



Universidad de Valladolid

Escuela de Ingeniería Informática

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática

**Sistema online de cálculo de
presupuestos para ofertas de desarrollo
tecnológico**

Autor:
Dña. Silvia del Carmen Montero Vega



Universidad de Valladolid

Escuela de Ingeniería Informática

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática

**Sistema online de cálculo de
presupuestos para ofertas de desarrollo
tecnológico**

Autor:
Dña. Silvia del Carmen Montero Vega

Tutor:
Dr. Diego R. Llanos Ferraris

Resumen

El siguiente Trabajo de Fin de Grado consiste en el desarrollo de un sistema de gestión de presupuestos de proyectos de carácter tecnológico, destinado a su uso por grupos de investigación. Dicho sistema está compuesto por el desarrollo de una parte de lógica de servidor y una interfaz web que aproveche dicho servicio.

A partir de esta premisa surgen las bases para este proyecto: Diferentes utilidades que permitan a los usuarios del sistema gestionar de una forma más eficiente y gráfica los presupuestos y todo lo relacionado con ellos, permitiéndoles su creación, modificación, eliminación y visualización como archivos.

Dicho sistema se divide en dos partes, un servicio REST y una aplicación web. El servicio REST se ha implementado utilizando Javascript, el entorno de ejecución Node.js [24] y el framework Express [4]. Por otro lado, la aplicación web se ha desarrollado utilizando Javascript y el framework React.js [13]. Todo ello se ha preparado para su despliegue utilizando la tecnología Docker [3].

Se escogió realizar un sistema de estas características ya que hasta ahora la generación de presupuestos se estaba llevando a cabo utilizando software de hojas de cálculo, lo que no proporcionaba ningún tipo de estructura ni facilidades para la creación de los mismos y dificultaba su organización para utilizarlos en futuras referencias.

Abstract

The following Final Degree Project consists in the development of a budget management system, focused on budgets for technological projects, and intended for use by research groups. Such system is composed by the development of server-side logic and a web interface that takes advantage of the aforementioned server.

Based on this premise the features for this project arise: Different utilities that allow the users of the platform the ability to manage their budgets, by enabling them to create new ones or edit, delete and view as files the existing ones in a more effective and graphic manner.

Said system is divided in two parts, a REST service and a web application. The REST server has been implemented using Javascript, runtime environment Node.js [24] and the Express [4] framework. On the other hand, the web application has been implemented using Javascript and the framework React.js [13]. The whole system has been set to be deployed using Docker [3].

It was chosen to make a system with these features since until now, budget creation had been done using spreadsheets, which provided no structure or facilities for this task and also hindered their management and storage for the future.

Tabla de Contenidos

1. Introducción	17
1.1. Contexto y motivación	17
1.2. Objetivos	17
1.3. Punto de partida	18
1.4. Estructura de esta memoria	18
1.5. Dominio del problema	19
1.5.1. ¿Qué es un presupuesto?	19
1.5.2. Cálculo de un presupuesto	19
1.5.3. Tipos de costes en un proyecto	20
1.5.4. Pasos para el cálculo de un presupuesto	21
2. Estado del Arte	23
2.1. Software de Hojas de Cálculo	23
2.1.1. Ventajas	23
2.1.2. Inconvenientes	23
2.2. Easy Projects	23
2.2.1. Ventajas	24
2.2.2. Inconvenientes	24
2.3. Hubstaff	24
2.3.1. Ventajas	24
2.3.2. Inconvenientes	24
2.4. Conclusión	25
3. Análisis de requisitos	27
3.1. Definición de los actores	27
3.2. Requisitos de usuario	27
3.3. Requisitos funcionales	27
3.4. Requisitos no funcionales	29
3.5. Requisitos de información	29
3.6. Reglas de negocio	31
3.7. Estados de un presupuesto	31
4. Plan de proyecto	33
4.1. Resumen del proyecto	33
4.1.1. Propósito, alcance y objetivos	33
4.1.2. Definiciones y acrónimos	33
4.1.3. Método escogido para el proyecto	34
4.1.4. Evolución del plan	34

4.1.5.	Ciclo de vida del proyecto	34
4.2.	Gestión del proyecto	35
4.2.1.	Plan de puesta en marcha	35
4.2.2.	Plan de trabajo	35
4.2.3.	Plan de control	42
4.2.4.	Plan de gestión de riesgos	43
4.2.5.	Seguimiento del proyecto	45
4.3.	Presupuesto	45
5.	Modelo de análisis	47
5.1.	Casos de uso del sistema	47
5.1.1.	Diagrama general de casos de uso	47
5.1.2.	Caso de uso 1: Iniciar sesión	48
5.1.3.	Caso de uso 2: Cerrar sesión	48
5.1.4.	Caso de uso 3: Gestión de Usuarios	49
5.1.5.	Caso de uso 4: Gestión de parámetros del sistema	51
5.1.6.	Caso de uso 5: Gestión de presupuestos	54
5.1.7.	Caso de uso 6: Diseño de presupuestos	57
5.2.	Modelo de dominio	66
6.	Diseño	67
6.1.	Arquitectura del sistema	67
6.1.1.	Arquitectura del Back-end	67
6.1.2.	Arquitectura Front-end	70
6.2.	Interfaz de usuario	73
6.2.1.	Vista sobre inicio de sesión	73
6.2.2.	Vistas sobre diseño y gestión de presupuestos	73
6.2.3.	Revisión de presupuestos y ajustes de parámetros del sistema	81
6.2.4.	Gestión de usuarios	82
6.3.	Diagrama de paths de la aplicación	83
6.4.	Diagramas de secuencia en diseño	85
6.4.1.	Diagramas de secuencia en diseño del Back-end	85
6.4.2.	Diagramas de secuencia en diseño del Front-end	88
7.	Implementación	91
7.1.	Diagrama de despliegue	91
7.2.	Entorno de desarrollo	91
7.3.	Versiones necesarias de software	92
7.4.	Herramientas utilizadas	92
7.4.1.	Front-end	92
7.4.2.	Back-end	92
7.4.3.	Despliegue	93
7.5.	Problemas encontrados y soluciones aportadas	93
7.5.1.	Back-end	93
7.5.2.	Front-end	93
7.6.	Control de versiones	93
7.7.	Servidor de la aplicación	94

7.7.1.	Back-end	94
7.7.2.	Front-end	94
8.	Plan de pruebas y evaluación	95
8.1.	Pruebas para el back-end	95
8.1.1.	Pruebas sobre gestión de usuarios	95
8.1.2.	Pruebas sobre gestión de cargos	97
8.1.3.	Pruebas sobre gestión de parámetros	99
8.1.4.	Pruebas sobre gestión de artículos	99
8.1.5.	Pruebas sobre gestión de presupuestos	101
8.1.6.	Pruebas sobre gestión de materiales	104
8.1.7.	Pruebas sobre gestión de servicios	106
8.1.8.	Pruebas sobre gestión de tareas	108
8.1.9.	Pruebas sobre gestión de viajes	110
8.2.	Pruebas para el front-end	112
8.2.1.	Pruebas sobre autenticación de usuarios	112
8.2.2.	Pruebas sobre gestión de usuarios	112
8.2.3.	Pruebas sobre gestión de cargos	113
8.2.4.	Pruebas sobre gestión de parámetros	115
8.2.5.	Pruebas sobre gestión de artículos	115
8.2.6.	Pruebas sobre gestión de presupuestos	117
8.2.7.	Pruebas sobre gestión de materiales	121
8.2.8.	Pruebas sobre gestión de servicios	123
8.2.9.	Pruebas sobre gestión de tareas	124
8.2.10.	Pruebas sobre gestión de viajes	125
9.	Manual de programador	127
9.1.	Back-end	127
9.1.1.	Directivas generales	127
9.1.2.	Carpeta routes	127
9.1.3.	Carpeta services	128
9.1.4.	Carpeta models	129
9.1.5.	Carpeta tools	129
9.1.6.	Otros ficheros relevantes	129
9.1.7.	Pruebas en Postman	129
9.2.	Front-end	133
9.2.1.	Directivas generales	133
9.2.2.	Carpeta components	134
9.2.3.	Carpeta pages	135
9.2.4.	Carpeta API	135
9.2.5.	Carpeta context	135
9.2.6.	Carpeta utils	135
9.2.7.	Otros ficheros de interés	136
10.	Manual de instalación	137
10.1.	Instalación de Docker	137
10.1.1.	Requisitos de software	137

10.1.2. Procedimiento	137
10.1.3. Breve introducción a Docker	137
10.2. Base de datos	138
10.2.1. Instalación de MySQL	138
10.2.2. Base de datos local	138
10.2.3. Despliegue con Docker	139
10.2.4. Conexión con Back-end	139
10.3. Back-end	140
10.3.1. Software necesario	140
10.3.2. Despliegue local	140
10.3.3. Despliegue con Docker	140
10.4. Front-end	140
10.4.1. Software necesario	140
10.4.2. Despliegue local	140
10.4.3. Despliegue con Docker	140
11. Manual de usuario	143
11.1. Acceso a la aplicación	143
11.2. Administrador del Sistema	143
11.3. Jefes	145
11.3.1. Revisión de Presupuestos	145
11.3.2. Gestión de Cargos	146
11.3.3. Parámetros del Sistema	149
11.4. Comerciales	149
11.4.1. Administración de Presupuestos	149
11.4.2. Editor de Presupuestos	150
11.5. Salir del sistema	158
12. Conclusiones y trabajo futuro	159
12.1. Conclusiones	159
12.2. Trabajo futuro	160

