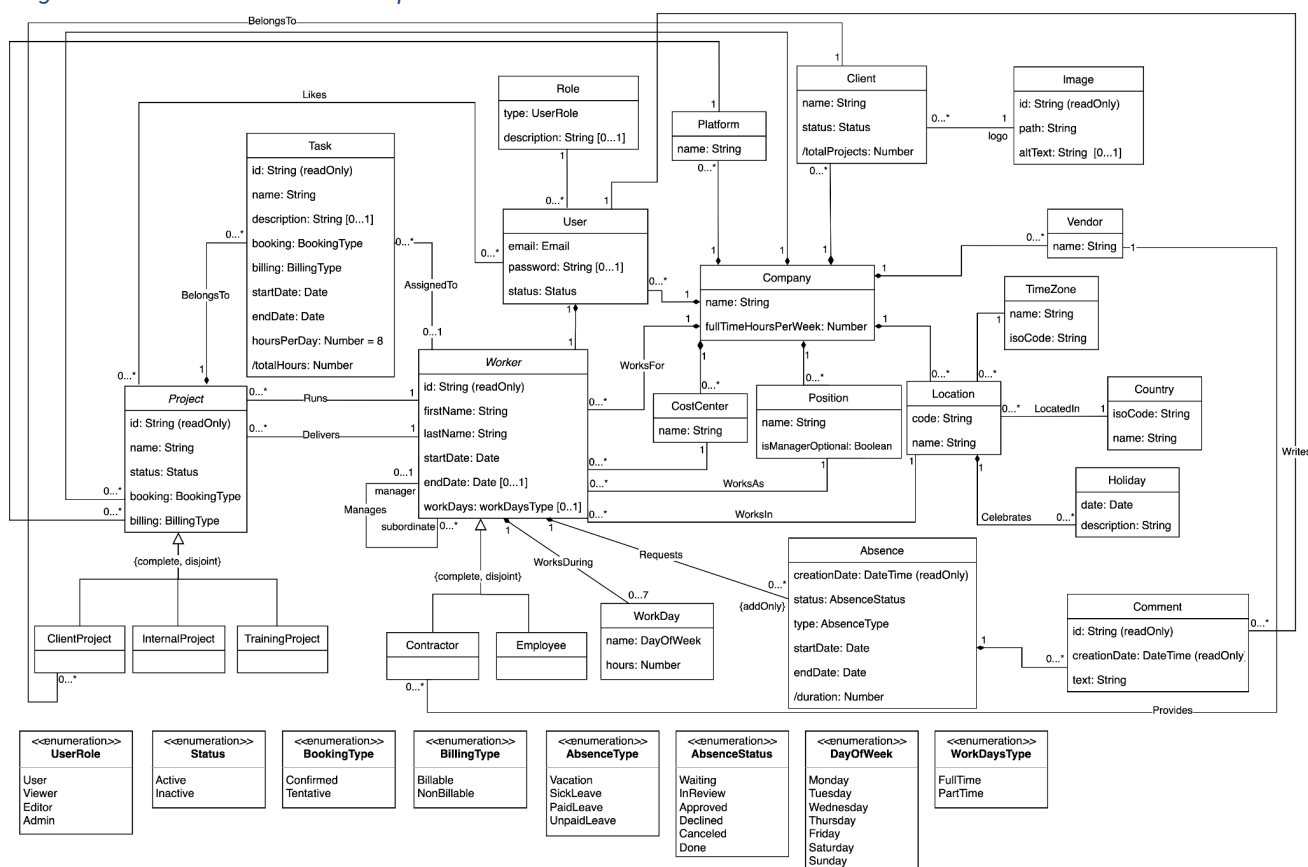


[Ver imagen a tamaño completo](#)

3.6 Modelo conceptual de clases

En este apartado describiremos mediante un diagrama de clases UML [Booch et al., 2005] la información que va a tratar eResourceKeeper, incluyendo los tipos de objetos que encapsulan dicha información, las relaciones existentes entre ellos y las posibles restricciones.

Figura 3.18
Diagrama de clases de eResourceKeeper



[Ver imagen a tamaño completo](#)

3.6.1 Reglas de integridad

Dado que muchas reglas de integridad no pueden ser representadas en el diagrama de clases las expresaremos textualmente.

Claves de las clases del dominio

Cada clase del dominio tendrá un atributo o conjunto de atributos que identifica las instancias de manera única. Por lo tanto, la aplicación debe garantizar que no existan dos instancias de una misma clase con idéntica clave. A continuación detallamos el atributo, o conjunto de atributos, que actuará como clave para cada clase:

- *Company: name*
- *Project: id*
- *Task: id*
- *Worker: id*

- *User: email*
- *Role: roleType*
- *Client: name + Company*
- *Image: id*
- *Platform: name + Company*
- *Vendor: name + Company*
- *Position: name + Company*
- *CostCenter: name + Company*
- *Location: code + Company*
- *TimeZone: name*
- *Country: isoCode*
- *Holiday: date + Location*
- *WorkDay: name + Worker*
- *Absence: creationDate + Worker*
- *Comment: id*

Otras restricciones de integridad

Aparte de las restricciones de clave, hay otras restricciones de integridad que UML no nos permite expresar gráficamente y que se detallan a continuación:

- Las instancias de la clase *Worker* relacionadas mediante la asociación *manages* deben formar una jerarquía con una estructura de árbol. No se pueden formar ciclos.
- El atributo *endDate* de clase *Task* debe ser posterior al atributo *startDate*.
- El atributo *startDate* de clase *Absence* debe ser posterior a la fecha actual.
- El atributo *endDate* de clase *Absence* debe ser posterior al atributo *startDate*.
- No se puede crear una instancia de la clase *Absence* con unos valores de *startDate* y *endDate* que se solapen, aunque sea parcialmente, con los de otra instancia de *Absence* preexistente para una misma instancia de *Worker*, salvo que la instancia preexistente se encuentre en estado *Declined* o *Canceled*.
- El atributo *hours* de clase *WorkDay* no puede ser mayor que 24.
- No se puede eliminar una instancia de *Vendor* si existen instancias de la clase *Contractor* asociadas.
- No se puede eliminar una instancia de *CostCenter* si existen instancias de la clase *Worker* asociadas.
- No se puede eliminar una instancia de *Position* si existen instancias de la clase *Worker* asociadas.
- No se puede eliminar una instancia de *Location* si existen instancias de la clase *Worker* asociadas.
- No se puede eliminar una instancia de *Platform* si existen instancias de la clase *Project* asociadas.
- No se puede eliminar una instancia de *Client* si existen instancias de la clase *ClientProject* asociadas.
- Tiene que haber como mínimo una instancia de *User* con rol *Admin*. Un *User* con rol *Admin* no se podrá eliminar si no hay al menos otro *User* con rol *Admin* en la aplicación.

- Las instancias de *Worker* solo se pueden eliminar si el status es '*Inactive*', no tienen gente a su cargo (relación *manages*) y no tienen la asociación *delivers* y *manages*.
- Las instancias de *Project* solo se pueden eliminar si el status es '*Inactive*'.

3.6.2 Información derivada

Igual que ocurre con las restricciones de integridad, UML no ofrece una manera gráfica de expresar cómo se obtiene la información derivada. Por ello, lo documentamos textualmente a continuación:

- *Task::totalHours* es el número total de horas para las que se ha programado esa tarea. Se obtiene haciendo el producto de *hoursPerDay* por el número de días para los que se ha programado la tarea.
- *Client::totalProjects* es el número de proyectos que tiene asociados un cliente.
- *Absence::duration* es el número de días hábiles que dura una ausencia.

3.7 Diseño de la base de datos

En este apartado se llevará a cabo el diseño conceptual y lógico de la base de datos. El diseño físico queda fuera del alcance de este proyecto, ya que depende del SGBD concreto que se vaya a usar.

3.7.1 Diseño conceptual

La etapa de diseño conceptual de la base de datos consiste en crear un esquema conceptual de alto nivel independiente de la tecnología, a partir del análisis de requisitos.

Existen varios modelos de datos de alto nivel independientes de la tecnología. Uno de los más conocidos y utilizados para el diseño conceptual de bases de datos es el modelo entidad-relación [Chen, 1976]. El modelo de clases UML es otra alternativa igualmente válida, dado que nos permite representar la información del dominio y sus restricciones. Por ello, consideramos que el diseño conceptual de la base de datos ya ha quedado especificado por el modelo conceptual de clase del apartado 3.6.

3.7.2 Diseño lógico

La etapa de diseño lógico consiste en obtener un esquema lógico a partir del esquema conceptual generado en la etapa anterior. El esquema lógico dependerá del tipo de base de datos que escojamos, aunque será independiente del SGBD concreto que se vaya a utilizar.

Nosotros optaremos por usar una base de datos según el modelo relacional [Codd, 1970], ya que es muy flexible, se adapta perfectamente a nuestras necesidades y su popularidad hace que existan sistemas, tanto propietarios (p. ej. Oracle) como libres (p. ej. MySQL), con amplia implantación y por tanto muy probados y con una gran base de conocimiento. Así pues, en esta etapa transformaremos el esquema conceptual, antes expresado mediante un diagrama de clases UML, en un modelo lógico adecuado para un tipo de base de datos

relacional. Del modelo lógico resultante obtendremos un esquema lógico relacional³, también conocido simplemente como esquema relacional.

La manera clásica de representar esquemas relacionales es mediante el lenguaje SQL [Date y Darwen, 1997], un estándar que permite trabajar con bases de datos relacionales. Por claridad y concisión, utilizaremos una notación simplificada del lenguaje SQL estándar, que detallamos a continuación:

- Identificaremos cada relación mediante un nombre único, que irá seguido por la lista de atributos entre paréntesis separados por comas.
- Subrayaremos el atributo que forma la clave primaria.
- Utilizaremos negrita para señalar los atributos que vamos a declarar NOT NULL.
- Expresaremos cada clave alternativa mediante una línea de texto indentada adicional, indicando qué atributos la forman (puede ser uno o varios).
- Expresaremos cada clave foránea mediante una línea de texto indentada adicional, indicando qué atributos la forman y a qué relación referencia.

A continuación resumimos la estrategia utilizada para definir las relaciones del modelo lógico, teniendo en cuenta su cardinalidad y tipo de la relación:

- Cardinalidad de 1 a 1: las representaremos mediante una clave foránea en alguna de las entidades de la relación
- Cardinalidad de 1 a 0..*: las representaremos mediante una clave foránea en la entidad con cardinalidad *
- Cardinalidad de 0..* a 0..1: crearemos una nueva entidad incluyendo las claves foráneas de ambas entidades para evitar la presencia de valores nulos (NULL)
- Cardinalidad de 0..* a 0..*: crearemos una nueva entidad incluyendo las claves foráneas de ambas entidades
- Relaciones de composición: las representaremos mediante una clave foránea en la entidad dependiente.

Hemos tomado la decisión de usar como clave primaria de todas las relaciones un identificador que será un entero por motivos de eficiencia en las operaciones de consulta a la base de datos y con el fin de contar con una clave primaria que sea independiente de los datos que maneja la aplicación.

A continuación se detalla el modelo lógico relacional de la base de datos:

```
Role(id, type, description)
    {type} is an alternate key
```

```
Company(id, name, fullTimeHoursPerWeek)
    {name} is an alternate key
```

³ En este contexto utilizaremos la palabra relación para referirnos al elemento básico del modelo lógico relacional. Esto no debe confundirse con las relaciones a las que nos hemos referido al realizar el diseño conceptual, que eran instancias de tipos de relación entre clases.

Client(id, name, status, logo_id, company_id)
 {name, company_id} is an alternate key
 {logo_id} is foreign key to Image
 {company_id} is foreign key to Company

Image(id, path, altText)

Platform(id, name, company_id)
 {name, company_id} is an alternate key
 {company_id} is foreign key to Company

CostCenter(id, name, company_id)
 {name, company_id} is an alternate key
 {company_id} is foreign key to Company

Position(id, name, company_id)
 {name, company_id} is an alternate key
 {company_id} is foreign key to Company

Vendor(id, name, company_id)
 {name, company_id} is an alternate key
 {company_id} is foreign key to Company

Location(id, code, name, timeZone_id, country_id, company_id)
 {code, company_id} is an alternate key
 {name, company_id} is an alternate key
 {timeZone_id} is foreign key to TimeZone
 {country_id} is foreign key to Country
 {company_id} is foreign key to Company

Country(id, isoCode, name)
 {isoCode} is an alternate key

TimeZone(id, name, isoCode)
 {name} is an alternate key

Holiday(id, date, description, location_id)
 {date, location_id} is an alternate key
 {location_id} is foreign key to Location

Project(id, name, status, booking, billing, projectManager_id, deliveryManager_id, company_id)
 {name, company_id} is an alternate key
 {projectManager_id} is foreign key to Worker
 {deliveryManager_id} is foreign key to Worker
 {company_id} is foreign key to Company

ClientProject(id, client_id)
 {id} is foreign key to Project
 {client_id} is foreign key to Client

InternalProject(id)
 {id} is foreign key to Project

TrainingProject(id)
 {id} is foreign key to Project

ProjectPlatform(id, project_id, platform_id)
 {project_id, platform_id} is an alternate key

{project_id} is foreign key to Project
{platform_id} is foreign key to Platform

Task(id, name, description, **booking**, **billing**, startDate, endDate, hoursPerDay, project_id)
{project_id} is foreign key to Project

User(id, email, password, status, role_id, company_id)
{email} is an alternate key
{role_id} is foreign key to Role
{company_id} is foreign key to Company

Worker(id, firstName, lastName, startDate, endDate, workDays, costCenter_id, position_id, user_id, location_id, company_id)
{costCenter_id} is foreign key to CostCenter
{position_id} is foreign key to Position
{user_id} is foreign key to User
{location_id} is foreign key to Location
{company_id} is foreign key to Company

Contractor(id, vendor)
{id} is foreign key to Worker
{vendor} is foreign key to Vendor

Employee(id)
{id} is foreign key to Worker

WorkerTask(id, worker_id, task_id)
{worker_id, task_id} is an alternate key
{worker_id} is foreign key to Worker
{task_id} is foreign key to Task

WorkDay(id, name, hours, worker_id)
{name, worker_id} is an alternate key
{worker_id} is foreign key to Worker

Absence(id, creationDate, status, type, startDate, endDate, worker_id)
{creationDate, worker_id} is an alternate key
{worker_id} is foreign key to Worker

Comment(id, creationDate, text, absence_id, author_id)
{absence_id} is foreign key to Absence
{author_id} is foreign key to User

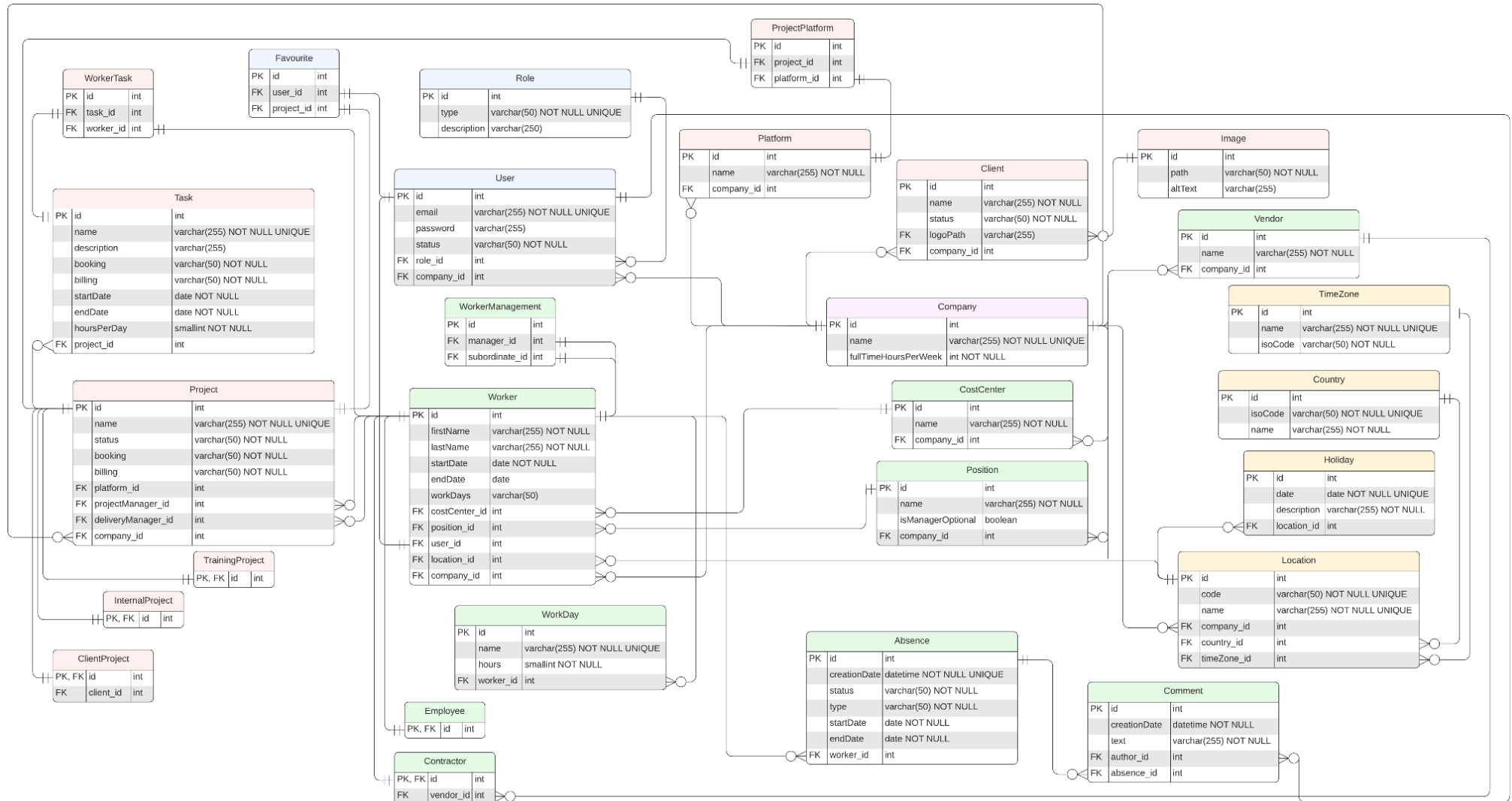
WorkerManagement(id, manager_id, subordinate_id)
{manager_id, subordinate_id} is an alternate key
{manager_id} is foreign key to Worker
{subordinate_id} is foreign key to Worker

Favourite(id, user_id, project_id)
{user_id, project_id} is an alternate key
{user_id} is foreign key to User
{project_id} is foreign key to Project

Complementamos el modelo lógico relacional anterior con un diagrama entidad-relación físico que muestra la estructura de todas las tablas, incluidos nombre de cada columna, tipo de dato, restricciones, clave primaria, clave foránea y relación con otras tablas.

Figura 3.19

Diagrama del esquema relacional de eResourceKeeper



[Ver imagen a tamaño completo.](#)

3.8 Arquitectura

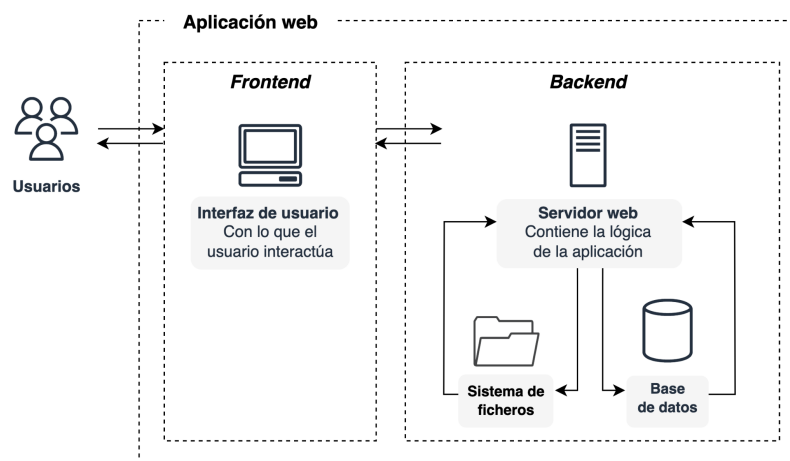
Tras el análisis de los requisitos de eResourceKeeper pasamos a definir la arquitectura de la aplicación, donde describiremos a muy alto nivel sus principales componentes y cómo interactúan entre sí.

Los requisitos no funcionales operacionales y de entorno (RNF04.1 y RNF04.2), establecen que eResourceKeeper será una aplicación web. El uso de una aplicación web nos garantiza el fácil acceso universal, ya que se podrá acceder desde cualquier dispositivo sin necesidad de instalar ningún software.

Como se muestra en la figura 3.20, dicha aplicación web consta de dos partes bien diferenciadas:

- *Frontend*: parte de la aplicación que se ejecuta en el lado del cliente (navegador web) y contiene la interfaz gráfica con la que interactúa el usuario a través de elementos como botones, enlaces, menús de navegación, formularios, etc.
- *Backend*: parte de la aplicación que se ejecuta en el servidor y contiene la lógica de la aplicación, así como la persistencia de los datos.

Figura 3.20
Diagrama de la aplicación web

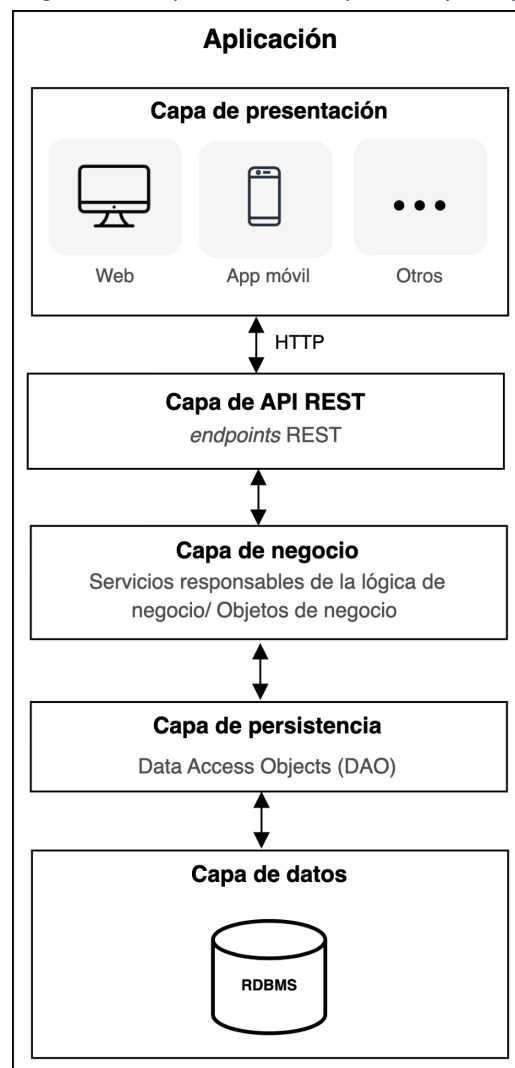


La arquitectura que mejor se adapta a este modelo de aplicación web es una arquitectura cliente-servidor organizada en capas [Richards, 2015]. El modelo en capas, uno de los más comunes en este tipo de desarrollos, consiste en dividir la aplicación en diferentes capas lógicas, de forma que cada una tiene una función y responsabilidad específicas. Esta división permite abordar el problema de forma modular, con lo que se llega a la solución de una manera más sencilla y flexible.

La figura 3.21, muestra las diferentes capas en las que está dividida eResourceKeeper. La función de cada una se detalla en la siguiente tabla:

Tabla 3.2*Correspondencia de capas de la aplicación y su función*

Capa	Función
Presentación	Se encarga de mostrar la información al usuario mediante una interfaz gráfica y de recoger las entradas del usuario al interactuar con dicha interfaz.
API REST	Permite la comunicación entre la capa de presentación y la de negocio por medio de una API que utiliza una arquitectura REST. La capa de negocio se expone a la capa de presentación mediante diferentes servicios web REST, accesibles haciendo una petición HTTP a un <i>endpoint</i> .
Negocio	Contiene la lógica de negocio que es lo que determina la funcionalidad de la aplicación. En esta capa se decide qué hacer con las entradas del usuario recogidas en la capa de presentación y con los datos almacenados. Contiene reglas de validación de los datos, de autorización (para determinar a qué recursos tiene acceso un usuario), etc.
Persistencia	Accede a los datos y los presenta a la capa de negocio para que puedan ser usados en la aplicación. También se encarga de persistir en la capa de datos los cambios realizados en los datos.
Datos	Contiene los datos de la aplicación que, en nuestro caso, se almacenan en una base de datos relacional.

Figura 3.21*Diagrama de arquitectura de la aplicación por capas*

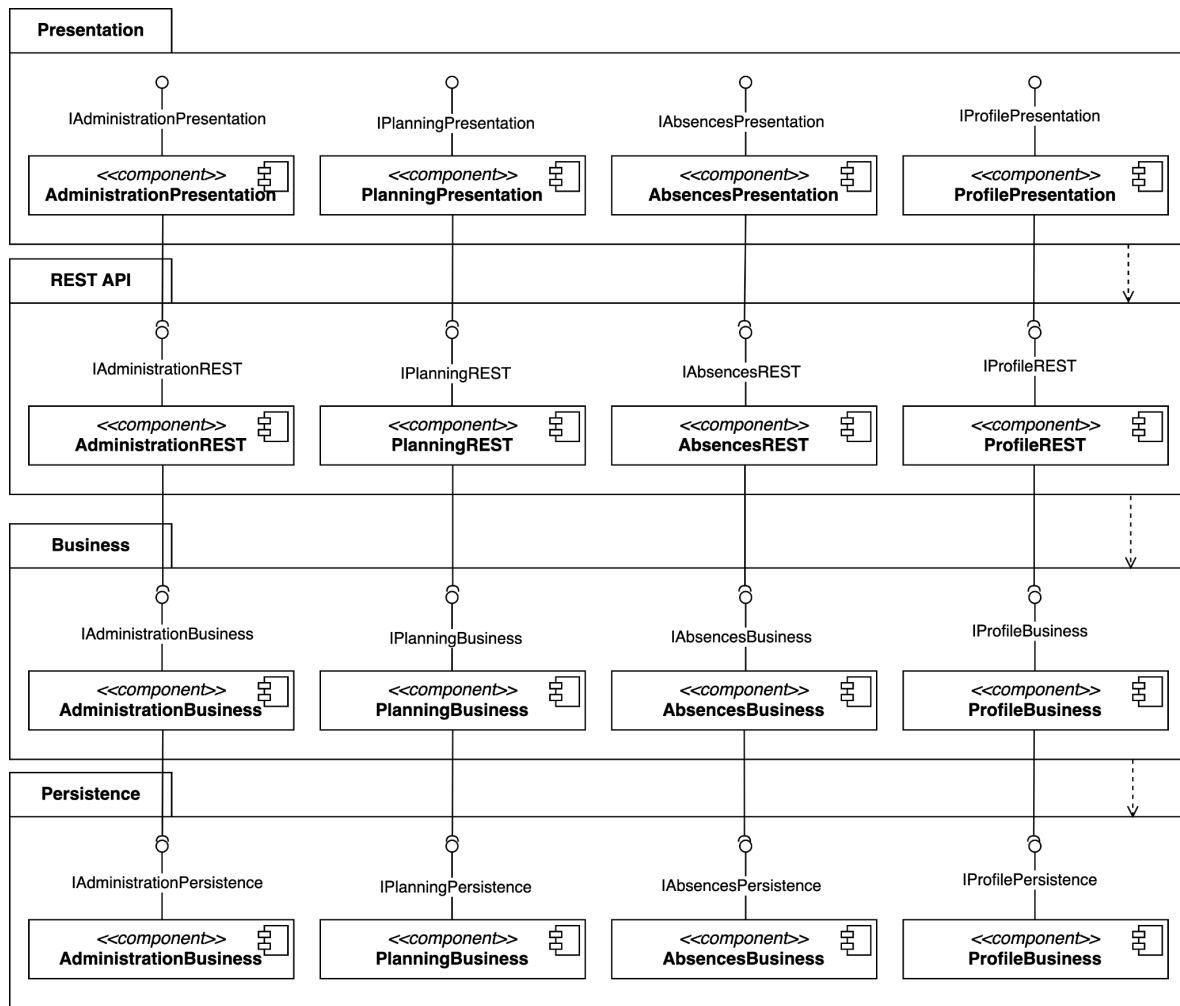
El uso de la API REST para exponer la capa de negocio, mediante un protocolo tan extendido como HTTP que no tiene un *frontend* estándar asociado, permite desacoplar completamente la capa de lógica de negocio de la capa de presentación, lo que tiene numerosas ventajas:

- Podemos disponer de varios canales de presentación (Web, aplicaciones móviles nativas, etc) que ofrezcan la misma funcionalidad, proporcionando una experiencia de usuario optimizada para diferentes dispositivos. Aunque eResourceKeeper estará disponible inicialmente únicamente como aplicación web, esto nos proporciona mayor flexibilidad. Si en un futuro se quisiera desarrollar una aplicación móvil, únicamente habría que añadir los cambios pertinentes en la capa de presentación, sin necesidad de realizar ningún cambio en el resto de capas.
- Podremos construir el *frontend* y el *backend* en diferentes *stacks* tecnológicos. De esta forma, tendremos la libertad de elegir la tecnología que consideremos más apropiada sin ninguna limitación. Además, si en un futuro queremos cambiar de tecnología se podrá hacer sin que tenga un gran impacto en el resto de la aplicación.
- Facilita el trabajo en paralelo y por tanto, aumenta el rendimiento de los equipos de desarrollo. Los desarrolladores *frontend* y *backend* pueden trabajar de forma independiente. Por un lado, el equipo de *backend* trabaja en desarrollar la lógica de negocio que expondrá mediante la API REST, mientras que el equipo de *frontend* desarrolla sobre la base de las interfaces de la API pactadas que puede simular fácilmente.
- Se puede desplegar de forma independiente el *frontend* y el *backend* de la aplicación. De esta forma, si se detecta un error en uno de las capas, no es necesario desplegar de nuevo toda la aplicación

El siguiente diagrama de componentes muestra la arquitectura de la aplicación organizada en capas y, dentro de cada capa, los diferentes componentes arquitectónicos de grano grueso con sus respectivas interfaces. Cada capa se representa en UML como un paquete. La interacción entre componentes de diferentes capas se da a través sus respectivas interfaces que, por cuestiones de legibilidad, se han incluido sin sus métodos y se detallan por separado a continuación.