Lista de Figuras

3.1. Estados de un presupuesto	31
4.1. Vista general del plan de trabajo	41
4.2. Fase de análisis	41
4.3. Fase de diseño	41
4.4. Fase de implementación	42
4.5. Fase de pruebas	42
4.6. Fase de mantenimiento	42
5.1. Diagrama general de casos de uso	47
5.2. Diagrama en detalle de gestión de usuarios	49
5.3. Diagrama en detalle de gestión de parámetros	51
5.4. Diagrama en detalle de gestión de presupuestos	54
5.5. Diagrama en detalle de diseño de presupuestos	57
5.6. Gestión de materiales de un presupuesto	61
5.7. Gestión de servicios de un presupuesto	65
5.8. Gestión de tareas de un presupuesto	65
5.9. Gestión de viajes de un presupuesto	65
5.10. Modelo de dominio	66
6.1. Arquitectura usada para el Back-end	68
6.2. Arquitectura usada para el Back-end (Simplificada)	69
6.3. Funcionamiento React + Context API	71
6.4. Arquitectura usada para el Front-end	72
6.5. Inicio de Sesión	73
6.6. Panel de Gestión de Presupuestos	74
6.7. Creación (o edición) de un presupuesto - Vista General	74
6.8. Creación (o edición) de un presupuesto - Gestión de Materiales	75
6.9. Añadir material a un presupuesto	75
6.10. Añadir artículo al histórico del sistema	76
6.11. Editar material de un presupuesto	76
6.12. Creación (o edición) de un presupuesto - Gestión de Servicios	77
6.13. Añadir un servicio a un presupuesto	77
6.14. Editar servicio de un presupuesto	78
6.15. Eliminar servicio de un presupuesto	78
6.16. Creación (o edición) de un presupuesto - Gestión de viajes	79
6.17. Creación (o edición) de un presupuesto - Gestión de Tareas	79
6.18. Creación (o edición) de un presupuesto - Gestión de Tareas	80

6.19. Gestión de parámetros y cargos del sistema	81
6.20. Panel de revisión de presupuestos	81
6.21. Panel de control para la gestión de usuarios	82
6.22. Registro de un Usuario	82
6.23. Diagrama de Paths simplificado de la aplicación	83
6.24. Diagrama de Paths de la aplicación	84
6.25. Inicio de Sesión	85
6.26. Registrar Usuario	86
6.27. Visualizar usuarios del sistema	86
6.28. Eliminar usuarios del sistema	86
6.29. Editar Presupuesto	87
6.30. Ver Materiales de un Presupuesto	88
6.31. Añadir Material a un Presupuesto	88
6.32. Editar un material	89
6.33. Eliminar un material	89
7.1. Diagrama de Despliegue de la aplicación	91
9.1. Header de autorización de las peticiones	130
9.2. GET todos los cargos	130
9.3. GET cargo con identificador 1	131
9.4. POST añadir un cargo	132
9.5. PUT editar un cargo	132
9.6. DELETE un cargo	133
11.1. Inicio de sesión	143
11.2. Panel de Administración de Usuarios	144
11.3. Dialogo de confirmación - Eliminar usuario	144
11.4. Panel de Revisión de Presupuestos	145
11.5. Visualizador de archivos PDF	146
11.6. Panel de Gestión de Cargos	147
11.7. Crear Cargo	147
11.8. Editar Cargo	148
11.9. Editar Parámetros	149
11.10. Panel de Administración de Presupuestos	150
11.11. Formulario del Presupuesto	151
11.12. Resumen del Presupuesto	152
11.13. Historial de Artículos	153
11.14. Formulario de Materiales	154
11.15. Tabla de Materiales del Presupuesto	154
11.16. Historial de un Artículo	155
11.17. Tabla de Servicios del Presupuesto	156
11.18. Tabla de Viajes del Presupuesto	157
11.19. Tabla de Tareas del Presupuesto	158

Lista de Tablas

4.1. Actividad 01	36
4.2. Actividad 02	36
4.3. Actividad 03	36
4.4. Actividad 04	36
4.5. Actividad 05	36
4.6. Actividad 06	36
4.7. Actividad 07	37
4.8. Actividad 08	37
4.9. Actividad 09	37
4.10. Actividad 10	37
4.11. Actividad 11	37
4.12. Actividad 12	37
4.13. Actividad 13	37
4.14. Actividad 14	38
4.15. Actividad 15	38
4.16. Actividad 16	38
4.17. Actividad 17	38
4.18. Actividad 18	38
4.19. Actividad 19	38
4.20. Actividad 20	38
4.21. Actividad 21	39
4.22. Actividad 22	39
4.23. Actividad 23	39
4.24. Actividad 24	39
4.25. Actividad 25	39
4.26. Actividad 26	39
4.27. Actividad 27	39
4.28. Actividad 28	40
4.29. Actividad 29	40
4.30. Actividad 30	40
4.31. Actividad 31	40
4.32. Actividad 32	40
4.33. Actividad 33	40
4.34. Presupuesto (IVA no incluido)	45
5.1. Caso de Uso 1 - Iniciar sesión	48
5.2. Caso de Uso 2 - Cerrar sesión	48

5.3. Caso de Uso 3.1 - Ver lista de usuarios	49
5.4. Caso de Uso 3.2 - Registrar usuario	50
5.5. Caso de Uso 3.3 - Eliminar usuario	50
5.6. Caso de Uso 4.1 - Editar parámetros del sistema	51
5.7. Caso de Uso 4.2 - Ver lista de cargos	52
5.8. Caso de Uso 4.3 - Crear cargo	52
5.9. Caso de Uso 4.4 - Eliminar cargo	53
5.10. Caso de Uso 4.5 - Editar cargo	53
5.11. Caso de Uso 5.1 - Revisar presupuestos	54
5.12. Caso de Uso 5.2 - Aprobar presupuesto	55
5.13. Caso de Uso 5.3 - Rechazar presupuesto	55
5.14. Caso de Uso 5.4 - Obtener archivo presupuesto	56
5.15. Caso de Uso 6.1 - Ver lista de presupuestos	57
5.16. Caso de Uso 6.2 - Crear presupuesto	58
5.17. Caso de Uso 6.3 - Eliminar presupuesto	58
5.18. Caso de Uso 6.4 - Editar presupuesto	59
5.19. Caso de Uso 6.5 - Duplicar presupuesto	59
5.20. Caso de Uso 6.6 - Obtener archivo presupuesto	60
5.21. Caso de Uso 6.7 - Ver lista de materiales	61
5.22. Caso de Uso 6.8 - Añadir material	62
5.23. Caso de Uso 6.9 - Eliminar material	62
5.24. Caso de Uso 6.10 - Editar material	63
5.25. Caso de Uso 6.11 - Crear artículo	63
5.26. Caso de Uso 6.12 - Eliminar artículo	64
8.1. Prueba 1.1	96
8.2. Prueba 1.2	96
8.3. Prueba 1.3	96
8.4. Prueba 1.4	97
8.5. Prueba 2.1	97
8.6. Prueba 2.2	98
8.7. Prueba 2.3	98
8.8. Prueba 2.4	98
8.9. Prueba 3.1	99
8.10. Prueba 3.2	99
8.11. Prueba 4.1	100
8.12. Prueba 4.2	100
8.13. Prueba 4.3	100
8.14. Prueba 4.4	101
8.15. Prueba 5.1	101
8.16. Prueba 5.2	102
8.17. Prueba 5.3	102
8.18. Prueba 5.4	102
8.19. Prueba 5.5	103
8.20. Prueba 5.6	103
8.21. Prueba 6.1	104
8.22. Prueba 6.2	104

8.23. Prueba 6.3	105
8.24. Prueba 6.4	105
8.25. Prueba 7.1	106
8.26. Prueba 7.2	106
8.27. Prueba 7.3	107
8.28. Prueba 7.4	107
8.29. Prueba 8.1	108
8.30. Prueba 8.2	108
8.31. Prueba 8.3	109
8.32. Prueba 8.4	109
8.33. Prueba 9.1	110
8.34. Prueba 9.2	110
8.35. Prueba 9.3	111
8.36. Prueba 9.4	111
8.37. Prueba 10.1	112
8.38. Prueba 10.2	112
8.39. Prueba 11.1	112
8.40. Prueba 11.2	113
8.41. Prueba 11.3	113
8.42. Prueba 12.1	113
8.43. Prueba 12.2	114
8.44. Prueba 12.3	114
8.45. Prueba 12.4	114
8.46. Prueba 13.1	115
8.47. Prueba 13.2	115
8.48. Prueba 14.1	115
8.49. Prueba 14.2	116
8.50. Prueba 14.3	116
8.51. Prueba 14.4	116
8.52. Prueba 15.1	117
8.53. Prueba 15.2	117
8.54. Prueba 15.3	117
8.55. Prueba 15.4	118
8.56. Prueba 15.5	118
8.57. Prueba 15.6	118
8.58. Prueba 15.7	118
8.59. Prueba 15.8	119
8.60. Prueba 15.9	119
8.61. Prueba 15.10	119
8.62. Prueba 15.11	119
8.63. Prueba 15.12	120
8.64. Prueba 15.13	120
8.65. Prueba 15.14	120
8.66. Prueba 15.15	120
8.67. Prueba 15.16	121
8.68. Prueba 15.17	121