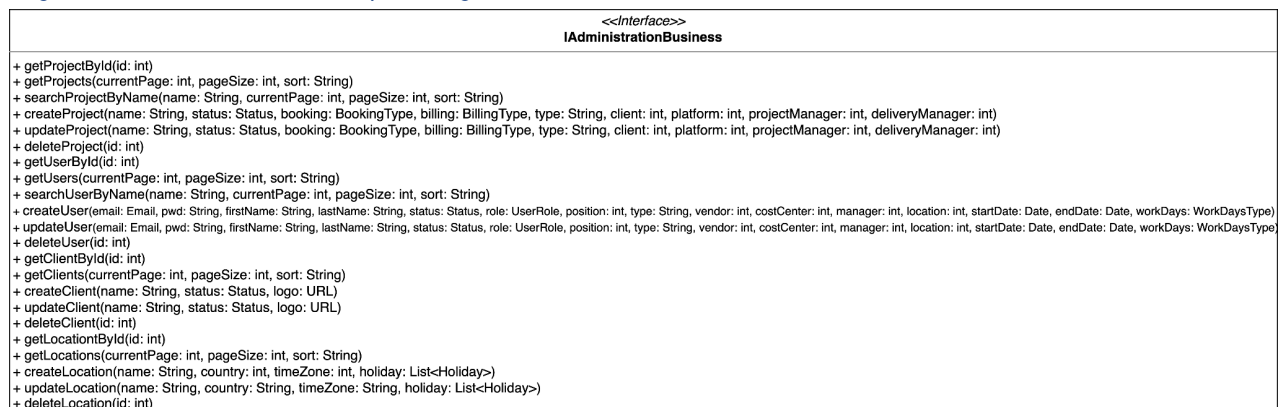
Para identificar los diferentes componentes hemos usado la agrupación de casos de uso realizada en el apartado 3.1, donde estos se han organizado según la funcionalidad con la que están relacionados.

Figura 3.22
Diagrama de componentes desde el punto de vista computacional



A continuación se detallan, a modo de ejemplo, las interfaces de la capa de negocio con la signatura de las operaciones que forman parte de ellas. El [anexo D](#) muestra la misma información de forma textual para facilitar la legibilidad. El resto de capas no se han desarrollado para evitar repetir la misma información.

Figura 3.23
Diagrama de las interfaces de la capa de negocio



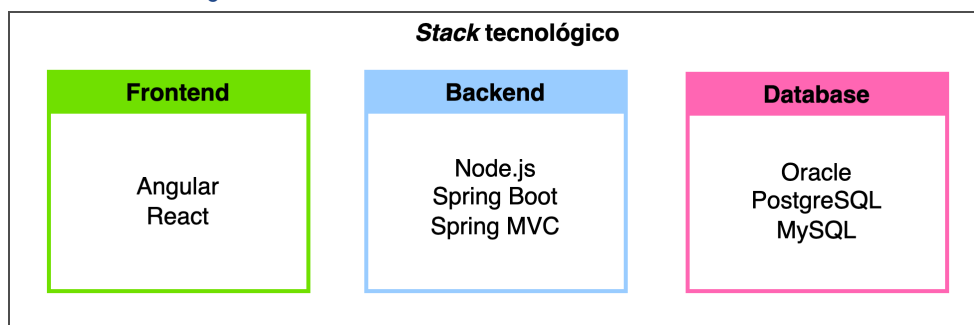
<<Interface>> IPlanningBusiness
<pre> + getTaskById(id: int): + createTask(name: String, description: String, booking: BookingType, billing: BillingType, startDate: Date, endDate: Date, hoursPerDay: int, project: int, user: int) + updateTask(name: String, description: String, booking: BookingType, billing: BillingType, startDate: Date, endDate: Date, hoursPerDay: int, project: int, user: int) + deleteTask(id: int) + getAllProjectsPlan(status: Status, currentPage: int, pageSize: int, sort: String, startDate: Date, endDate: Date) + getUserProjectsPlan(userId: int, currentPage: int, pageSize: int, sort: String, startDate: Date, endDate: Date) + searchProjectsPlan(query: String, status: Status, currentPage: int, pageSize: int, sort: String, startDate: Date, endDate: Date) + getFavouritesProjectsPlan(userId: int) + addProjectPlanToFavourites(projectId: int, userId: int) + getAllSchedules(currentPage: int, pageSize: int, sort: String, startDate: Date, endDate: Date) + getUserSchedule(userId: int, currentPage: int, pageSize: int, sort: String, startDate: Date, endDate: Date) + searchSchedules(query: String, currentPage: int, pageSize: int, sort: String, startDate: Date, endDate: Date) + getBench(currentPage: int, pageSize: int, sort: String, startDate: Date, endDate: Date) </pre>
<<Interface>> IAbsencesBusiness
<pre> + getTeamAbsences(managerId: int, currentPage: int, pageSize: int) + getUserAbsences(userId: int, currentPage: int, pageSize: int) + createAbsence(startDate: Date, endDate: Date, type: AbsenceType, comment: String) + cancelRequest(absenceId: int) + reviewRequest(absenceId: int) + approveRequest(absenceId: int) + declineRequest(absenceId: int, comment: String) </pre>
<<Interface>> IProfileBusiness
<pre> + login(email: Email, pwd: String) + logout() + updatePassword(oldPwd: String, newPwd: String) + recoverPassword(email: Email) + resetPassword(newPwd: String) </pre>

3.9 Tecnología

Para llevar a cabo nuestro desarrollo necesitamos tres componentes: el *frontend*, el *backend* y una base de datos. Para cada uno de ellos tenemos diferentes opciones entre las que elegir. Hemos hecho una primera selección de las posibles alternativas teniendo en cuenta los productos más habituales en cada caso, así como la experiencia previa de los desarrolladores que trabajan en la empresa Pivotree, ya que estos serían los que eventualmente llevarían a cabo el desarrollo de una futura implementación de la aplicación.

La siguiente figura muestra de forma esquemática las alternativas evaluadas:

Figura 3.24
Alternativas tecnológicas evaluadas



Tras analizar las diferentes opciones, el *stack* tecnológico escogido es el siguiente:

- *Frontend*: el *frontend* se desarrollará con Angular, al tratarse de un *framework* más completo que React, que únicamente es una librería. Angular incorpora patrones de diseño como la inyección de dependencias. Además, al estar escrito en TypeScript,

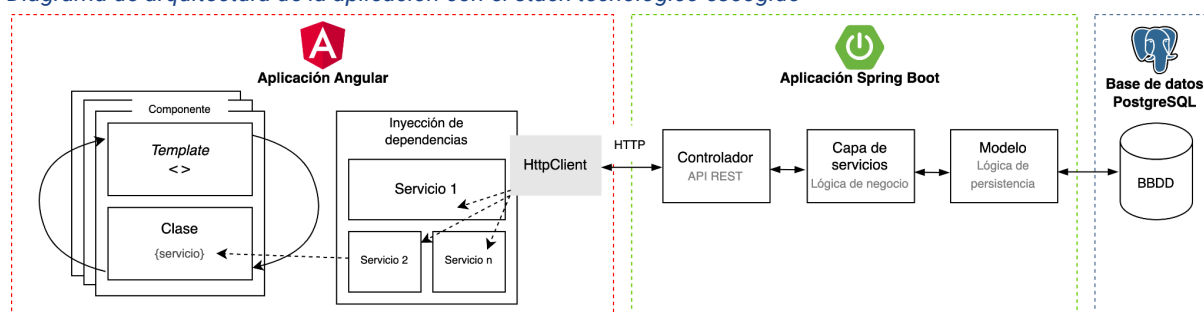
un lenguaje fuertemente tipado, nos aseguramos una mayor calidad del código. Aunque la curva de aprendizaje de Angular es mayor que la de React, esto no supone un problema al contar los desarrolladores de Pivotree con experiencia previa.

- **Backend:** dado que contamos con un gran número de desarrolladores con experiencia en Java, el *backend* se implementará con Spring Boot, un *framework* basado en Java, ampliamente utilizado para el desarrollo de APIs REST y que simplifica el desarrollo de aplicaciones con el *framework* de Spring. Descartamos la opción de implementar el *backend* con Node.js, ya que requiere desarrolladores con experiencia en JavaScript, y los que tenemos disponibles estarán ocupados con el desarrollo del *frontend* con Angular. Node.js sería una buena alternativa si contáramos con desarrolladores *full stack* o con un equipo de desarrolladores *frontend* más grande.
- **Base de datos:** Oracle podría ser la opción recomendada si se esperara almacenar gran cantidad de registros, del orden de millones, y hubiera severos requerimientos en cuanto a eficiencia, fiabilidad y disponibilidad. Como este no es nuestro caso vamos a descartarla y nos centraremos en PostgreSQL y MySQL, que no tienen coste y están ambas muy extendidas, por lo que es fácil obtener soporte en Internet en caso de problemas. Cualquiera de las dos puede cubrir sobradamente las necesidades actuales de eResourceKeeper, pero PostgreSQL ofrece algunas ventajas que podrían ser interesantes en el futuro, como por ejemplo un completo soporte del estándar SQL, cosa que no ocurre con MySQL [Ravoof, 2022]. Por otro lado, los técnicos de Pivotree tienen más experiencia en PostgreSQL, por lo que el desarrollo será más fácil y por tanto menos costoso. Por todo ello, el SGBD será PostgreSQL.

A continuación, se muestra un diagrama de la arquitectura resultante para el *stack* tecnológico escogido, donde podemos distinguir las capas antes mencionadas.

Figura 3.25

Diagrama de arquitectura de la aplicación con el *stack* tecnológico escogido



La capa de presentación está compuesta por la aplicación de Angular que está formada por un número variable de componentes. Cada componente está formado a su vez por un *template*, que consta de código HTML con algunas expresiones propias de la sintaxis de Angular, y una clase donde se define la lógica de presentación. Por su parte, la inyección de dependencias proporciona servicios a los componentes. Hay una gran variedad de servicios, uno de los cuales permite la comunicación con el *backend* mediante HTTP. Para ello Angular utiliza el servicio `HttpClient`, disponible con el *framework*.

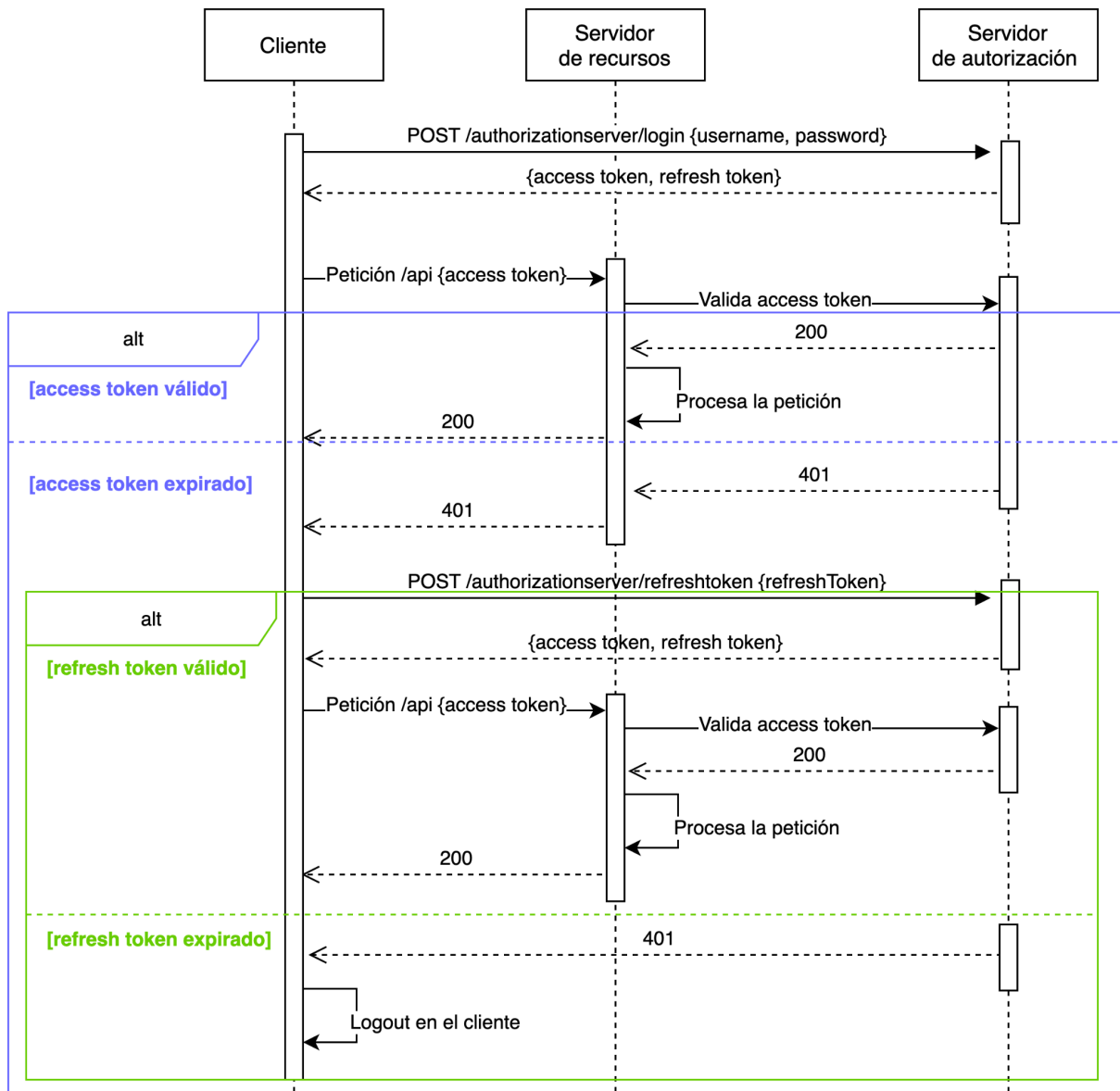
En la aplicación Spring Boot, el controlador se encarga de recibir y gestionar las peticiones HTTP que envía el cliente, pasándolas a la capa de servicios, si procede. La capa de

servicios, encargada de llevar a cabo toda la lógica de negocio de la aplicación, se comunica con el modelo, que es la capa encargada del acceso y la persistencia de los datos.

Para securizar la API REST (evitar que accedan a ella usuarios no autorizados) utilizaremos Spring Security el estándar de facto para proteger aplicaciones basadas en Spring, y el protocolo OAuth 2.0. El siguiente diagrama de secuencia muestra el flujo de autenticación y autorización.

Figura 3.26

Diagrama de secuencia del flujo de autenticación y autorización de la aplicación



Los componente involucrados son los siguientes:

- Cliente: la aplicación de Angular.
- Servidor de recursos: la API REST de Spring Boot que contiene los recursos a los que el cliente quiere acceder y que queremos securizar.

- Servidor de autorización: la aplicación encargada de validar las credenciales del usuario y expedir los *tokens*. Puede ser parte de la aplicación de Spring o se puede usar un proveedor externo.

El flujo de autenticación es el siguiente:

1. El cliente envía una petición al servidor de autorización para obtener un *token* de acceso proporcionando las credenciales que el usuario ha introducido en el formulario de la página de *login*.
2. El servidor de autorización valida las credenciales del usuario y devuelve un *token* de acceso y un *token* de refresco que el cliente almacena.
3. En cada petición posterior, el cliente añade como parte de la petición HTTP el *token* de acceso que el servidor de recursos validará contra el servidor de autorización para determinar si el usuario puede acceder al recurso que solicita.
4. Si el *token* de acceso ha caducado, se usará el *token* de refresco para obtener un nuevo *token*.
5. Si el *token* de refresco ha caducado, el cliente cierra la sesión, redirigiendo al usuario de nuevo a la página de *login*.

3.10 Diseño de la API REST

La API REST es una parte fundamental de la aplicación, ya que se trata de la interfaz que ofrece acceso a los datos almacenados y la funcionalidad asociada. Por ello, hemos considerado interesante trabajar en su diseño y crear la documentación correspondiente. Dicha documentación será la base sobre la que trabajarán tanto el equipo de *frontend* como el de *backend*.

Realizar el diseño de toda la API REST no es posible por el volumen de trabajo que conlleva y debido a la naturaleza del proyecto. Por tanto, únicamente se han documentado los siguientes *endpoints* a modo de ejemplo y los modelos requeridos por ellos:

```
Projects
GET /projects
GET /projects/{projectId}
POST /projects
PUT /projects
DELETE /projects/{projectId}
GET /projects/findOne
```

A la hora de trabajar en el diseño de la API REST se han tomado las siguientes decisiones:

- Se utilizan sustantivos en plural para representar la entidad sobre la que se quiere realizar la operación. El tipo de acción se indica mediante el método HTTP (GET, POST, PUT, DELETE, ...).
- Se utiliza JSON como formato principal para enviar y recibir datos.
- Cuando se obtienen listados de entidades se incluye la paginación y criterios de ordenación.
- Se devuelven respuestas estandarizadas, tanto en caso de éxito como de error.

- En caso de error se devuelve información suficiente para que el cliente pueda mostrar información detallada al usuario.

La documentación parcial de la API REST se encuentra en el [anexo E](#).

4. Conclusiones

Para poder llevar a cabo el análisis y diseño de eResourceKeeper, ha sido crucial el trabajo previo de obtención y gestión de requisitos. En este sentido, cabe destacar la relevancia del análisis que hemos realizado tanto de la solución actual como de otros productos existentes en el mercado. Este análisis nos ha permitido conocer las funcionalidades esperadas por los *stakeholders*, las carencias de la solución actual que había que suplir y los aspectos clave de los productos de la competencia.

La metodología de trabajo escogida también ha tenido un gran impacto en el resultado obtenido. Aunque el proyecto se ha ejecutado siguiendo una metodología de desarrollo en cascada que se caracteriza por ser secuencial, hemos aplicado retroalimentación a las fases previas para corregir los errores que hemos detectado e incorporar las sugerencias hechas por el profesor consultor. Esta retroalimentación se ha mostrado especialmente útil durante la fase de análisis de los requisitos en la que se han detectado requisitos que se habían pasado por alto en la etapa anterior.

El proceso de mejora continua nos ha permitido ir refinando el producto final y obtener una especificación de mayor calidad. Además, dado que la implementación quedaba fuera del alcance, creemos que la elección del modelo en cascada ha sido acertada, pues nos ha permitido obtener mayor documentación formal que la que proporcionan las metodologías de desarrollo ágiles. Una metodología de desarrollo ágil habría sido una mejor elección si el proyecto hubiese incluido la implementación.

Como queda patente en el resultado obtenido, hemos cumplido plenamente los objetivos propuestos para este TFG. Además, la especificación obtenida es una alternativa mejor que la herramienta actualmente en uso, pues provee todas las funcionalidades anteriores y otras nuevas, elimina las limitaciones existentes y proporciona una mayor eficiencia.

Para la consecución de los objetivos ha sido muy importante el trabajo realizado en la fase de planificación, que ha permitido tener muy claras las tareas a realizar en cada etapa del proyecto con el fin de alcanzar los objetivos establecidos. Aunque ha sido necesario realizar algún ajuste a dicha planificación a lo largo del proyecto, esta no ha sufrido cambios significativos y se han podido cumplir los hitos establecidos con éxito.

A lo largo del presente trabajo nos hemos encontrado con ciertas limitaciones, siendo la principal de ellas el carácter académico del proyecto. Esto nos ha impedido realizar ciertas actividades como, por ejemplo, entrevistar a los *stakeholders* para la obtención de requisitos o validar con ellos los requisitos seleccionados. De la misma forma, hemos tenido que adaptar ciertas prácticas al ámbito académico. El ejemplo más claro ha sido la fase de gestión de requisitos en la que la estimación coste/valor y priorización de los requisitos no se ha podido hacer de forma colaborativa y se ha tenido que efectuar de forma individual.

A nivel personal, la elaboración del presente trabajo me ha supuesto un gran aprendizaje. Por un lado, me ha permitido poner en práctica los conocimientos adquiridos en varias asignaturas del Grado de Ingeniería Informática y me ha dado la oportunidad de adquirir conocimientos nuevos que he tenido que aprender por mi cuenta. Además, aplicar dichos

conocimientos a un caso real me ha dado la oportunidad de constatar la utilidad de lo aprendido a largo de estos años en la UOC. Este trabajo me ha permitido, además, profundizar en el área de la ingeniería del software, lo cual me resultaba especialmente interesante por estar íntimamente relacionada con el trabajo que desempeño a diario. El trabajo realizado en este TFG es el que habitualmente desempeña el *business analyst* junto con el arquitecto del proyecto, que suele ser el punto de partida de mi trabajo. Por último, realizar este trabajo me ha permitido darme cuenta de la importancia de identificar correctamente los requisitos en una fase temprana del proyecto y de la utilidad de realizar un buen análisis de los requisitos.

Por todo lo expuesto anteriormente, estoy muy satisfecha con el resultado obtenido. No obstante, si tuviese la oportunidad de empezar de nuevo trataría de reducir el alcance de los requisitos seleccionados, para hacer una primera versión de la aplicación más sencilla. Esto disminuiría significativamente los recursos empleados en la documentación de los requisitos mediante casos de uso y en la elaboración de los *wireframes*, las dos partes más costosas del proyecto, y me permitirá profundizar más en la etapa de diseño técnico de la aplicación.

La línea de evolución más evidente del presente trabajo es la implementación de la aplicación web eResourceKeeper. Sin embargo, como paso previo a dicha implementación sería necesario, en primer lugar, validar los requisitos obtenidos con los *stakeholders*. Una vez validados, habría que elaborar los diseños de la interfaz de usuario aplicando la guía de estilos corporativa de la empresa Pivotree y asegurando que cumplen con el estándar WCAG 2.1. En cuanto al diseño técnico, sería interesante continuar trabajando en la especificación de la API REST y en el diseño de los componentes de software.

Para terminar, a continuación expongo de forma resumida los resultados obtenidos en el presente proyecto:

- Se ha propuesto el desarrollo de eResourceKeeper, una aplicación web para gestionar los recursos humanos de una empresa. Para ello se han planteado primero los objetivos y a continuación se ha realizado una planificación temporal, detallando las actividades a desarrollar, incluyendo una serie de hitos que se corresponden con las entregas de las PACs.
- Se han identificado los *stakeholders* del proyecto, así como los actores que participarán en la aplicación. Se ha desarrollado también un modelado de perfiles de usuario aplicando el concepto de 'usuario persona'.
- Se ha llevado a cabo un análisis de mercado, comparando las funcionalidades que ofrecen cuatro productos comerciales, elegidos de entre los más populares. A partir de dicho estudio se ha obtenido una lista de '*insights* positivos' y otra de '*insights* a evitar'.
- Se ha realizado un análisis detallado de la solución actualmente utilizada para la gestión de recursos humanos en la empresa Pivotree, consistente en una hoja de cálculo, con el fin de conocer sus funcionalidades y limitaciones.
- A partir de la información recabada en los dos análisis (el de mercado y el de la solución actual) se ha elaborado una lista de requisitos funcionales y una de requisitos no funcionales. A continuación, se ha estimado el coste y valor de los

requisitos funcionales y, aplicando unos criterios objetivos, se han seleccionado los que se van a implementar y se han documentado mediante casos de uso.

- Se ha elaborado el diagrama de casos de uso, y se ha modelizado la interfaz gráfica mediante *wireframes*. También se han realizado diversos diagramas de estados para mostrar el mapa navegacional de la aplicación. Algunas funcionalidades se han especificado más en detalle mediante varios diagramas UML (diagramas de estado y de actividades) complementarios.
- Se ha presentado el diagrama de clases de la aplicación, detallando los tipos de objetos que encapsulan la información, las relaciones existentes entre ellos y las reglas de integridad que deben cumplirse.
- Se ha efectuado el diseño conceptual y lógico de la base de datos, presentando el esquema relacional correspondiente.
- Para la implementación se ha propuesto una arquitectura basada en el modelo de capas, haciendo uso de tres componentes: un *frontend*, un *backend* y una base de datos. Para cada uno de ellos se plantean varias alternativas tecnológicas, de entre las que se lleva a cabo una selección en base a las funcionalidades de cada una y a los conocimientos y experiencia de los técnicos que eventualmente llevarían a cabo el desarrollo.

5. Glosario

Actor. Agrupación de usuarios que interactúan con la aplicación de una misma forma y que presentan un comportamiento propio.

API. Application programming interface.

BBDD. Base de datos.

BOE. Boletín Oficial del Estado.

CU. Caso de uso.

DAO. Data access object.

DSDM. Dynamic system development method.

HTTP. Hypertext Transfer Protocol.

HTTPS. Hypertext Transfer Protocol Secure.

iOS. iPhone Operating System.

IT. Information Technology.

KPI. Key performance indicator.

LOPD-GDD. Ley Orgánica de Protección de Datos Personales y garantía de los derechos digitales.

OAuth. Open Authorization.

OMG. Object Management Group

PAC. Prova d'avaluació contínua.

PC. Personal computer.

Perfil de usuario. Agrupación de usuarios de la aplicación hecha según sus características, basada en aspectos sociodemográficos, actitudinales, de expectativas, etc.

RDBMS. Sistemas de gestión de bases de datos relacionales

Requisito funcional. Requisito que describe una funcionalidad esperada de la aplicación, es decir, que describe el comportamiento de la aplicación ante los estímulos que le llegan del exterior.

Requisito no funcional. Requisito que establece algún tipo de restricción al conjunto de soluciones posible, sin hablar de funcionalidades.

REST. Representational state transfer.

RFC. Requisito funcional candidato.

RFS. Requisito funcional seleccionado.

RNF. Requisito no funcional.

RNF. Requisito no funcional candidato.

Rol. Sinónimo de **Actor**.

RRHH. Recursos humanos.

SGBD. Sistema gestor de base de datos.

SP. Story point.

SQL. Structured Query Language.

Stakeholder. Entidad (persona, organización o sistema) que tiene algún impacto o interés en el proyecto.

TBD. To be determined.

TFG. Trabajo fin de grado.

UML. Unified Modeling Language.

Usuario. Toda persona que utiliza la aplicación e interactúa con ella.

WCAG. Web Content Accessibility Guidelines.

W3C. World Wide Web Consortium.

XHTML. Extensible HyperText Markup Language.

6. Referencias

Agile Business Consortium (2014): *DSDM Agile Project Framework Handbook*. Disponible en: <https://www.agilebusiness.org/dsdm-project-framework.html> [Consulta: 30 de diciembre de 2022]

BOE (2018): “Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales”, *Boletín Oficial del Estado*, núm. 294 de 06 de diciembre de 2018. Disponible en: <https://www.boe.es/buscar/pdf/2018/BOE-A-2018-16673-consolidado.pdf> [Consulta: 30 de diciembre de 2022]

Booch, G., Rumbaugh, J. y Jacobson, I. (2005): *The Unified Modeling Language User Guide*, 2ª ed., Boston, MA: Addison-Wesley.

Brown, D. M. (2010): *Communicating Design: Developing Web Site Documentation for Design and Planning*, 2ª ed., San Francisco, CA: New Riders.

Chen, P. (1976): “The Entity-Relationship Model - Toward a Unified View of Data”, *ACM Transactions on Database Systems*, 1 (1), 9–36.
<https://dl.acm.org/doi/10.1145/320434.320440>

Clegg, D. y Barker, R. (1994): *Case Method Fast-Track: A RAD Approach*, Boston, MA: Addison-Wesley.

Cockburn, A. (2000): *Writing Effective Use Cases*, Boston, MA: Addison-Wesley.

Codd, E. F. (1970): “A Relational Model of Data for Large Shared Data Banks”, *Communications of the ACM. Classics*, 13 (6), 377–87.
<https://dl.acm.org/doi/10.1145/362384.362685>

Cohn, M. (2006): *Agile Estimating and Planning*, Upper Saddle River, NJ: Pearson Education.

Cooper, A. (2004): *The Inmates Are Running the Asylum: Why High Tech Products Drive Us Crazy and How to Restore the Sanity*, 2ª ed., Pearson Higher Education.

Date, C. J. y Darwen, H. (1997): *A Guide to the SQL Standard*, 4ª ed., Boston, MA: Addison-Wesley.

Horrocks, I. (1999): *Constructing the User Interface with Statecharts*, Boston, MA: Addison-Wesley.

Ishida, R. (2015): *Internationalization Quick Tips for the Web*, W3C. Disponible en: <https://www.w3.org/International/quicktips/index.en> [Consulta: 30 de diciembre de 2022].

OMG (2017): *OMG Unified Modeling Language (OMG UML) Version 2.5.1*. Disponible en: <https://www.omg.org/spec/UML/2.5.1/PDF> [Consulta 30 de diciembre de 2022].

Ravoof, F. (2022): *PostgreSQL vs MySQL: Explore Their 12 Critical Differences*. Disponible en: <https://kinsta.com/blog/postgresql-vs-mysql/#:~:text=MySQL%20is%20a%20purely%20relational,%2C%20ACID%2Dcompliant%20storage%20engine>. [Consulta: 30 de diciembre de 2022].

Richards, M. (2015). *Software Architecture Patterns*, Sebastopol, CA: O'Reilly.

Robertson, J. y Robertson, S. (2012). *Volere Requirements Specification Template, Edition 16-2012*. Disponible en: <https://www.cs.uic.edu/~i440/VolereMaterials/templateArchive16/c%20Volere%20template16.pdf> [Consulta: 30 de diciembre de 2022].

Royce, W. (1970): "Managing the Development of Large Software Systems", *Proceedings of IEEE WESCON*, 26 (August), 1-9. Disponible en: <http://www-scf.usc.edu/~csci201/lectures/Lecture11/royce1970.pdf> [Consulta: 31 de diciembre de 2022].

Wysocki, R. K. (2014): *Effective Project Management: Traditional, Agile, Extreme*, 7ª ed., New York, NY: John Wiley & Sons.

W3C (2018): *Web Content Accessibility Guidelines (WCAG) 2.1: W3C Recommendation 05 June 2018*, W3C. Disponible en: <https://www.w3.org/TR/WCAG21/> [Consulta: 30 de diciembre de 2022].

Anexo A - Perfiles de usuario

Figura A.1
Personaje arquetípico Persona 1

Persona 1



Julie

Demografía
 47 años
 Toronto, Canada

Formación universitaria
 Trabaja en el ámbito de la gestión de recursos desde 2003

Casada y con dos hijos de 8 y 12 años

Actor
 Admin

Puesto
 Capacity manager

Tolerancia al cambio


Acerca de Jullie

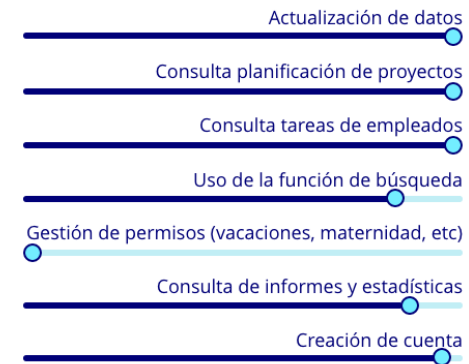
Tras casi 20 años especializándose en la gestión y planificación de recursos humanos en diferentes empresas multinacionales, empezó a trabajar en Pivotree hace 10 meses donde una de sus principales labores es el mantenimiento de la hoja de cálculo que se usa para la gestión de empleados y planificación de proyectos.