8.69. Prueba 16.1	121
8.70. Prueba 16.2	122
8.71. Prueba 16.3	122
8.72. Prueba 16.4	122
8.73. Prueba 17.1	123
8.74. Prueba 17.2	123
8.75. Prueba 17.3	123
8.76. Prueba 17.4	124
8.77. Prueba 18.1	124
8.78. Prueba 18.2	124
8.79. Prueba 18.3	125
8.80. Prueba 18.4	125
8.81. Prueba 19.1	125
8.82. Prueba 19.2	126
8.83. Prueba 19.3	126
8.84. Prueba 19.4	126

Capítulo 1

Introducción

1.1. Contexto y motivación

Actualmente, los equipos de investigación viven rodeados de gastos diarios que requieren un control, con la finalidad de que los proyectos desarrollados por los mismos prosperen y puedan ser finalizados. Para realizar este tipo de seguimiento, lo ideal es realizar facturas o desgloses de los desembolsos que tendrán lugar, así como de los cobros que se esperan. Esta tarea puede resultar ardua dependiendo del contexto en el que nos encontremos, ya que no es lo mismo la factura de una compra por Internet que la factura de un extenso y complejo proyecto de investigación.

En el ámbito tecnológico esto no es distinto, ya que realizar una aproximación de lo que va a conllevar a nivel monetario el desarrollo de un proyecto software no es tarea sencilla.

El presente TFG está motivado por todo lo mencionado anteriormente, con el fin de desarrollar una herramienta que ayude a centralizar y facilitar dichas tareas para lograr el crecimiento y éxito de los proyectos, permitiendo a los equipos de investigación la generación de presupuestos en respuesta a solicitudes de transferencia tecnológica por parte del tejido empresarial.

A lo largo del presente documento se detallarán todas las fases que forman parte del desarrollo del proyecto, desde la elicitación de requisitos y análisis del mismo hasta la implementación total de la aplicación.

1.2. Objetivos

El objetivo del presente Trabajo de Fin de Grado es dotar a equipos de investigación de pequeño o mediano tamaño de una herramienta que les permita gestionar los presupuestos de sus proyectos, permitiéndoles almacenarlos todos en un mismo lugar y con una estructura estándar. Esto incluye la gestión de proveedores, costes de personal, costes recurrentes, viajes y dietas, y de costes indirectos.

Para conseguir todo esto se deberá implementar una parte de lógica de servidor que se encargue del almacenamiento de toda la información de los presupuestos, clientes, materiales utilizados, servicios contratados, tareas a llevar a cabo en los proyectos así como posibles viajes que sean necesarios para la realización de los mismos. Además, también se implementará una interfaz web que aproveche la lógica del servidor y permita a los usuarios realizar las tareas de forma simple y rápida. Por tanto, las tareas a desarrollar son las siguientes:

- **Desarrollo de un back-end:** Nos permitirá gestionar toda la información relacionada con los presupuestos, materiales, tareas, servicios, viajes, así como los parámetros genéricos que se utilizarán para calcular los costes de los proyectos, los perfiles profesionales con sus respectivos salarios, y finalmente las cuentas de los usuarios de la organización.

- **Desarrollo de un front-end:** La interfaz de usuario, que se apoyará en el servidor previamente implementado, utilizará la información del mismo para permitir a los usuarios de la organización realizar las tareas de gestión, creación, edición, visualización y eliminación de presupuestos y de las entidades asociadas a ellos.

1.3. Punto de partida

Actualmente este tipo de tareas se realizan utilizando software de hojas de cálculo, lo que limita significativamente la estructura que el usuario puede darle a los presupuestos, y dificulta su organización cuando se tiene un número elevado de los mismos. Otra alternativa es la utilización de software de pago similares disponibles en el mercado, pero dichos programas están orientados a la monitorización de proyectos en curso y a su planificación, por lo que no permite una generación de presupuestos propiamente dichos.

Por tanto, el objetivo de este trabajo será automatizar y facilitar el proceso de creación y almacenamiento para futura referencia de los presupuestos de proyectos de carácter tecnológico.

1.4. Estructura de esta memoria

La memoria del presente Trabajo de Fin de Grado está formada por una serie de capítulos, cada uno sobre un tema específico, a fin de organizar todo el trabajo realizado:

1. Estado del Arte: Dedicado a analizar los pros y los contras de soluciones ya existentes para el problema que se desea abordar.
2. Análisis de Requisitos: Dedicado a extraer del problema planteado las necesidades del sistema y consolidar unos requisitos sobre los que se realizará el trabajo.
3. Plan de Proyecto: Dedicado a elaborar la planificación del proceso de desarrollo del proyecto, detallando las fases por las que pasará y sus respectivos riesgos.
4. Modelo de Análisis: Detalla los diferentes casos de uso del sistema y el modelo de dominio extraídos de los requisitos.
5. Diseño: Detalla la arquitectura del sistema, diagramas de secuencia y bocetos sobre los que se basarán las vistas del sistema.
6. Implementación: Dedicado a detallar los procesos seguidos durante el desarrollo del sistema, así como la descripción de lo desarrollado.
7. Plan de Pruebas y Evaluación: Se planifican y elaboran una serie de pruebas a realizar para comprobar el correcto funcionamiento del sistema basándose en los requisitos.
8. Manual del Programador: Explica todo lo necesario para que un desarrollador pueda mantener o ampliar el sistema en un futuro.
9. Manual de Instalación: Detalla los pasos a seguir para la instalación del sistema en un nuevo entorno.
10. Manual de Usuario: Explica a un usuario todo lo necesario para poder utilizar el sistema desarrollado.
11. Conclusiones y Trabajo Futuro: Expone las conclusiones extraídas de la realización del trabajo y posibles vías de trabajo futuro sobre el sistema desarrollado.

1.5. Dominio del problema

En esta sección se explicará brevemente qué es un presupuesto, qué finalidad tiene y qué es necesario tener en cuenta para calcular uno, con el fin de facilitar la comprensión acerca del dominio del problema que motiva el presente trabajo.

1.5.1. ¿Qué es un presupuesto?

El presupuesto de un proyecto es una herramienta utilizada para estimar el coste total de un proyecto. Generalmente, el detalle de un presupuesto incluye una estimación detallada de todos los costes que deberán ser afrontados antes de que el proyecto se considere finalizado [18].

Ayuda al responsable de su gestión (normalmente un *project manager*) a realizar una estimación acerca del coste que ese proyecto va a suponer para su empresa u organización, así como las ganancias o pérdidas que implique.

La creación de un presupuesto suele tener lugar con antelación al inicio del proyecto, generalmente para poder realizar una oferta al cliente al que se le va a proveer el servicio, y una vez iniciado el proyecto, el presupuesto se utiliza como referencia para la monitorización del mismo.

La importancia de tener un presupuesto a la hora de realizar un proyecto tiene tres razones principales:

- Permite obtener la financiación necesaria para la realización del mismo.
- Permite llevar un control sobre el coste del proyecto.
- Favorece el aumento de los márgenes de beneficio de la empresa u organización que lleva a cabo el proyecto.

1.5.2. Cálculo de un presupuesto

A la hora de realizar el cálculo de un presupuesto se deben tener en cuenta dos cosas: el tipo de estimación que se quiere realizar y los tipos de costes que se deben de tener en cuenta. A continuación, se detallará brevemente en que consiste cada uno de esos aspectos.

Enfoques de estimación de costes de un proyecto

La estimación del coste total de un proyecto puede ser algo difícil de calcular inicialmente, ya que al abordar un problema nuevo suelen existir ciertas dificultades para estimar su tamaño y complejidad. Es por eso que es necesario escoger una técnica para realizar dicha estimación, lo que permitirá comenzar la creación del presupuesto con un cierto orden y estructura. Algunas de las técnicas de estimación [32] más utilizadas son las siguientes:

- **Bottom-up estimation (de abajo hacia arriba):** Consiste en realizar una estimación de cada una de las partes más pequeñas que compone el proyecto (tareas, objetivos o fases) y sumarlas para obtener su coste total. Se recomienda su uso cuando se conocen las implicaciones del proyecto con detalle.
- **Top-down estimation (de arriba hacia abajo):** Se parte de una cifra total, que es el presupuesto con el que contará el proyecto, y a partir de ahí se divide el trabajo en partes más pequeñas. Se recomienda su uso para proyectos que tienen un presupuesto fijo.
- **Estimación análoga:** Analizar los datos de proyectos previamente realizados que sean similares para decidir el coste.

- **Estimación paramétrica:** Utilizar datos y parámetros concretos de proyectos específicos y aplicarlos al proyecto actual, para tomar decisiones utilizando la relación estadística entre datos históricos y variables.
- **Estimación en tres puntos:** Realizar una media ponderada entre la estimación más pesimista, más optimista y la más probable.

1.5.3. Tipos de costes en un proyecto

Para realizar estimaciones detalladas acerca de los distintos costes que implica el proyecto, es necesario tener en cuenta que no todos son iguales. A la hora de clasificarlos, destacamos dos tipos de clasificaciones distintas: una en función de la **categoría**, y otra en función de si el coste es **directo** o **indirecto**. En primer lugar, en cuanto a la clasificación por categorías, destacamos algunas de las más comunes:

- **Mano de Obra:** Salarios de empleados a tiempo parcial y a tiempo completo.
- **Gastos de viajes:** Cualquier viaje que sea necesario realizar para la finalización del proyecto (conferencias, reuniones...) Para calcular el coste de los viajes hay que tener en cuenta ciertos parámetros que generalmente serán globales a toda la organización y que no se incluyen en el salario de los trabajadores, ya que no son una retribución por el trabajo realizado, si no que compensan los gastos que se producen al trabajar fuera del lugar habitual [17]:
 - **Dieta completa:** Compensa los gastos de desayuno, comida y cena del trabajador.
 - **Media dieta:** Compensa los gastos cuando se tiene que realizar o bien la comida o bien la cena fuera del centro habitual del trabajo. No tiene necesariamente por qué ser la mitad de la dieta completa.
 - **Alojamiento:** Compensa los gastos del alojamiento cuando un trabajador debe pernoctar fuera del domicilio habitual debido al trabajo.
 - **Coste del kilómetro:** Cuantía que compensa el gasto de combustible y el desgaste del vehículo cuando el trabajador tiene que utilizar su vehículo personal para un desplazamiento fuera del centro habitual de trabajo.
- **Materiales:** Cualquier objeto material que se necesite para llevar a cabo el proyecto (software, equipación, hardware...)
- **Servicios:** Incluye todo aquello no necesariamente tangible (es decir, que no se pueda considerar material) que sea necesario para el proyecto (contratación de servicios externos de telecomunicaciones, asesoramiento legal, consultores, investigación de mercado...)

Por otra parte, la clasificación de los costes también se puede realizar dependiendo de si estos son directos o indirectos [31]. Los **costes directos** son aquellos que se pueden atribuir de forma directa a un proyecto concreto, como por ejemplo los salarios de los empleados que trabajan en él, el alquiler de equipación, costes materiales, etc. Sin embargo, los **costes indirectos** son aquellos que no se pueden atribuir a un proyecto en concreto, ya que que benefician a todos los proyectos de la organización, como por ejemplo el alquiler y los gastos de la oficina, el personal de administración, el material de oficina...

Para incluir dichos costes en los presupuestos de los distintos proyectos de la organización, se suele realizar una estimación que proporciona un porcentaje de margen teórico para la recuperación de costes indirectos, y dicho porcentaje se aplica a los costes directos de cada uno de los proyectos. Generalmente, suele ser distinto dependiendo de la categoría del coste, y puede ser menor en las organizaciones de menor tamaño, ya que normalmente tienen menos costes de tipo indirecto a los que hacer frente (oficinas más pequeñas,

menor cantidad de personal...). Por ejemplo, si una organización decide tener un porcentaje de retorno de costes indirectos para materiales del 6 %, y en dicha organización existe un proyecto cuyos costes directos de materiales ascienden a 5.000€, a ese presupuesto se añadirá un coste indirecto de materiales de 300€.

1.5.4. Pasos para el cálculo de un presupuesto

El procedimiento de cálculo de un presupuesto puede variar entre organizaciones e incluso entre proyectos de la misma organización, ya que cada tipo de proyecto puede tener unas necesidades distintas, pero generalmente el proceso de cálculo y estimación sigue los siguientes pasos básicos:

1. **Dividir el proyecto en tareas y objetivos:** Decidir que tareas se deben llevar a cabo para el desarrollo del proyecto, y los objetivos que se quieren cumplir.
2. **Identificación de recursos:** Determinar que materiales y servicios se necesitarán para realizar las distintas tareas del proyecto, así como los viajes que sean necesarios.
3. **Suma de todos los costes:** Tras haber dividido el proyecto en cada una de las subtareas y haber determinado los materiales, se necesitará sumar todos ellos para obtener una serie de valores que permitirán tener una vista global del proyecto.
 - **Costes directos del proyecto:** Generalmente incluyen materiales, servicios, viajes, mano de obra...
 - **Recuperación de costes indirectos del proyecto**
 - **Coste total del proyecto:** Se calcula sumando el total de los costes directos e indirectos.
 - **Oferta:** Generalmente suele ser la financiación de la que se dispone para realizar el proyecto o la oferta que se realiza al cliente del mismo, dependiendo del contexto.
 - **Beneficio antes de impuestos:** La diferencia entre la oferta y el coste total que el proyecto supone a la empresa. Cabe destacar que generalmente las cifras contempladas en los presupuestos no incluyen los impuestos pertinentes, por lo que estos se deberán añadir posteriormente en el caso de que fuese necesario.

Capítulo 2

Estado del Arte

En el presente capítulo se van a analizar brevemente tres soluciones software encontradas que podrían resolver el problema que motiva el presente TFG, destacando sus ventajas e inconvenientes.

2.1. Software de Hojas de Cálculo

En primer lugar, cabe destacar una de las herramientas más utilizadas globalmente para la creación de presupuestos y para realizar contabilidad de costes de forma genérica. Algunos de los programas de hojas de cálculo más conocidos son: Microsoft Excel [15], Numbers [9], Google Sheets [5], Libre Office Calc [2], etc. A continuación se destacarán las ventajas y desventajas de la siguiente posible solución de forma genérica, sin centrarse en ningún programa de los previamente mencionados en concreto:

2.1.1. Ventajas

- Fácil de utilizar. Curva de aprendizaje no muy pronunciada, lo que permite que los usuarios se adapten rápidamente a su uso y puedan ser productivos antes.
- Gratuito.
- Extrema flexibilidad. Se pueden crear presupuestos sin seguir ningún patrón establecido.

2.1.2. Inconvenientes

- Extrema flexibilidad. Aunque esto se ha considerado también como una ventaja, la extrema flexibilidad del software implica que es difícil mantener una estructura cohesiva entre todos los presupuestos que pertenecen a una misma organización.
- No permite visualizar todos los presupuestos de forma global.
- No permite generar archivos con los presupuestos.
- Poco robusto. Los cálculos pueden ser fácilmente estropeados sin que los usuarios se den cuenta.

2.2. Easy Projects

Easy Projects [11] es un software de gestión de proyectos desarrollado por Logic Software. La cantidad de herramientas que ofrece esta solución se extiende más allá de la generación de presupuestos, pero en esta sección simplemente se analizará la utilidad asociada con los presupuestos, destacando sus ventajas e inconvenientes.

2.2.1. Ventajas

- Interfaz visual e intuitiva para los usuarios.
- Permite visualizar el presupuesto en formato PDF, incluyendo una lista de items y precios.
- Permite calcular el coste asociado a mano de obra, utilizando para ello las horas introducidas por los empleados en la herramienta asociada y multiplicándolas por su salario.
- Ofrece distintas opciones de facturación para los clientes.
- Permite introducir gastos realizados asociados al proyecto.
- Permite visualizar un resumen del presupuesto.

2.2.2. Inconvenientes

- Software de pago.
- No existe una diferenciación entre los distintos items que puede tener un presupuesto. Se consideran por igual materiales, servicios, licencias...
- Para calcular la mano de obra, requiere que los empleados introduzcan sus horas en el sistema, lo que impide realizar una estimación inicial de tareas y horas de trabajo necesarias para la realización del proyecto.
- La herramienta está más centrada en monitorizar el progreso de un proyecto, no tanto en la creación de presupuestos para el mismo.

2.3. Hubstaff

Hubstaff [1] es una herramienta de seguimiento del tiempo y gestión de productividad. Igual que sucede con la herramienta mencionada en la sección anterior, este software también incluye más funcionalidades adicionales a la gestión de presupuestos, por lo que sólo se analizarán las funcionalidades relevantes a la gestión de costes.

2.3.1. Ventajas

- Interfaz visual.
- Permite utilizar proyectos previos como guía a la hora de crear presupuestos.
- Permite al usuario poner límites de tiempo o de coste al presupuesto, notificándole cuando el presupuesto ha superado dichos límites establecidos.
- Permite visualizar todos los presupuestos asociados a los clientes de la organización.

2.3.2. Inconvenientes

- Software de pago.
- Relativamente complejo, con una curva de aprendizaje elevada.
- No permite asociar listas itemizadas de materiales o servicios a los presupuestos.

- No permite crear un presupuesto inicial basado en estimaciones. El usuario debe ir actualizando los datos de progreso a medida que el proyecto avanza para finalmente poder facturar al cliente una vez que el trabajo ha finalizado.

2.4. Conclusión

Al no encontrarse una solución que se adapte completamente a lo que buscamos, se ha decidido desarrollar una solución propia, tal y como se va a detallar en el presente documento.

Capítulo 3

Análisis de requisitos

3.1. Definición de los actores

Debido a la naturaleza del sistema a desarrollar, existirán tres actores en el mismo: el administrador, el jefe y el comercial.