Todos los días revisa la información de la hoja de cálculo para poder acudir a las reuniones de planificación con los *delivery* y *resource managers* de la empresa. Dedicar gran parte de su jornada a revisar que los datos están actualizados y son correctos.

Tiene conocimientos amplios de tecnología, pues toda su trayectoria laboral está vinculada al sector de IT.

Necesidades y objetivos

- Sistema funcional e intuitivo.
- Poca diferencia con el sistema actual. Está acostumbrada a trabajar con la hoja de cálculo y se siente cómoda usándola.
- Actualización fácil de los datos.
- Elaboración de informes con estadísticas de los empleados.
- Integrar en el mismo sistema las vacaciones, ya que tiene un gran impacto en la planificación.



Pain points

- Replicación manual de los datos entre las diferentes pestañas de la hoja de cálculo.
- Importación manual de ciertos datos, por ejemplo las vacaciones, que residen en otra hoja de cálculo.

Figura A.2
Personaje arquetípico Persona 2

Persona 2



Tod

Demografía
53 años
Ottawa, Canada

Formación universitaria
Trabaja en el ámbito de la gestión de proyectos desde 2000

Casado y con dos hijos de 14 y 16 años

Actor
Editor

Puesto
Delivery manager

Tolerancia al cambio


Acerca de Tod

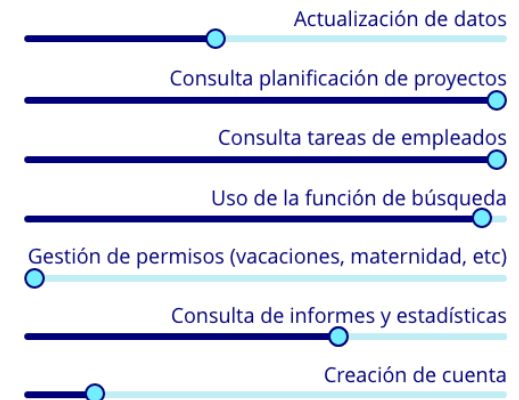
Su primera experiencia en el mundo laboral fue hace 30 años como desarrollador de software. Poco a poco fue ascendiendo hasta llegar a project manager, cargo en el que tiene mayor experiencia y con el que empezó a trabajar en Pivotree hace 4 años. Al cabo de 2 años pasó a ocupar su actual puesto de *delivery manager*.

Como *delivery manager* su responsabilidad principal es la ejecución exitosa de proyectos. Para ello trabaja con los diferentes equipos desempeñando el papel de patrocinador del proyecto. Además, se encarga de realizar una gestión avanzada de recursos humanos, teniendo en cuenta las tendencias comerciales dentro de la empresa. Para ello, hace un gran uso de la hoja de cálculo actual.

Tiene conocimientos avanzados de tecnología, pues toda su trayectoria laboral está vinculada al sector de IT y empezó como programador.

Necesidades y objetivos

- Sistema funcional e intuitivo.
- Poca diferencia con el sistema actual.
- Poder tener una vista que únicamente muestre los proyectos que gestiona.
- Posibilidad de tener proyectos tentativos y planificarlo con empleados provisionales.
- Poder crear tareas no facturables y que se distinguan fácilmente en la planificación.



Pain points

- No tiene forma de filtrar solo los proyectos que gestiona.
- Dificultad en el proceso de ver qué empleados están libres.

Figura A.3
Personaje arquetípico Persona 3

Persona 3



Janet

Demografía
44 años
Ottawa, Canada

Formación universitaria
Trabaja en el ámbito de la gestión de proyectos desde 2000

Casada y con un hijo de 7 años

Actor
Editor

Puesto
Project manager

Tolerancia al cambio

Acerca de Janet

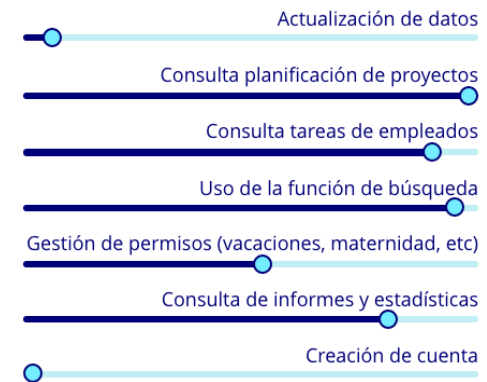
Janet lleva trabajando como *project manager* en empresas del sector IT desde hace 12 años. Empezó a trabajar en Pivotree hace 8 años y es actualmente una de las *project managers* con mayor trayectoria dentro de la empresa.

Una vez por semana revisa la información de la hoja de cálculo para verificar que la planificación de los proyectos que gestiona es correcta y comprobar si necesita comunicar que se extienda o acorte la asignación de algún empleado.

Tiene conocimientos amplios de tecnología pero le cuesta hacer uso de la actual hoja de cálculo.

Necesidades y objetivos

- Sistema funcional e intuitivo.
- Poder tener una vista que únicamente muestre los proyectos que gestiona.
- Poder consultar fácilmente todas las tareas de los recursos que están en sus proyectos.
- Poder crear tareas no facturables y que se distingan fácilmente en la planificación.
- Elaboración de informes con estadísticas de sus proyectos.



Pain points

- Solo le interesa ver los proyectos que gestiona pero no tiene forma de hacerlo.
- Dificultad en el proceso de ver qué otras asignaciones tienen planificadas los empleados que están en sus proyectos.

Figura A.4
Personaje arquetípico Persona 4

Persona 4



David

Demografía
33 años
Valencia, España

Formación universitaria
Trabaja en proyectos de desarrollo de software desde 2011

Casado y sin hijos

Actor
Viewer

Puesto
Resource manager

Tolerancia al cambio

Acerca de David

David empezó su trayectoria laboral como desarrollador de *software*. Tras 4 años trabajando en Pivotree como desarrollador *senior*, pasó a ser *resource manager*, puesto que ejerce desde hace 2 años.

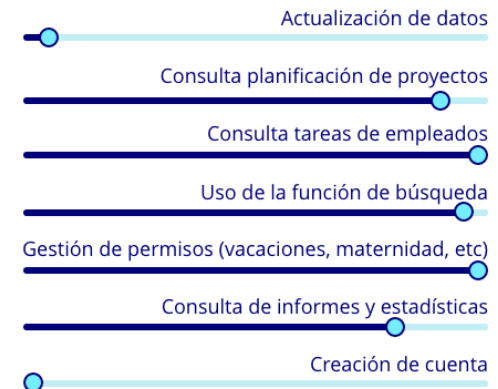
Como *resource manager* su responsabilidad principal es velar por el desarrollo profesional de su equipo. Para ello está en continuo contacto con los *delivery managers*, con el fin de conocer futuras oportunidades dentro de la empresa.

Una vez por semana revisa la información de la hoja de cálculo para verificar las tareas planificadas para los miembros de su equipo.

Pertenece a la generación *millennial*, por lo que la tecnología es una parte muy importante de su vida y ha crecido con ella.

Necesidades y objetivos

- Sistema funcional e intuitivo.
- Poder tener una vista que únicamente muestre los proyectos en los que participa gente de su equipo.
- Poder consultar fácilmente todas las tareas de sus recursos.
- Elaboración de informes con estadísticas de su equipo.
- Poder gestionar la aprobación de vacaciones directamente en la aplicación.
- Tener acceso a los posibles proyectos futuros y las necesidades de personal previstas.



Pain points

- Principalmente le interesa ver los proyectos en los que participa gente de su equipo pero no tiene forma de hacerlo.
- Con la importación manual de los datos de las vacaciones es muy fácil cometer un error.

Figura A.5
Personaje arquetípico Persona 5

Persona 5



Josep

Demografía
29 años
Valencia, España

Formación universitaria
Trabaja en proyectos de desarrollo de software desde 2015

Soltero y sin hijos

Actor
User

Puesto
Software developer

Tolerancia al cambio


Acerca de Josep

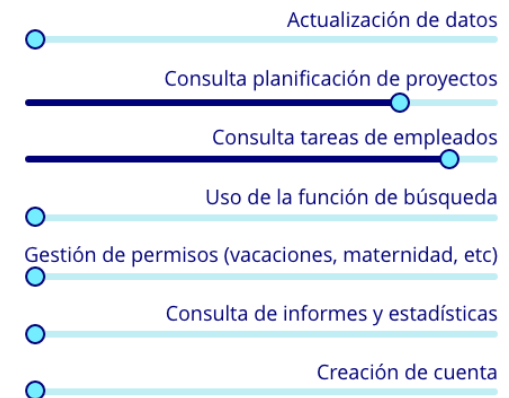
Comenzó su carrera laboral como desarrollador de software hace 7 años. Tras pasar por varias empresas, empezó a trabajar en Pivotree hace 2 años y medio.

Actualmente no tiene acceso a la hoja de cálculo por lo que la única forma que tiene de conocer sus tareas es a través de su *resource manager*.

Pertenece a la generación *millennial*, por lo que la tecnología es una parte muy importante de su vida y ha crecido con ella.

Necesidades y objetivos

- Sistema funcional e intuitivo.
- Poder consultar sus tareas.
- Conocer la planificación de aquellos proyectos en los que actualmente participa.



Pain points

- No tiene acceso a la hoja de cálculo y por tanto, no tiene forma de ver sus tareas.
- No tiene forma de ver la planificación para los proyectos en el que participa.

Anexo B - Análisis de productos existentes

Tabla B.1

Análisis de productos de la competencia

	Float	Runn	Teamdeck	ResourceGuru
1. Creación cuenta de usuario				
Nº permisos de acceso	6	5	3	4
Configurar permisos de acceso	Para ciertos roles de usuario existe la posibilidad de definir en detalle los permisos de escritura y/o lectura que tienen.	Permite definir qué tipo de acceso tiene a los datos financieros.	No	Control muy avanzado. Permite definir para cada rol de usuario qué permisos tiene sobre cada tipo de contenido.
2. Búsqueda				
Texto libre	Sí	Sí	Sí	Sí
Filtros preestablecidos	Sí	Sí	Sí	Sí
Filtros acumulativos	Sí	Sí	Sí	Sí
Filtros aplicados al cambiar página	Sí	Sí	No	Sí
Guardar búsquedas	Sí	Sí	Sí	Sí
3. Gestionar proyectos				
Cliente	Sí	Sí	Se puede conseguir creando un campo personalizado.	Sí
Facturable/No facturable	Sí	Sí	Se puede conseguir creando un campo personalizado o etiquetas.	Sí
Tentativo/Confirmado	Sí	Sí	Se puede conseguir creando un campo personalizado o etiquetas.	No. Solo tareas tentativas.
Etiquetas	Sí	Sí	Sí	No
Presupuesto	Sí	Sí	No	No
Identificación visual	Color	Logo del cliente	Color	Color
Creación	Sí	Sí	Sí. Muy escondido. No se puede hacer desde la vista de planificación. Hay que ir a la sección de ajustes.	Sí
Edición	Sí	Sí	Sí	Sí

	Float	Runn	Teamdeck	ResourceGuru
Eliminar	Sí. Pide confirmación.	No	Sí. Pide confirmación.	Sí. Pide confirmación.
Archivar	Sí	Sí. No permite archivar si el proyecto tiene tareas planificadas.	Sí	Sí. Al archivar un proyecto se pierde su planificación.
4. Gestionar empleados				
Puesto	Sí	Sí	Se puede conseguir creando un campo personalizado.	Sí
Departamento	Sí	No. Se puede usar el concepto de equipo, que es opcional.	Se puede conseguir creando un campo personalizado.	Sí
Tipo de trabajador	Sí. <i>Employee, contractor y placeholder.</i>	Sí. <i>Employee y contractor.</i>	Se puede conseguir creando un campo personalizado o etiquetas.	Sí. <i>Employee y contractor</i> por defecto. Permite crear más.
Tarifa por hora	Sí. Propia.	Sí. Puede heredar la del puesto o tener una propia.	Se puede conseguir creando un campo personalizado.	No
Etiquetas	Sí	Sí	No	No
Permisos de acceso	Sí	No. Los empleados son un tipo de entidad diferente que no tiene acceso a la aplicación.	No. Los empleados son un tipo de entidad diferente que no tiene acceso a la aplicación.	Sí
Empleado sin acceso	Sí	Sí	Sí	No
Email	Sí, opcional.	Sí	No	Sí
Zona horaria	No	No	No	Sí
Habilidades	No	Sí	No	Sí
Creación	Sí	Sí	Sí. Muy escondido. No se puede hacer desde la vista de planificación. Hay que ir a la sección de ajustes.	Sí
Edición	Sí	Sí	Sí	Sí
Eliminar	No	No	No	Sí
Archivar	Sí	Sí	No. Existe la opción de esconderlos del calendario y los informes.	Sí
5. Planificación de proyectos				
Fases	Sí	Sí	No	No
Hitos	Sí	No	Sí	No
Tareas	Sí. La tarea tiene un nombre que la	Sí. La tarea no tiene nombre. Queda	Sí. La tarea no tiene nombre. Queda	Sí. La tarea no tiene nombre.

	Float	Runn	Teamdeck	ResourceGuru
	identifica. Se puede crear una nueva o seleccionar una existente.	identificada por el empleado. Se crea una nueva, no se selecciona una ya existente.	identificada por el empleado. Se crea una nueva, no se selecciona una ya existente.	Queda identificada por el empleado y el proyecto.
Empleados	Sí. La planificación muestra para cada empleado las tareas que tiene planificadas en el proyecto.	Sí. La planificación muestra para cada empleado las tareas que tiene planificadas en el proyecto.	No. La planificación no muestra empleados, muestra un listado de tareas y cada tarea está asignada a un empleado.	Sí. La planificación muestra para cada empleado las tareas que tiene planificadas en el proyecto. Esta no es la vista por defecto, por defecto se muestran los empleados y sus tareas. Pero se puede agrupar por proyecto para conseguir ver la planificación prevista por proyectos.
Creación arrastrando en el timeline	Sí	Sí	Sí	Sí
Edición arrastrando en el timeline	Sí	Sí	Sí	Sí
Dividir tarea	Sí	Sí	Sí	Sí
Distingue visualmente proyectos facturable/no facturable	No	No	No. Se muestran al pasar el ratón por encima, si previamente se ha añadido como etiqueta.	No
Distingue visualmente tareas facturable/no facturable	No	Sí	No. Se muestran al pasar el ratón por encima, si previamente se ha añadido como etiqueta.	No
Muestra solo proyectos activos	Sí	Sí	Sí	Sí
Muestra proyectos tentativos	Sí. Se distingue visualmente por como se muestran sus tareas.	Sí, si se selecciona la opción de mostrar los proyectos tentativos. Hay un botón que permite alternar entre añadir o no los proyectos tentativos. Hay un indicador visual a nivel de proyecto que indica que es tentativo.	Sí. A no ser que se filtre por etiqueta.	No
Muestra todos los proyectos en la misma vista	Sí	Sí	Sí	Sí
Colapsar/expandir proyectos	Sí. Todos o de forma individual.	Sí. De forma individual.	No	Sí
Personalizar la forma en la que se muestran los proyectos	Sí. Calendario: permite definir si muestra días, semanas o meses. Define el nivel de detalle al que se muestran los días, es decir, el	Sí. Calendario: permite elegir entre mes, trimestre, semestre o año. Define el nivel de detalle al que se muestra el calendario, es decir, el tamaño de cada celda y si esta	Sí. Permite definir qué información se muestra en el calendario: tareas, vacaciones, horas registradas, hitos. Cada una de estas se puede mostrar/ocultar.	Sí. Solo permite hacer zoom en el calendario para ajustar el tamaño de la celda que representa un día. Solo hay dos niveles de zoom.