3.2. Requisitos de usuario

Los requisitos proporcionados por el usuario son los siguientes:

- RU-01: Se deberá realizar un API REST que permita llevar a cabo todas las tareas relacionadas con la creación y gestión de presupuestos del sistema.
- RU-02: Se deberá realizar una interfaz web que aproveche el API REST mencionado anteriormente.

3.3. Requisitos funcionales

Las funciones que debe llevar a cabo la aplicación son las siguientes:

- RF-01: El sistema deberá permitir al usuario iniciar sesión.
- RF-02: El sistema deberá permitir al usuario cerrar sesión.
- RF-03: El sistema deberá permitir al administrador dar de alta a un usuario.
- RF-04: El sistema deberá permitir al administrador visualizar un listado de los usuarios del sistema.
- RF-05: El sistema deberá permitir al administrador dar de baja a un usuario.
- RF-06: El sistema deberá permitir al jefe visualizar los parámetros del sistema.
- RF-07: El sistema deberá permitir al jefe crear nuevos parámetros para el sistema.
- RF-08: El sistema deberá permitir al jefe visualizar un listado de los perfiles profesionales.
- RF-09: El sistema deberá permitir al jefe añadir nuevos perfiles profesionales.
- RF-10: El sistema deberá permitir al jefe modificar los perfiles profesionales.
- RF-11: El sistema deberá permitir al jefe eliminar perfiles profesionales.
- RF-12: El sistema deberá permitir al jefe visualizar un listado de los presupuestos aprobados y pendientes de aprobación.

- RF-13: El sistema deberá permitir al jefe aprobar un presupuesto.
- RF-14: El sistema deberá permitir al jefe rechazar un presupuesto.
- RF-15: El sistema deberá permitir al jefe obtener un archivo de un presupuesto.
- RF-16: El sistema deberá permitir al jefe ordenar el listado de presupuestos por nombre.
- RF-17: El sistema deberá permitir al jefe ordenar el listado de presupuestos por estado.
- RF-18: El sistema deberá permitir al jefe ordenar el listado de presupuestos por autor del presupuesto.
- RF-19: El sistema deberá permitir al comercial crear un presupuesto.
- RF-20: El sistema deberá permitir al comercial modificar un presupuesto en estado editable.
- RF-21: El sistema deberá permitir al comercial cambiar el estado de un presupuesto a pendiente de aprobación.
- RF-22: El sistema deberá permitir al comercial visualizar una lista del histórico de materiales.
- RF-23: El sistema deberá permitir al comercial visualizar un historial de los presupuestos en los que ha sido utilizado un artículo.
- RF-24: El sistema deberá permitir al comercial añadir un artículo al histórico de materiales.
- RF-25: El sistema deberá permitir al comercial eliminar un artículo del histórico de materiales.
- RF-26: El sistema deberá permitir al comercial visualizar una lista de materiales asociados al presupuesto.
- RF-27: El sistema deberá permitir al comercial añadir un material del histórico de materiales al presupuesto.
- RF-28: El sistema deberá permitir al comercial eliminar un material del presupuesto.
- RF-29: El sistema deberá permitir al comercial editar un material del presupuesto.
- RF-30: El sistema deberá permitir al comercial visualizar una lista de viajes asociados al presupuesto.
- RF-31: El sistema deberá permitir al comercial añadir un viaje al presupuesto.
- RF-32: El sistema deberá permitir al comercial editar un viaje del presupuesto.
- RF-33: El sistema deberá permitir al comercial eliminar un viaje del presupuesto.
- RF-34: El sistema deberá permitir al comercial visualizar una lista de tareas asociadas al presupuesto.
- RF-35: El sistema deberá permitir al comercial añadir una tarea al presupuesto.
- RF-36: El sistema deberá permitir al comercial eliminar una tarea del presupuesto.
- RF-37: El sistema deberá permitir al comercial editar una tarea del presupuesto.
- RF-38: El sistema deberá permitir al comercial visualizar una lista de servicios asociados al presupuesto.
- RF-39: El sistema deberá permitir al comercial añadir un servicio al presupuesto.

- RF-40: El sistema deberá permitir al comercial eliminar un servicio del presupuesto.
- RF-41: El sistema deberá permitir al comercial editar un servicio del presupuesto.
- RF-42: El sistema deberá permitir al comercial visualizar un listado de todos los presupuestos del sistema.
- RF-43: El sistema deberá permitir al comercial duplicar un presupuesto.
- RF-44: El sistema deberá permitir al comercial eliminar un presupuesto editable.
- RF-45: El sistema deberá permitir al comercial obtener un archivo de un presupuesto pendiente de aprobación.
- RF-46: El sistema deberá permitir al comercial obtener un archivo de un presupuesto aprobado.
- RF-47: El sistema deberá permitir al comercial ordenar el listado de presupuestos por nombre.
- RF-48: El sistema deberá permitir al comercial ordenar el listado de presupuestos por estado.
- RF-49: El sistema deberá permitir al comercial ordenar el listado de presupuestos por autor del presupuesto.

3.4. Requisitos no funcionales

Las características de funcionamiento del sistema son las siguientes:

- RNF-01: Los archivos de los presupuestos tendrán formato PDF.

3.5. Requisitos de información

Los datos necesarios para el funcionamiento de la aplicación son los siguientes:

- RI-01: Para cada usuario del sistema se almacenará:
 1. Nombre.
 2. Usuario.
 3. Contraseña.
 4. Correo Electrónico.
 5. Rol en el Sistema.
- RI-02: Para cada presupuesto se almacenará:
 1. Nombre.
 2. Ingresos previstos en el proyecto.
 3. Fecha del presupuesto.
 4. Estado del presupuesto.
 5. Timestamp de la aprobación del presupuesto.
 6. Nombre del cliente al que va dirigido el presupuesto.
 7. CIF del cliente al que va dirigido el presupuesto.
 8. Dirección del cliente al que va dirigido el presupuesto.

9. Persona de contacto en la organización del cliente al que va dirigido el presupuesto.

- RI-03: Para cada artículo almacenado en el histórico de materiales del sistema se almacenará:
 1. Nombre.
 2. Proveedor.
- RI-04: Para cada material utilizado en un presupuesto se almacenará:
 1. Precio.
 2. Cantidad.
- RI-05: Para cada servicio utilizado en un presupuesto se almacenará:
 1. Concepto.
 2. Precio.
 3. Cantidad.
 4. Tiempo.
- RI-06: Para cada viaje contemplado en un presupuesto se almacenará:
 1. Concepto.
 2. Kilómetros.
 3. Dias completos que va a durar el viaje.
 4. Dias parciales que va a durar el viaje.
 5. Coste del transporte por tren.
 6. Coste del transporte por avión.
 7. Costes adicionales del viaje.
 8. Cantidad.
- RI-07: Para cada tarea contemplada en un presupuesto se almacenará:
 1. Concepto.
 2. Unidades (Meses, semanas, días...) que se tardará en llevar a cabo la tarea.
 3. Horas por unidad que se tardará en realizar la tarea.
 4. Incertidumbre: Dependiendo del perfil profesional al que se le asigne la tarea, la incertidumbre determinará las posibilidades de que la tarea se alargue más de lo previsto.
- RI-08: Para cada conjunto de parámetros del sistema se almacenará:
 1. Coste de una Dieta Completa Nacional.
 2. Coste de Media Dieta Nacional.
 3. Coste de un kilómetro.
 4. Coste de un hotel.
 5. Porcentaje de margen teórico para la recuperación de costes indirectos por viajes.
 6. Porcentaje de margen teórico para la recuperación de costes indirectos por materiales y servicios.
 7. Porcentaje de margen teórico para la recuperación de costes indirectos por mano de obra.
- RI-09: Para cada perfil profesional o cargo considerado en el sistema se almacenará:
 1. Título.
 2. Salario por horas.

3.6. Reglas de negocio

Las características de dominio en las que se encuadra el sistema son las siguientes:

- RN-01: Sólo se podrá obtener un archivo de los presupuestos que estén pendientes de ser aprobados y de los presupuestos aprobados.
- RN-02: Sólo se podrá eliminar los presupuestos editables.
- RN-03: Los presupuestos en estado editable siempre tendrán que utilizar los parámetros más recientes registrados en el sistema.

3.7. Estados de un presupuesto

Durante su ciclo de vida, un presupuesto puede pasar por distintos estados, que influirán en las acciones que se pueden llevar a cabo sobre dicho presupuesto. A continuación se mostrará en el siguiente diagrama el flujo entre los distintos estados y se detallará que se puede realizar en cada uno de ellos.