	Float	Runn	Teamdeck	ResourceGuru
	tamaño de cada celda. Densidad de las filas: compacta, confortable y espaciosa. Define como de alta es cada fila de un proyecto una vez expandido.	representa días o semanas. Agrupar: permite agrupar los proyectos por diferentes criterios: estado, cliente, equipo, etc.		
Ordenar proyectos	Sí. Criterios preestablecidos: nombre proyecto, cliente y fecha, ascendente y descendente.	Sí. Criterios preestablecidos: nombre proyecto, cliente, fecha inicio, fecha fin. Solo ascendente.	No	No
Vista individual de proyecto	Sí. Usando los filtros. No tiene vista de detalle. Solo el formulario de edición.	Sí. Usando los filtros. También tiene una vista de detalle del proyecto pero no muestra la planificación.	Sí. Usando los filtros. No tiene vista de detalle. Solo el formulario de edición.	Sí. Usando los filtros. No tiene vista de detalle. Solo el formulario de edición.
Marcar proyecto favorito	No	Sí	No	No
Horas planificadas	Sí. Para cada empleado muestra las horas que tiene planificadas en el proyecto. Y a nivel de proyecto se muestra el total de horas planificadas.	Sí. Para cada proyecto y tipo de puesto muestra las horas planificadas frente a las horas presupuestadas. No las muestra a nivel de empleado.	No	No
Puesto que ocupa cada empleado	Sí.	Los empleados se muestran agrupados por puesto.	No	Sí
Asignar al proyecto empleado sin tareas	Sí. Se muestra inactivo hasta que se le asigna una tarea.	No. Si el empleado no tiene ninguna tarea asignada se elimina al dejar la página.	No. Al crear una tarea es obligatorio seleccionar el empleado.	No. Solo se muestran los empleados y estos se pueden agrupar por proyecto. Pero no se muestran los proyectos como tal. Por tanto, no se puede asignar un empleado a un proyecto.
Intercambiar empleados	Sí. A nivel de empleado existe una opción de "Switch person"	No. Permite transferir una tarea de un empleado a otro, pero no intercambiar los empleados.	No, porque no se muestran empleados, solo tareas. Sí que se permite transferir una tarea de un empleado a otro.	No
Añadir empleado provisional	No	Sí	No	No
6. Tareas de empleados				
Misma vista que planificación de proyectos	No	No	No	Sí
Muestra todos los empleados en la misma vista	Sí	Sí	Sí	Sí
Colapsar/expandir empleados	No	Sí	No	No

	Float	Runn	Teamdeck	ResourceGuru
Vista individual de empleado	Sí. Usando los filtros. No tiene vista de detalle. Solo el formulario de edición.	Sí. Usando los filtros. También tiene una vista de detalle del empleado que muestra las tareas de ese empleado usando una tabla.	Sí. Usando los filtros. No tiene vista de detalle. Solo el formulario de edición.	Sí. Usando los filtros. Al pasar el ratón por encima muestra el detalle del empleado.
Personalizar la forma en la que se muestran los empleados	Sí. Calendario: permite definir si muestra días, semanas o meses. Define el nivel de detalle al que se muestran los días, es decir, el tamaño de cada celda. Densidad de las filas: compacta, confortable y espaciosa. Define como de alta es cada fila de un proyecto una vez expandido.	Sí. Calendario: permite elegir entre mes, trimestre, semestre o año. Define el nivel de detalle al que se muestra el calendario, es decir, el tamaño de cada celda y si esta representa días o semanas. Agrupar: permite agrupar los empleados por diferentes criterios: rol, equipo, habilidades, etc.	Sí. Permite definir qué información se muestra en el calendario: tareas, vacaciones, horas registradas, disponibilidad. Cada una de estas se puede mostrar/ocultar.	Sí. Solo permite hacer zoom en el calendario para ajustar el tamaño de la celda que representa un día. Solo hay dos niveles de zoom.
Ordenar empleados	Sí. Criterios preestablecidos: nombre, horas asignadas, departamento y rol, ascendente y descendente.	Sí. Criterios preestablecidos: disponibilidad, nombre, puesto y equipo, solo ascendente.	No	No
Desglose por tareas	Sí	Sí. Oculta por defecto. Se ve al expandir el empleado.	Sí	Sí
Resumen de horas sin asignar	Sí. Se muestra al lado del nombre el total para el periodo de tiempo seleccionado	Sí. Se muestra detallado a lo largo del timeline, por empleado. Por defecto se muestra el número de horas (7h free, 7 over). Se da la opción de mostrar el % de ocupación. Además, se puede mostrar como bloques continuos o partidos por semanas.	No	No
Resumen de disponibilidad	No	Sí	No	No
Avisa cuando sobrepasa la jornada laboral	Sí	Sí	Sí, pero no es muy evidente.	Sí
Estados (desde casa, de viaje,...)	Sí	No	No	No
7. Disponibilidad de empleados				
Diferentes tipos de permiso (pagado, maternidad, etc)	Sí	No	Sí	Sí
Calendarios festivos por localización	Sí. Se añaden automáticamente los festivos oficiales. Estos se pueden eliminar y se pueden añadir a mano otros festivos además de los reconocidos oficialmente.	No tiene calendario de festivos.	No. Pero cuando se añade un festivo hay que vincularlo a una "organization unit". Se podría crear una por localización. Hay que añadirlos a mano.	No tiene calendario de festivos. Hay que añadir festivos a mano para los empleados que se quiera.

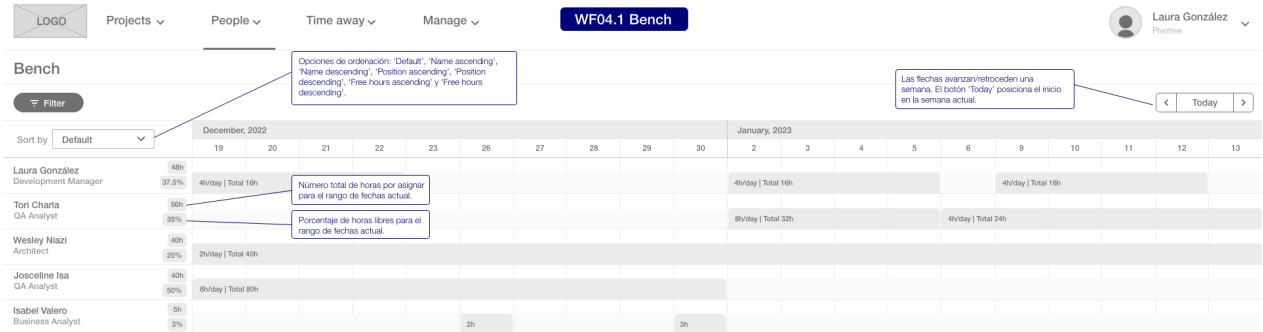
	Float	Runn	Teamdeck	ResourceGuru
Añadir permisos "provisionales"	Sí	No	No	No
Conflicto entre permiso y tareas	Se elimina automáticamente la tarea planificada.	Hay que resolver el conflicto a mano.	Se elimina automáticamente la tarea planificada.	Da a elegir entre eliminar el conflicto automáticamente o a mano.
Vista conjunta de todos los permisos	No. Pero se consigue fácilmente mediante el uso de filtros.	Sí	Sí, con la opción de mostrar solo las vacaciones, tanto en la vista de empleados como de proyectos.	No
Tipo de jornada (completa/parcial)	Sí	Sí	Sí	Sí
Configurar jornada completa	No	Sí	Sí	Sí
Configurar jornada parcial	Sí	Sí	Sí	Sí
Jornada parcial diferente duración por día	Sí	No	Sí	Sí
Fijar un rango de fechas en la jornada parcial	Sí	Sí	Sí	No. Se aplica a toda la planificación.
8. Estadísticas e informes				
Estadísticas de proyectos	Sí. Control presupuestario.	Sí. Control presupuestario.	Sí. Control presupuestario.	Sí. Solo horas planificadas. No hay estadísticas de presupuesto.
Capacidad del equipo, qué parte de esas horas tienen planificadas, si están por encima de su jornada laboral....	Sí. Se muestra el total para el periodo de tiempo seleccionado. También por empleado o departamento.	Sí. Se muestra el total para el periodo de tiempo seleccionado. También se muestra por empleado, por rol, por equipo, por tipo de trabajador.	Sí. Se muestra el total para el periodo de tiempo seleccionado. También se muestra por empleado y por proyecto. No se puede ver la información por puesto o departamento.	Sí. Se muestra el total para el periodo de tiempo seleccionado. También detallado por empleado.
Horas facturables y no facturables	Sí. Se muestra el total en una gráfica a lo largo del tiempo, por días, semanas o meses. También se muestran por empleado y por departamento.	Sí. Se muestra el total para el periodo de tiempo seleccionado. También se muestra por empleado, por rol, por equipo, por tipo de trabajador.	Sí. Se muestra el total para el periodo de tiempo seleccionado. También se muestra por empleado y por proyecto. No se puede ver la información por puesto o departamento	Sí. Se muestra el total para el periodo de tiempo seleccionado. También se muestra por empleado.
% Utilización (horas facturables y total)	El % total se da como parte del informe de empleados. No hay un informe separado.	Sí, informe separado.	Sí, informe separado.	El % total se da como parte del informe de empleados. No hay un informe separado.
Vacaciones	Se muestra como parte del informe de empleados.	No	Sí, informe separado.	Se muestra como parte del informe de empleados.
Definir rango de fechas	Sí	Sí	Sí	No, solo rangos preestablecidos.
Filtros	Sí	Sí	Sí	Sí

	Float	Runn	Teamdeck	ResourceGuru
Formato	Tabla y gráfica	Tabla	Tabla y gráfica	Tabla y gráfica
Exportar CSV	Sí	Sí	Sí	Sí
9. Otros				
Gestión de más de un equipo	Sí	No	Sí	No
Añadir campos personalizados	No	No	Sí	Sí
Importar csv proyectos y personas	Sí	Sí	Sí	No
Exportar csv planificación y tarea	Sí	Sí	Sí	No
Registro de horas trabajadas	Sí	Sí	Sí	No
Notificaciones	Sí	No	Sí	Sí
Precio	\$6 por usuario por mes	\$10 por usuario por mes	\$0.9 por empleado sin acceso a la aplicación por mes / \$3.60 por usuario con acceso por mes	\$1.25 por empleado sin acceso a la aplicación por mes / \$4.16 por usuario con acceso por mes

Anexo C - Wireframes

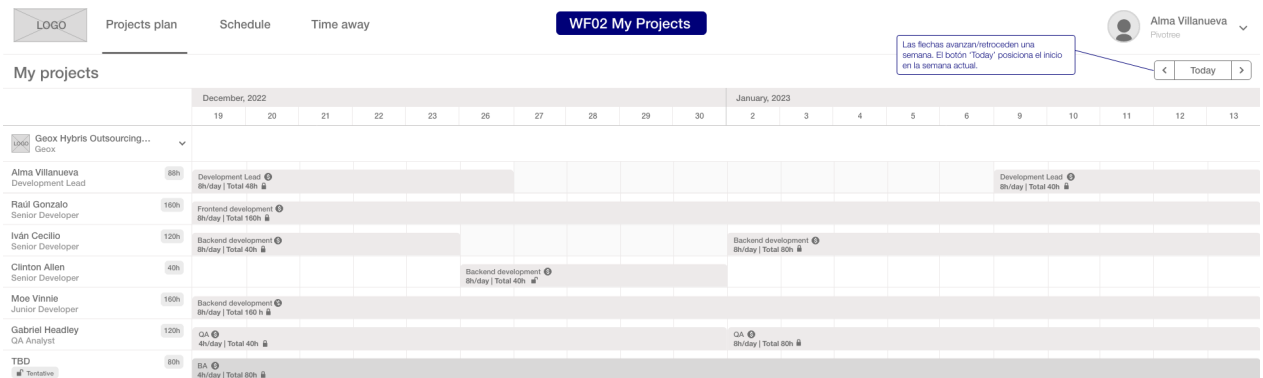
Bench - Página con los usuarios que tienen horas libres

Figura C.1
Usuarios con horas libres



My projects - Página con la planificación de los proyectos del usuario

Figura C.2
Planificación de los proyectos del usuario



My schedule - Página con la planificación de las tareas del usuario

Figura C.3
Planificación de las tareas del usuario



My time away - Página con los permisos solicitados por el usuario

Figura C.4
Permisos solicitados por el usuario

WF07.1 My time away

Alma Villanueva *Proctor*

My time away Request time away

Requested	Approved by	1 day
01/12/2023 Vacation	Requested 12/23/2022 Approved by Laura Gonzalez	Waiting
12/26/2022 - 12/29/2022 Vacation	Requested 10/24/2022 Approved by Laura Gonzalez	4 days Approved
05/06/2022 - 05/07/2022 Sick	Requested 04/12/2022 Approved by Laura Gonzalez	2 days Closed
08/22/2022 - 09/13/2022 Vacation	Requested 03/09/2022 Approved by Laura Gonzalez	18 days Closed
06/26/2022 - 06/27/2022 Vacation	Requested 05/07/2022 Approved by Laura Gonzalez	2 days Cancelled
06/16/2022 - 06/21/2022 Vacation	Requested 05/11/2022 Approved by Laura Gonzalez	4 days Declined

Comments

Leave a comment

Cancel Request

Aquí se muestra el historial de comentarios de esta petición. Por defecto se muestran colapsados. Si el usuario quiere consultarlos puede expandirlos.

Figura C.5
Formulario de solicitud de permiso

WF07.2 Create time away request

Reason (required)

Select a reason

Start date (required) End date (required)

mm/dd/yyyy mm/dd/yyyy

Comment

Cancel Submit

Aquí se muestra esta petición colapsada puede expandir

Team time away - Página con los permisos que tiene que gestionar

Figura C.6
Permisos pendientes de gestionar

WF06 Team time away

Laura González *Proctor*

Team time away

Requested	Submitted by	1 day
01/12/2023 Vacation	Requested 12/19/2022 Submitted by Alma Villanueva	Waiting
12/26/2022 - 12/29/2022 Vacation	Requested 10/24/2022 Submitted by Sàlvia Dugall	4 days In review
05/06/2022 - 05/07/2022 Sick	Requested 04/12/2022 Submitted by Tori Charla	2 days In review

Start date End date Reason Submitted by

05/06/2022 05/07/2022 Sick Tori Charla

Comments

Leave a comment

Approve Request Decline Request Request Changes

Sólo se muestran las solicitudes de días de permisos que tienen alguna acción pendiente. Las solicitudes que han sido aceptadas o denegadas no se muestran en este listado.

Aquí se muestra el historial de comentarios de esta petición. Por defecto se muestran colapsados. Si el usuario quiere consultarlos puede expandirlos.

Gestión de proyectos

Figura C.7
Listado de proyectos

Projects

Search projects Busqueda por nombre

Sort by

Opciones de ordenación: "Default", "Name ascending", "Name descending", "Client ascending" y "Client descending".

NAME	CLIENT	PLATFORM	STATUS	BILLING TYPE	BOOKING	TYPE	
Geox Hybris Outsourcing & Hybris OMS 2022	Geox	SAP	Active	Billable	Confirmed	Client	<input type="button" value="Edit"/> <input type="button" value="View"/>
Decathlon - PS ATG Development	Decathlon Inc.	Oracle	Active	Billable	Confirmed	Client	<input type="button" value="Edit"/> <input type="button" value="View"/>
Bloomie's MDM Microservice Integration	Macy's, Inc.	Oracle	Active	Billable	Tentative	Client	<input type="button" value="Edit"/> <input type="button" value="View"/>
Accessibility AA Update	Walmart	SAP	Active	Billable	Tentative	Client	<input type="button" value="Edit"/> <input type="button" value="View"/>
Dish MVP	Makro	VTEX	Active	Billable	Confirmed	Client	<input type="button" value="Edit"/> <input type="button" value="View"/>
Pivotree accelerator		VTEX	Active	Non-billable	Confirmed	Internal	<input type="button" value="Edit"/> <input type="button" value="View"/>
Accessibility training			Active	Non-billable	Confirmed	Training	<input type="button" value="Edit"/> <input type="button" value="View"/>
Cascades	Cascades	SAP	Inactive	Billable	Confirmed	Client	<input type="button" value="Edit"/> <input type="button" value="View"/>
Sony D2C Site	USC	SAP	Inactive	Billable	Confirmed	Client	<input type="button" value="Edit"/> <input type="button" value="View"/>
SAFe training			Inactive	Non-billable	Confirmed	Training	<input type="button" value="Edit"/> <input type="button" value="View"/>

Paginación visible si hay más de 10 ítems. Se muestran 10 ítems por página. Se marca la página activa.

Botón solo visible para usuarios Editor y Admin.

Botón solo visible para usuarios Editor y Admin.

Figura C.8
Detalle de un proyecto de cliente

Aldo Hybris Outsourcing & Hybris OMS 2022
Client project

PROJECT DETAILS

Name: Aldo Hybris Outsourcing & Hybris OMS 2022
Status: Active

Type: Client
Booking: Confirmed

Client: Aldo
Billing type: Billable

Platform: SAP

Delivery Manager: Tod Chipman

Project Manager: Janet Hays

Botón solo visible para usuarios Editor y Admin.

Botón solo visible para usuarios Editor y Admin.

Figura C.9
Detalle de un proyecto interno

Pivotree accelerator
Internal project

PROJECT DETAILS

Name: Pivotree Accelerator
Status: Active

Type: Internal
Booking: Confirmed

Client: -
Billing type: Non-billable

Platform: VTEX

Delivery Manager: James Smith

Project Manager: Janet Hays

Botón solo visible para usuarios Editor y Admin.

Botón solo visible para usuarios Editor y Admin.

Figura C.10
 Modificar un proyecto de cliente

LOGO Projects People Time away Manage **WF08.4 Client project edit** Laura González Pivotee

< Back to Projects

EDIT PROJECT

Name (required)
 Aldo Hybris Outsourcing & Hybris OMS 2022

Type (required)
 Client

Client (required)
 Aldo

Platform (required)
 VTEX

Delivery Manager (required)
 Tod Chipman

Project Manager (required)
 Janet Hays

Status (required)
 Active Inactive

Booking (required)
 Confirmed Tentative

Billing type (required)
 Billable Non-billable

Si el proyecto seleccionado es de tipo 'Internal' o 'Training' se oculta el campo 'Billing type'.

El campo 'Client' solo está disponible si el proyecto es de tipo 'Client', en cuyo caso además es obligatorio.

Delete project Cancel Save

Figura C.11
 Modificar un proyecto interno

LOGO Projects People Time away Manage **WF08.5 Internal project edit** Laura González Pivotee

< Back to Projects

EDIT PROJECT

Name (required)
 Pivotee Accelerator

Type (required)
 Internal

Platform
 VTEX

Delivery Manager (required)
 James Smith

Project Manager (required)
 Janet Hays

Status (required)
 Active Inactive

Booking (required)
 Confirmed Tentative

El campo 'Platform' es opcional para los proyectos que no son de tipo 'Client'.

Delete project Cancel Save

Figura C.12
 Crear un proyecto

LOGO Projects People Time away Manage **WF08.6 Create project** Laura González Pivotee

< Back to Projects

ADD NEW PROJECT

Name (required)
 Project name

Type (required)
 Select type

Platform (required)
 Select platform

Delivery Manager (required)
 Select Delivery Manager

Project Manager (required)
 Select Project Manager

Status (required)
 Active Inactive

Booking (required)
 Confirmed Tentative

Billing type (required)
 Billable Non-billable

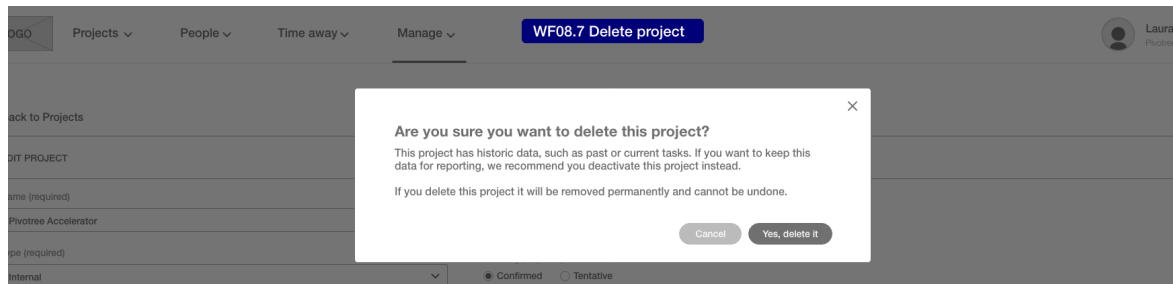
Si el proyecto seleccionado es de tipo 'Internal' o 'Training' se oculta el campo 'Billing type'.

Si el tipo de proyecto seleccionado es 'Client' se mostrará el desplegable 'Client' a continuación.

Deshabilitar el botón 'Save' hasta que se rellenen todos los campos obligatorios.

Cancel Save

Figura C.13
Eliminar un proyecto



Buscar un proyecto por nombre

Figura C.14
Búsqueda de proyecto por nombre con resultados

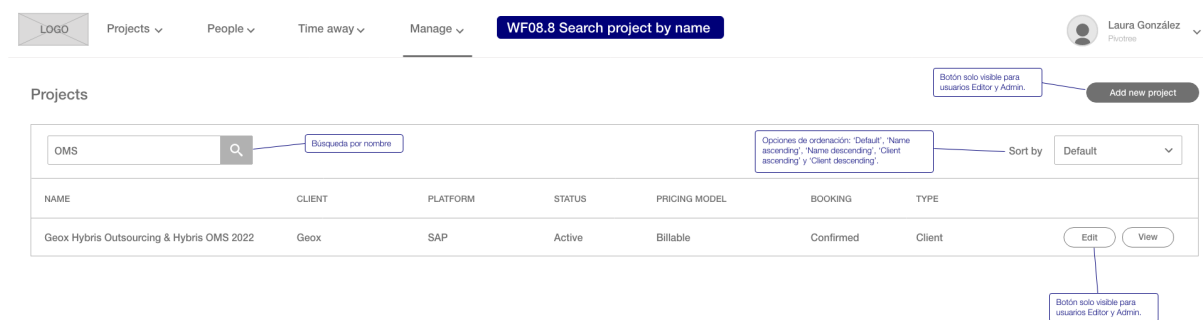


Figura C.15
Búsqueda de proyecto por nombre sin resultados



Gestión de usuarios

Figura C.16
Listado de usuarios

The screenshot shows the 'People list' interface. At the top, there is a navigation bar with 'LOGO', 'Projects', 'People', 'Time away', and 'Manage' menus. A blue button labeled 'WF09.1 People list' is visible. On the right, the user profile 'Laura González' is shown. Below the navigation, there is a search bar with the placeholder 'Search people' and a search icon. A callout box points to the search bar with the text 'Búsqueda por nombre'. To the right of the search bar is a 'Sort by' dropdown menu set to 'Default'. A callout box points to the dropdown with the text 'Opciones de ordenación: "Default", "ID ascending", "ID descending", "Name ascending", "Name descending", "Position ascending" y "Position descending"'. Below the search and sort options is a table with the following columns: EMAIL, NAME, POSITION, STATUS, ROLE, and TYPE. The table contains 8 rows of user data. Each row has 'Edit' and 'View' buttons. A callout box points to the 'Add new person' button with the text 'Botón solo visible para usuarios Admin.'. At the bottom of the table, there is a pagination control showing '1', '2', and '3' with arrows. A callout box points to the pagination with the text 'Paginación visible si hay más de 10 ítems. Se muestran 10 ítems por página. Se marca la página activa.'. Another callout box points to the 'Edit' and 'View' buttons with the text 'Botón solo visible para usuarios Admin.'.

EMAIL	NAME	POSITION	STATUS	ROLE	TYPE
alma.villanueva@pivotree.com	Alma Villanueva	Development Lead	Active	User	Employee
janet.hickey@pivotree.com	Janet Hickey	Project Manager	Active	Editor	Employee
laura.gonzalez@pivotree.com	Laura Gonzalez	Development Manager	Active	Viewer	Employee
tori.chawla@pivotree.com	Tori Chawla	QA Analyst	Active	User	Employee
sean.putler@pivotree.com	Sean Putler	Delivery Manager	Active	Editor	Employee
satis.duggal@pivotree.com	Satis Duggal	Senior Developer	Inactive	User	Contractor
wesley.niazi@pivotree.com	West Niazi	Architect	Active	User	Employee

Figura C.17
Detalle de un usuario de tipo Employee

The screenshot shows the 'Employee details' interface for Janet Hickey. At the top, there is a navigation bar with 'LOGO', 'Projects', 'People', 'Time away', and 'Manage' menus. A blue button labeled 'WF09.2 Employee details' is visible. On the right, the user profile 'Laura González' is shown. Below the navigation, the name 'Janet Hickey' and title 'Project Manager' are displayed. There is an 'Active' status indicator. A callout box points to the 'Delete' and 'Edit' buttons with the text 'Botón solo visible para usuarios Admin.'. Below this is a 'PERSON DETAILS' section with the following information:

Email	Location
janet.hickey@pivotree.com	Ottawa, Canada
Name	Start date
Janet Hickey	10/01/2015
Position	End date
Project Manager	-
Status	Work days
Active	Full time
Role	
Editor	
Type	
Employee	
Cost center	
SAP	
Manager	
Joel Leroux	

Figura C.18
Detalle de un usuario de tipo Contractor

LOGO Projects People Time away Manage **WF09.3 Contractor details** Laura González Pivotree

Sati Duggal
Senior Developer
Inactive

PERSON DETAILS

Delete Edit

Botón solo visible para usuarios Admin

Email	satis.duggal@pivotree.com	Location	Colombia
Name	Satis Duggal	Start date	05/15/2022
Position	Senior Developer	End date	10/01/2022
Status	Inactive	Work days	Part time (4 days)
Role	User	Monday: 8 hours	Tuesday: 8 hours
Type	Contractor	Wednesday: 8 hours	Thursday: 8 hours
Vendor	Talos		
Cost center	SAP		
Manager	-		

Figura C.19
Modificar un usuario de tipo Employee

LOGO Projects People Time away Manage **WF09.4 Employee edit** Laura González Pivotree

< Back to People

EDIT PERSON

Access Details Employee Details Contract Details

email (required) Janet.hickey@pivotree.com

First name (required) Janet Last name (required) Hickey

Location (required) Ottawa, Canada

Status (required) Active Inactive

Position (required) Project Manager

Start Date (required) 10/01/2015

Role (required) Editor

Type (required) Employee

End date mm/dd/yyyy

Password

Min. 8 characters, a combination of uppercase letters, lowercase letters, numbers, and symbols.

Confirm password

Send reset password email

Cost Center (required) SAP

Manager (required) Rob LeRoux

Work days (required) Full time Part time Null

Delete person Cancel Save

El campo "Manager" es obligatorio o no en función del puesto seleccionado en el campo "Position".

La duración de la jornada completa se define de forma global para toda la empresa.

Los usuarios con una jornada laboral nula no estarán disponibles para la asignación de tareas.

Los campos de fechas mostrarán un calendario donde el usuario puede seleccionar la fecha (datepicker). Opcionalmente, el usuario puede escribir la fecha a mano.

Si el usuario Admin no quiere generar la contraseña, opcionalmente puede enviar un email al usuario para que este la establezca (mismo email utilizado en el proceso de recuperación de contraseña).

El usuario Admin puede generar la contraseña en el momento de dar de alta el usuario y comunicársela utilizando un canal seguro. El usuario luego podrá actualizar la contraseña desde los ajustes su cuenta.

Figura C.20
 Modificar un usuario de tipo Contractor

EDIT PERSON

Access Details

Email (required):

Status (required): Active Inactive

Role (required):

Password:

Confirm password:

[Send reset password email](#)

Employee Details

First name (required): Last name (required):

Position (required):

Type (required): Vendor (required):

Cost Center (required):

Manager:

Contract Details

Location (required):

Start Date (required):

End date:

Work days (required): Full time Part time None

Hours per day (required): Monday: Tuesday: Wednesday: Thursday: Friday:

[Delete person](#) [Cancel](#) [Save](#)

Callouts:

- Los usuarios inactivos no podrán acceder a la aplicación.
- Si el usuario Admin no quiere generar la contraseña, opcionalmente puede enviar un email al usuario para que este le establezca (mismo email utilizado en el proceso de recuperación de contraseña).
- El campo "Manager" es opcional si el tipo de trabajador es "Contractor".
- El campo "Vendor" solo se muestra si el tipo de trabajador es "Contractor", en cuyo caso es obligatorio.
- Los usuarios con una fecha de fin anterior a la fecha actual no estarán disponibles para la asignación de tareas.

Gestión de clientes

Figura C.21
 Listado de clientes

Clients

[Add new Client](#)

Sort by:

NAME	PROJECTS	STATUS
Aldo	1	Active
Decathlon Inc.	1	Active
Macy's, Inc.	1	Active
Walmart	1	Active
Macy's Inc.	1	Active
Groupe Touchette	0	Inactive
Cascades	0	Active

[Edit](#) [View](#) (per row)

Paginación visible si hay más de 10 items. Se muestran 10 items por página. Se marca la página activa.

[<](#) [1](#) [2](#) [3](#) [>](#)

Callouts:

- Botón solo visible para usuarios Admin.
- Opciones de ordenación: "Default", "Name ascending" y "Name descending".
- Botón solo visible para usuarios Admin.