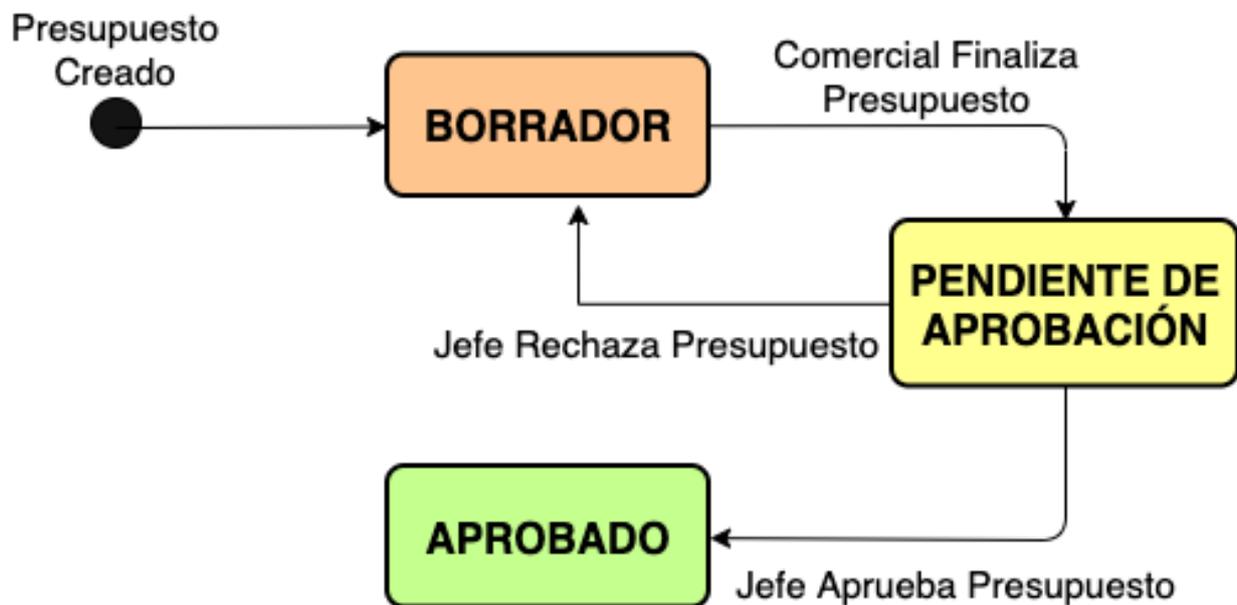


Figura 3.1: Estados de un presupuesto

- **Borrador:** Estado inicial de todos los presupuestos al ser creados. Permite a los usuarios con rol de comercial editar los presupuestos y eliminarlos.
- **Pendiente de Aprobación:** Para que un presupuesto pase a este estado, el comercial deberá finalizarlo. Permite a los usuarios con rol de comercial o de jefe visualizar un archivo del presupuesto, y a los usuarios con rol de jefe aprobarlo o rechazarlo. Si el presupuesto es rechazado, volverá a estado de borrador.
- **Aprobado:** Para que un presupuesto pase a este estado, el jefe deberá aprobarlo. Permite que los usuarios con rol de comercial o jefe visualicen un archivo del presupuesto.

Capítulo 4

Plan de proyecto

4.1. Resumen del proyecto

En este apartado se detallará el plan de proyecto seguido para llevar a cabo el desarrollo del sistema planteado en este TFG.

4.1.1. Propósito, alcance y objetivos

El objetivo del proyecto es el desarrollo de un sistema completo de contabilidad de costes que permita a sus usuarios la creación y gestión de presupuestos para sus proyectos, así como una forma organizada de almacenarlos. Las características que se incluirán son: capacidad por parte del administrador de gestionar el alta y baja de los usuarios del sistema, capacidad por parte de los jefes de la organización de modificar ciertos parámetros de los presupuestos, así como aprobar o denegar los presupuestos de los comerciales. También permitirá que dichos comerciales puedan crear, editar, duplicar, visualizar y eliminar los presupuestos de la organización. Las funciones serán completamente accesibles a través de la interfaz web desarrollada para el sistema.

4.1.2. Definiciones y acrónimos

A continuación se enumeran un conjunto de definiciones y acrónimos que serán usados a lo largo del presente documento.

- REST: REpresentational State Transfer, transferencia de estado representacional.
- CRUD: Create, Read, Update and Delete - Crear, Leer, Actualizar y Borrar.
- API: Application Programming Interface, Interfaz de programación de aplicaciones. Conjunto de funciones, subrutinas y procedimientos que se utilizan para el desarrollo e integración del software de una aplicación.
- Front-end: Desarrollo de una interfaz web que permite al usuario interactuar con los datos que se manejan en una aplicación.
- Back-end: Parte del desarrollo de una aplicación que corresponde al servidor, donde se procesa y se almacenan los datos de la misma.
- Factor Incertidumbre (de una tarea): Dependiendo del perfil profesional al que se le asigne una tarea de un proyecto, este parámetro determina la probabilidad de que dicha tarea se alargue más de lo previsto.

4.1.3. Método escogido para el proyecto

Para llevar a cabo el desarrollo del proyecto se ha decidido emplear un desarrollo basado en cascada, ya que los requisitos a los que está sujeto el sistema son claros y no admiten mucha interpretación, por lo que es improbable que durante el transcurso del proyecto dichos requisitos cambien.

Este método de desarrollo se caracteriza por ser secuencial, constando de un conjunto de etapas que se ejecutan una tras otra. En el caso de este proyecto, el desarrollo se dividirá en 5 fases: análisis, diseño, implementación, pruebas y mantenimiento. Tal y como indica el desarrollo en cascada, cada fase deberá ser completada antes de poder comenzar con la siguiente.

4.1.4. Evolución del plan

El plan de desarrollo de proyecto debe detallar cada una de las fases por las que pasará el proceso de desarrollo del sistema, y además detallar como se realizará la gestión de su evolución. Por tanto, para obtener un plan completo debemos considerar al menos:

- Especificación de los objetivos del proceso.
- Estimación de costes materiales, temporales y de servicios necesarios para el desarrollo del sistema.
- Especificación de las tareas a realizar y su extensión en el tiempo, así como determinar las distintas fases que atravesará el proyecto.
- Recursos necesarios para realizar cada una de las tareas detalladas.
- Estudio de los posibles riesgos que pueden surgir, un plan de protección que indique como podemos evitarlos y un plan de contingencia asociado que indique como actuar ante cada uno de ellos en caso de que sucedan.

4.1.5. Ciclo de vida del proyecto

Como se ha mencionado previamente, el ciclo de vida de este proyecto constará de cinco fases completamente diferenciadas entre sí, que requerirán ciertos artefactos de entrada al comienzo de la fase y generarán otros artefactos de salida. Los artefactos de salida de una fase serán utilizados como artefactos de entrada de la siguiente fase, y así sucesivamente a lo largo del desarrollo del proyecto.

A continuación se detalla brevemente los objetivos a cumplir en cada una de las fases del proyecto:

- Fase de análisis: Establecer el alcance y el ámbito del proyecto, casos críticos y estimación temporal.
- Fase de diseño: Analizar el dominio del problema, establecer una arquitectura base apropiada y desarrollar un plan de proyecto apropiado.
- Fase de implementación: Minimizar costes de desarrollo en la medida de lo posible pero obteniendo aun así una calidad adecuada en el producto final.
- Fase de pruebas: Comprobar que el producto desarrollado cumple con los requisitos establecidos en fases previas de forma satisfactoria.
- Fase de mantenimiento: Conseguir que el usuario sea capaz de utilizar el producto de forma correcta y alcanzar un resultado final rápido, eficiente y con altos niveles de usabilidad que sea aceptado por el cliente.

4.2. Gestión del proyecto

4.2.1. Plan de puesta en marcha

Para comenzar el desarrollo del proyecto adecuadamente se realizará una estimación de cada una de las fases, teniendo en cuenta la experiencia obtenida en la realización de proyectos previos similares a este. El desarrollo se realizará de forma individual, por lo que no se tendrá en cuenta la gestión de personal para la realización del proyecto.

Antes de empezar el proceso de desarrollo, será necesario tener conocimiento de ciertos aspectos clave para su elaboración. A continuación se muestra una lista de aquellas tecnologías o conceptos sobre los que se deberá tener conocimiento previo para el proyecto, divididos según los apartados que se van a desarrollar dentro del mismo, a fin de que se pueda entender con una mayor claridad:

Base de datos

- Conocimiento sobre bases de datos relacionales, incluyendo tanto para su uso como para el diseño de una base de datos desde cero.

Back-end

- Conocimiento sobre Javascript.
- Conocimiento sobre la estructura y el funcionamiento del framework Node.js.

Front-end

- Conocimiento sobre HTML, CSS y Javascript.
- Conocimiento sobre la estructura y el funcionamiento del framework React.js.

Despliegue

- Conocimiento sobre Docker.

4.2.2. Plan de trabajo

En este apartado se detalla cada una de las fases del proyecto y las actividades por las que estarán compuestas.

Fase de análisis

ID: 01 Inicio de la fase de Análisis	
Predecesoras:	-
Duración:	-
	-

Tabla 4.1: Actividad 01

ID: 02 Comprensión del dominio del problema	
Predecesoras:	01
Duración:	2 días
	Aprendizaje de conceptos económicos relacionados con la creación de presupuestos para proyectos de empresas tecnológicas.

Tabla 4.2: Actividad 02

ID: 03 Elicitación de requisitos	
Predecesoras:	01
Duración:	3 días
	Obtención de los requisitos funcionales, no funcionales y de información.

Tabla 4.3: Actividad 03

ID: 04 Definición de actividades	
Predecesoras:	02, 03
Duración:	2 días
	Definición de las actividades necesarias para el desarrollo del proyecto y su duración.