Figura C.22
Detalle de un cliente

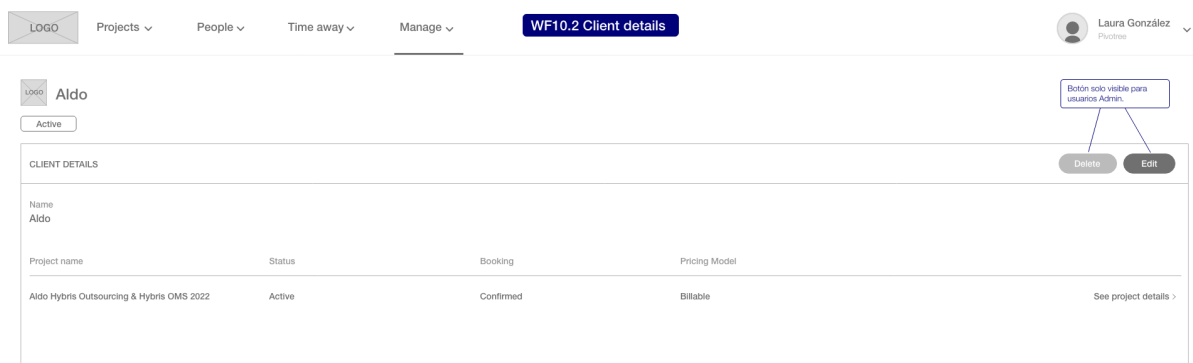


Figura C.23
Modificar un cliente

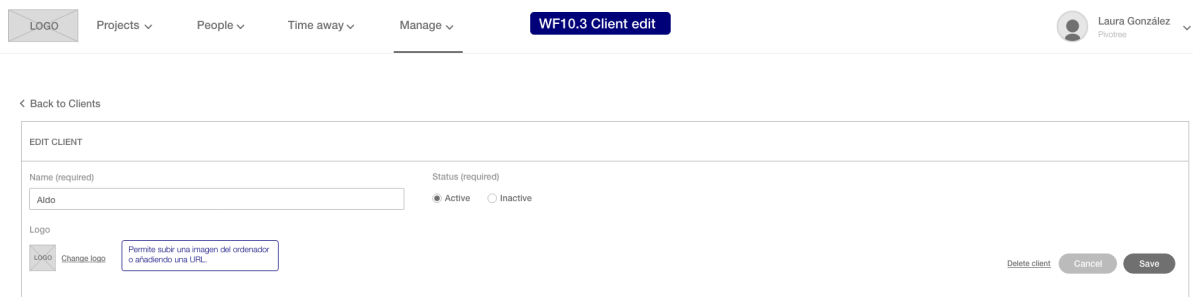
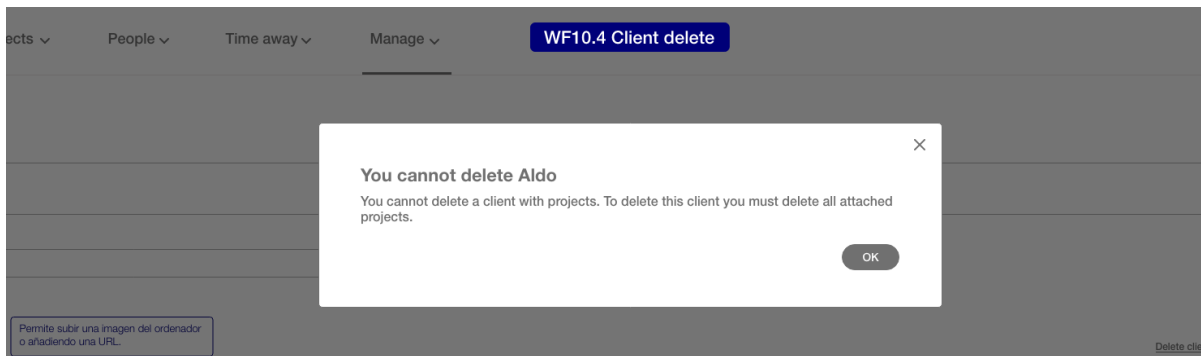


Figura C.24
Eliminar un cliente



Gestión de localizaciones

Figura C.25
Listado de localizaciones

Locations

Sort by: Default

NAME	COUNTRY	Edit	View
Valencia	Spain	Edit	View
Ottawa	Canada	Edit	View
Toronto	Canada	Edit	View
Montreal	Canada	Edit	View
Buenos Aires	Argentina	Edit	View
Rio de Janeiro	Brasil	Edit	View
Noida	India	Edit	View

1 2 3

Figura C.26
Detalle de una localización

Valencia

LOCATION DETAILS

Name	Country	Time Zone	Public Holidays
Valencia	Spain	(GMT+01:00) Central European Time - Madrid	January 1st, 2022: New Year's Day January 6th, 2022: Epiphany January 22nd, 2022: Saint Vicente (regional holiday) March 19th, 2022: Saint Josep (regional holiday) April 14th, 2022: Maundy Thursday (regional holiday) April 15th, 2022: Good Friday April 18th, 2022: Easter Monday (regional holiday) April 25th, 2022: Saint Vicente (regional holiday) June 24th, 2022: Saint John the Baptist Day (regional holiday) August 15th, 2022: Assumption of Mary October 12th, 2022: Hispanic Day November 1st, 2022: All Saints Day

Figura C.27
 Modificar una localización

LOGO
Projects ▾
People ▾
Time away ▾
Manage ▾

WF11.3 Location edit

Laura González
Pivotes

[← Back to Locations](#)

EDIT CLIENT

<small>Name (required)</small>	<small>Country (required)</small>	<small>Time Zone (required)</small>
<input type="text" value="Valencia"/>	<input type="text" value="Spain"/>	<input type="text" value="(GMT+01:00) Central European Time - Madrid"/>

Public holidays

Se muestran los festivos del año actual, el siguiente y el anterior.

2023					
2022					
<small>Date (required)</small>	<small>Description (required)</small>	<small>Date (required)</small>	<small>Description (required)</small>	<small>Date (required)</small>	<small>Description (required)</small>
<input type="text" value="01/01/2022"/>	<input type="text" value="New Year's Day"/>	<input type="text" value="04/14/2022"/>	<input type="text" value="Maundy Thursday (regional holiday)"/>	<input type="text" value="06/24/2022"/>	<input type="text" value="Saint John the Baptist Day (regional holiday)"/>
<input type="text" value="01/06/2022"/>	<input type="text" value="Epiphany"/>	<input type="text" value="04/15/2022"/>	<input type="text" value="Good Friday"/>	<input type="text" value="08/15/2022"/>	<input type="text" value="Assumption of Mary"/>
<input type="text" value="01/22/2022"/>	<input type="text" value="Saint Vicente (regional holiday)"/>	<input type="text" value="04/18/2022"/>	<input type="text" value="Easter Monday (regional holiday)"/>	<input type="text" value="10/12/2022"/>	<input type="text" value="Hispanic Day"/>
<input type="text" value="03/19/2022"/>	<input type="text" value="Saint Josep (regional holiday)"/>	<input type="text" value="04/25/2022"/>	<input type="text" value="Saint Vicente (regional holiday)"/>	<input type="text" value="11/01/2022"/>	<input type="text" value="All Saints' Day"/>
<small>Add holiday</small>					
2021					

Delete location
Cancel
Save

Anexo D - Interfaces de la capa de negocio

<<Interface>>
IAdministrationBusiness

+ *getProjectById(id: int)*
+ *getProjects(currentPage: int, pageSize: int, sort: String)*
+ *searchProjectByName(name: String, currentPage: int, pageSize: int, sort: String)*
+ *createProject(name: String, status: Status, booking: BookingType, billing: BillingType, type: String, client: int, platform: int, projectManager: int, deliveryManager: int)*
+ *updateProject(name: String, status: Status, booking: BookingType, billing: BillingType, type: String, client: int, platform: int, projectManager: int, deliveryManager: int)*
+ *deleteProject(id: int)*
+ *getUserById(id: int)*
+ *getUsers(currentPage: int, pageSize: int, sort: String)*
+ *searchUserByName(name: String, currentPage: int, pageSize: int, sort: String)*
+ *createUser(email: Email, pwd: String, firstName: String, lastName: String, status: Status, role: UserRole, position: int, type: String, vendor: int, costCenter: int, manager: int, location: int, startDate: Date, endDate: Date, workDays: WorkDaysType)*
+ *updateUser(email: Email, pwd: String, firstName: String, lastName: String, status: Status, role: UserRole, position: int, type: String, vendor: int, costCenter: int, manager: int, location: int, startDate: Date, endDate: Date, workDays: WorkDaysType)*
+ *deleteUser(id: int)*
+ *getClientById(id: int)*
+ *getClients(currentPage: int, pageSize: int, sort: String)*
+ *createClient(name: String, status: Status, logo: URL)*
+ *updateClient(name: String, status: Status, logo: URL)*
+ *deleteClient(id: int)*
+ *getLocationById(id: int)*
+ *getLocations(currentPage: int, pageSize: int, sort: String)*
+ *createLocation(name: String, country: int, timeZone: int, holiday: List<Holiday>)*
+ *updateLocation(name: String, country: String, timeZone: String, holiday: List<Holiday>)*
+ *deleteLocation(id: int)*

<<Interface>>
IPlanningBusiness

+ *getTaskById(id: int)*
+ *createTask(name: String, description: String, booking: BookingType, billing: BillingType, startDate: Date, endDate: Date, hoursPerDay: int, project: int, user: int)*
+ *updateTask(name: String, description: String, booking: BookingType, billing: BillingType, startDate: Date, endDate: Date, hoursPerDay: int, project: int, user: int)*
+ *deleteTask(id: int)*
+ *getAllProjectsPlan(status: Status, currentPage: int, pageSize: int, sort: String, startDate: Date, endDate: Date)*
+ *getUserProjectsPlan(userId: int, currentPage: int, pageSize: int, sort: String, startDate: Date, endDate: Date)*
+ *searchProjectsPlan(query: String, status: Status, currentPage: int, pageSize: int, sort: String, startDate: Date, endDate: Date)*
+ *getFavouritesProjectsPlan(userId: int)*

```
+ addProjectPlanToFavourites(projectId: int, userId: int)
+ getAllSchedules(currentPage: int, pageSize: int, sort: String, startDate: Date, endDate: Date)
+ getUserSchedule(userId: int, currentPage: int, pageSize: int, sort: String, startDate: Date, endDate:
Date)
+ searchSchedules(query: String, currentPage: int, pageSize: int, sort: String, startDate: Date, endDate:
Date)
+ getBench(currentPage: int, pageSize: int, sort: String, startDate: Date, endDate: Date)
```

**<<Interface>>
IAbsencesBusiness**

```
+ getTeamAbsences(managerId: int, currentPage: int, pageSize: int)
+ getUserAbsences(userId: int, currentPage: int, pageSize: int)
+ createAbsence(startDate: Date, endDate: Date, type: AbsenceType, comment: String)
+ cancelRequest(absenceId: int)
+ reviewRequest(absenceId: int)
+ approveRequest(absenceId: int)
+ declineRequest(absenceId: int, comment: String)
```

**<<Interface>>
IProfileBusiness**

```
+ login(email: Email, pwd: String)
+ logout()
+ updatePassword(oldPwd: String, newPwd: String)
+ recoverPassword(email: Email)
+ resetPassword(newPwd: String)
```

Anexo E - Documentación API REST

1. Overview

1.1. Version information

Version : 1.0.0

1.2. URI scheme

Scheme: https

Host : eresourcekeeper.com

BasePath : /api/v1

2. Security

2.1. oauth2_Password

Type : oauth2

Flow : password

Token URL : /authorizationserver/oauth/token

3. Endpoints

Summary

Projects

[GET /projects](#)

[GET /projects/{projectId}](#)

[POST /projects](#)

[PUT /projects](#)

[DELETE /projects/{projectId}](#)

[GET /projects/findByName](#)

2.1 Projects

GET /projects

Description

Finds all projects.

Security

[oauth2_Password](#)

Parameters

Type	Name	Description	Schema	Default
Query	currentPage <i>required</i>	Page number	integer	0
Query	PageSize <i>required</i>	Number of records per page	integer	20
Query	Sort <i>required</i>	Sorting of results	String	default

Responses

HTTP code	Description	Content type	Schema
200	OK	application/json	ProjectList
401	Unauthorized	application/json	Error
403	Forbidden	application/json	Error
404	Not found	application/json	Error

GET /projects/{projectId}

Description

Returns a single project.

Security

[oauth2_Password](#)

Parameters

Type	Name	Description	Schema	Default
Path	projectId <i>required</i>	Project identifier.	String	

Responses

HTTP code	Description	Content type	Schema
200	OK	application/json	Project
400	Bad request	application/json	Error
401	Unauthorized	application/json	Error
403	Forbidden	application/json	Error
404	Not found	application/json	Error

POST /projects

Description

Creates a project.

Security[oauth2_Password](#)**Parameters**

Type	Name	Description	Schema	Default
Body	project <i>required</i>	Project object	Project	

Responses

HTTP code	Description	Content type	Schema
201	Created	application/json	Project
400	Bad request	application/json	Error
401	Unauthorized	application/json	Error
403	Forbidden	application/json	Error
404	Not found	application/json	Error

Consumes

application/json

PUT /projects/{projectId}**Description**

Updates the project.

Security[oauth2_Password](#)**Parameters**

Type	Name	Description	Schema	Default
Path	projectId <i>required</i>	Project identifier.	String	
Body	project <i>required</i>	Project object	Project	

Responses

HTTP code	Description	Content type	Schema
201	Created	application/json	No content
400	Bad request	application/json	Error
401	Unauthorized	application/json	Error
403	Forbidden	application/json	Error
404	Not found	application/json	Error

Consumes

application/json

DELETE /projects/{projectId}**Description**

Deletes the project.

Security[oauth2_Password](#)**Parameters**

Type	Name	Description	Schema	Default
Path	projectId <i>required</i>	Project identifier.	String	

Responses

HTTP code	Description	Content type	Schema
200	OK	application/json	No content
401	Unauthorized	application/json	Error
403	Forbidden	application/json	Error
404	Not found	application/json	Error

GET /projects/findByName**Description**

Find projects by name.

Security[oauth2_Password](#)**Parameters**

Type	Name	Description	Schema	Default
Query	currentPage <i>required</i>	Page number	integer	0
Query	pageSize <i>required</i>	Number of records per page	integer	20
Query	query <i>optional</i>	Free text search query.	String	
Query	sort <i>required</i>	Sorting of results	String	default

4. Models

4.1 Client

Representation of a client.

Name	Description	Schema
id <i>required</i>	Client identifier.	integer
name <i>required</i>	Name of the client.	string

4.2 Error

Representation of an error.

Name	Description	Schema
code <i>required</i>	Error code.	string
message <i>optional</i>	Descriptive, human readable error message.	string

4.3 Pagination

Representation of a results pagination.

Name	Description	Schema
pagination <i>optional</i>	The current page number. The first page is number 0.	integer
pageSize <i>optional</i>	The number of results per page.	integer
sort <i>optional</i>	The selected sort option.	string
totalPages <i>optional</i>	The total number of pages, each of pageSize, required to display the totalResults.	integer
totalResults <i>optional</i>	The total number of matched results across all pages	integer

4.4 Person

Representation of a person.

Name	Description	Schema
id <i>required</i>	Person identifier.	integer
name <i>required</i>	Name of the person.	string

4.5 Platform

Representation of a platform.

Name	Description	Schema
id <i>required</i>	Platform identifier.	integer
name <i>required</i>	Name of the platform.	string

4.6 Project

Representation of a project.

Name	Description	Schema
billing <i>required</i>	Billing type of the project.	string
booking <i>required</i>	Type of booking of the project.	string
client <i>optional</i>	Client the project belongs to.	Client
deliveryManager <i>required</i>	Delivery manager running the project.	Person
id <i>required</i>	Project identifier.	integer
name <i>required</i>	Name of the project.	String
platform <i>optional</i>	Platform the project belongs to.	Platform
projectManager <i>required</i>	Project manager of the project.	Person
status <i>required</i>	Status of the project.	string
type <i>required</i>	Type of project.	String

4.7 ProjectList

Representation of a Product List.

Name	Description	Schema
pagination <i>required</i>	Pagination number	Pagination
projects <i>required</i>	List of projects	< Project > array
sorts <i>required</i>	List of sort options	< Sort > array

4.8 Sort

Representation a sort option

Name	Description	Schema
------	-------------	--------

<i>code required</i>	Code of sort option.	string
<i>name required</i>	Name of sort option.	string
<i>selected required</i>	Flag indicating if sort option is selected.	boolean