Tabla 4.4: Actividad 04

ID: 05 Estudio de los riesgos	
Predecesoras:	04
Duración:	2 días
	Análisis de los riesgos que presenta el proyecto y creación de un plan de contención para cada uno de ellos.

Tabla 4.5: Actividad 05

ID: 06 Elaboración del calendario	
Predecesoras:	05
Duración:	1 día
	Elaboración de una planificación temporal para la ejecución de las distintas fases del proyecto.

Tabla 4.6: Actividad 06

ID: 07 Aprendizaje de Node.js
Predecesoras: 06
Duración: 7 días
Aprendizaje del funcionamiento y arquitectura del framework Node.js, que se utilizará para el desarrollo del back-end.

Tabla 4.7: Actividad 07

ID: 08 Aprendizaje de React.js
Predecesoras: 06
Duración: 7 días
Aprendizaje del funcionamiento y arquitectura del framework React.js, que se utilizará para el desarrollo del front-end.

Tabla 4.8: Actividad 08

ID: 09 Aprendizaje de Docker
Predecesoras: 06
Duración: 9 días
Aprendizaje de la tecnología Docker, que será utilizada para el despliegue del sistema desarrollado.

Tabla 4.9: Actividad 09

ID: 10 Elaboración del plan de proyecto
Predecesoras: 07, 08, 09
Duración: 3 días
Elaboración del documento que describe el plan de proyecto, incluyendo todas las actividades y fases, su planificación temporal y la gestión de los riesgos asociados a las mismas.

Tabla 4.10: Actividad 10

ID: 11 Fin de la fase de Análisis
Predecesoras: 10
Duración: -
-

Tabla 4.11: Actividad 11

Fase de Diseño

ID: 12 Inicio de la fase de Diseño
Predecesoras: 11
Duración: -
-

Tabla 4.12: Actividad 12

ID: 13 Elaboración de los casos de uso
Predecesoras: 12
Duración: 3 días
Elaboración del diagrama de casos de uso del sistema y especificación de cada uno de ellos a partir de los requisitos concretados en la fase de análisis.

Tabla 4.13: Actividad 13

ID: 14	Elaboración del modelo de dominio
Predecesoras:	12
Duración:	3 días
Elaboración del modelo de dominio de las entidades del sistema que serán utilizadas, únicamente indicando los atributos de las mismas.	

Tabla 4.14: Actividad 14

ID: 15	Elaboración del despliegue del sistema
Predecesoras:	12
Duración:	6 días
Elaboración del diagrama de despliegue del sistema, junto con las dependencias de cada apartado del conjunto.	

Tabla 4.15: Actividad 15

ID: 16	Diseño de la arquitectura del sistema
Predecesoras:	13, 14, 15
Duración:	6 días
Elaboración de un diagrama que permita definir la arquitectura que tendrá el sistema.	

Tabla 4.16: Actividad 16

ID: 17	Diseño de los bocetos del sistema
Predecesoras:	16
Duración:	2 días
Elaboración de bocetos que muestren las vistas que será necesario implementar en el sistema en base a los casos de uso previamente definidos.	

Tabla 4.17: Actividad 17

ID: 18	Elaboración de diagramas de secuencia
Predecesoras:	16
Duración:	6 días
Elaboración de diagramas de secuencia tanto para el front-end como para el back-end, que detallen la realización de los casos de uso previamente detallados.	

Tabla 4.18: Actividad 18

ID: 19	Fin de la fase de Diseño
Predecesoras:	17, 18
Duración:	-
-	

Tabla 4.19: Actividad 19

Fase de Implementación

ID: 20	Inicio de la fase de Implementación
Predecesoras:	19
Duración:	-
-	

Tabla 4.20: Actividad 20

ID: 21 Preparación del entorno de desarrollo
Predecesoras: 20
Duración: 1 día
Se instalarán todas las dependencias necesarias para poder llevar a cabo el desarrollo del back-end, el front-end y el despliegue de todo el sistema.

Tabla 4.21: Actividad 21

ID: 22 Desarrollo del back-end
Predecesoras: 21
Duración: 30 días
Se llevará a cabo la implementación del servicio en base al análisis del sistema previamente realizado.

Tabla 4.22: Actividad 22

ID: 23 Desarrollo del front-end
Predecesoras: 22
Duración: 30 días
Se llevará a cabo la implementación de una interfaz web que utilice el servicio implementado anteriormente.

Tabla 4.23: Actividad 23

ID: 24 Fin de la fase de Implementación
Predecesoras: 22, 23
Duración: -
-

Tabla 4.24: Actividad 24

Fase de Pruebas

ID: 25 Inicio de la fase de Pruebas
Predecesoras: 24
Duración: -
-

Tabla 4.25: Actividad 25

ID: 26 Realización de pruebas del sistema
Predecesoras: 25
Duración: 4 días
Realización de las pruebas necesarias para comprobar que el sistema cumple de forma satisfactoria los requisitos establecidos.

Tabla 4.26: Actividad 26

ID: 27 Fin de la fase de Pruebas
Predecesoras: 26
Duración: -
-

Tabla 4.27: Actividad 27

Fase de Mantenimiento

ID: 28 Inicio de la fase de Mantenimiento	
Predecesoras:	27
Duración:	-
-	

Tabla 4.28: Actividad 28

ID: 29 Despliegue del sistema	
Predecesoras:	28
Duración:	6 días
Despliegue del sistema desarrollado.	

Tabla 4.29: Actividad 29

ID: 30 Elaboración del manual de usuario	
Predecesoras:	29
Duración:	2 días
Elaboración de un documento que incluya de forma clara y concisa la información necesaria para el correcto uso del sistema.	

Tabla 4.30: Actividad 30

ID: 31 Elaboración del manual de instalación	
Predecesoras:	30
Duración:	2 días
Elaboración de un documento que incluya los pasos a seguir y los elementos necesarios para el despliegue del sistema por parte de un administrador.	

Tabla 4.31: Actividad 31

ID: 32 Elaboración del manual del programador	
Predecesoras:	31
Duración:	2 días
Elaboración de un documento que cuente con la información necesaria para que otro desarrollador pueda continuar con el desarrollo del sistema.	

Tabla 4.32: Actividad 32

ID: 33 Fin de la fase de Mantenimiento	
Predecesoras:	32
Duración:	-
-	

Tabla 4.33: Actividad 33

Diagramas de Gantt

En este apartado se mostrarán los diagramas de Gantt del plan de trabajo, que nos permitirán ver las diferentes fases por las que ha pasado el proyecto.

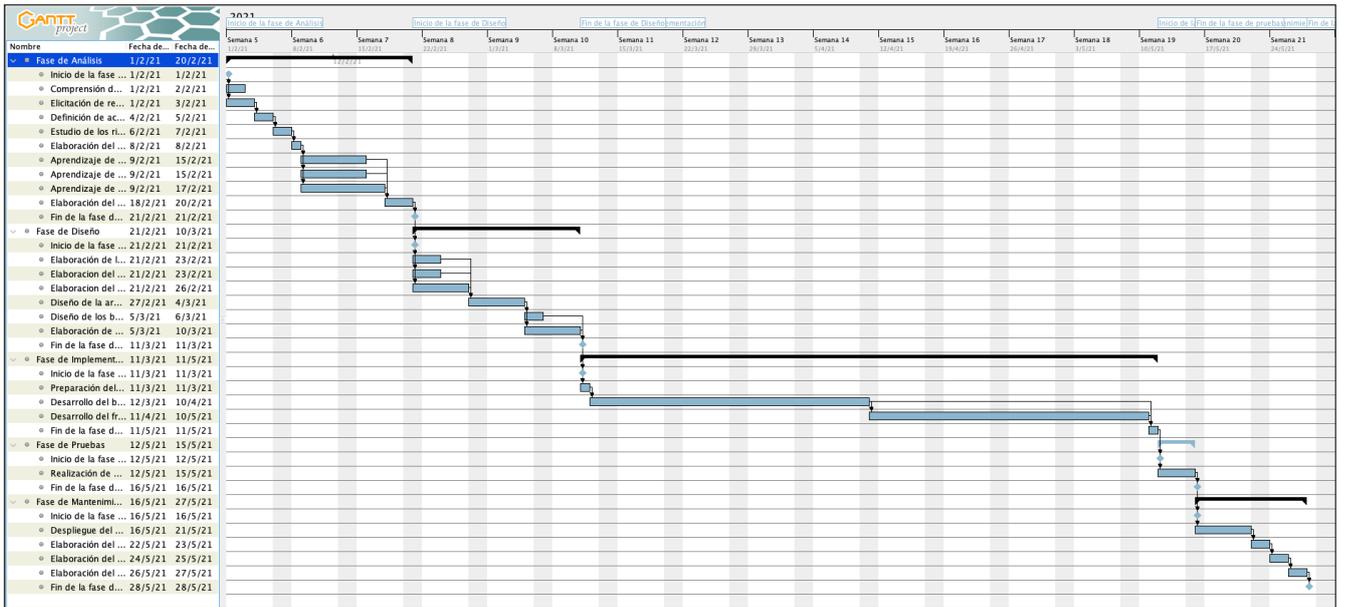


Figura 4.1: Vista general del plan de trabajo

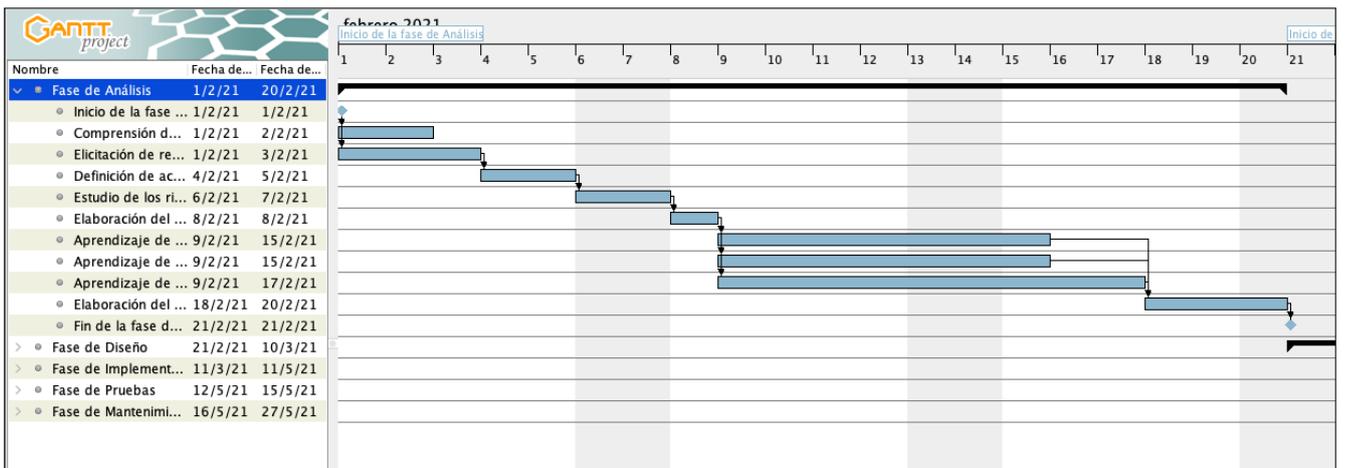


Figura 4.2: Fase de análisis

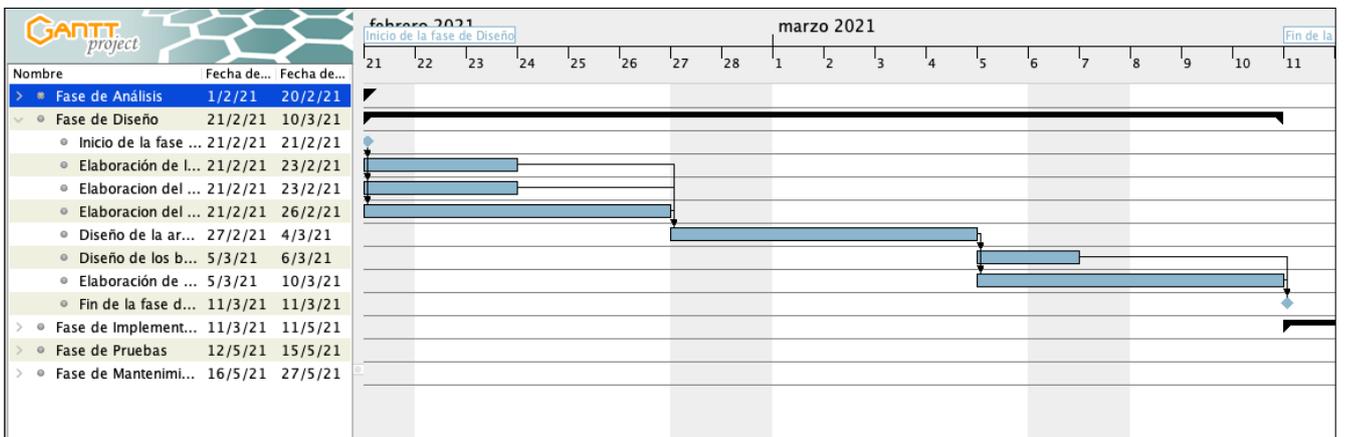


Figura 4.3: Fase de diseño

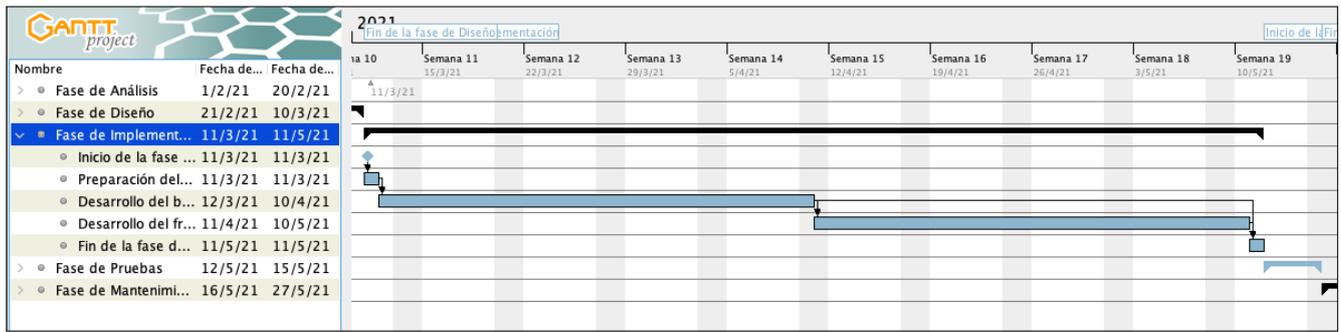


Figura 4.4: Fase de implementación

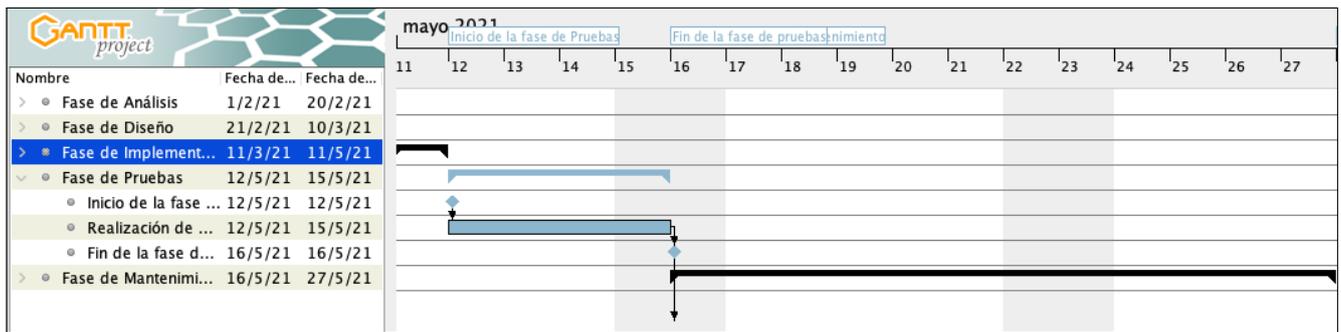


Figura 4.5: Fase de pruebas

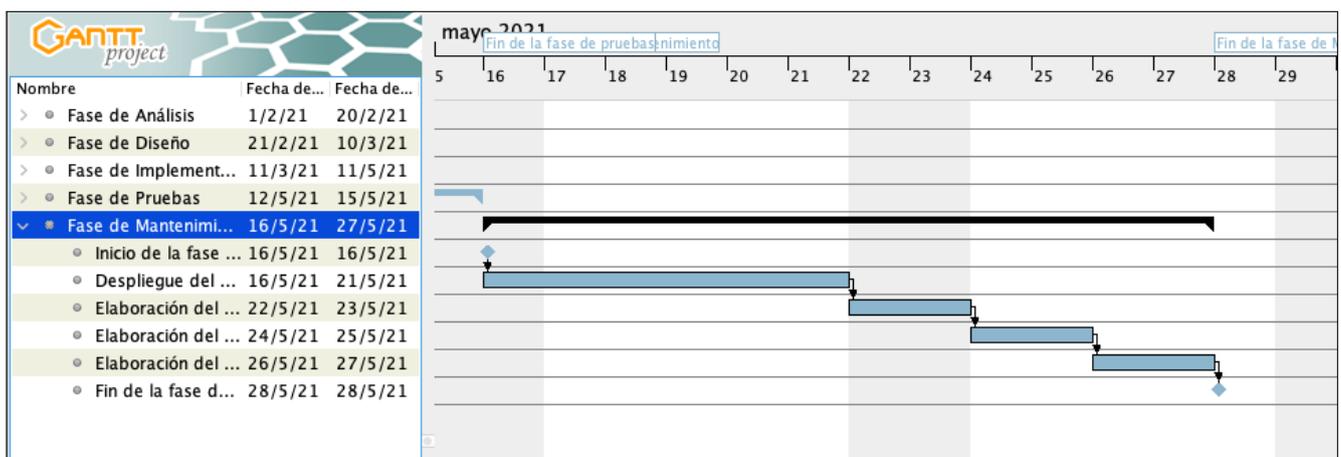


Figura 4.6: Fase de mantenimiento

4.2.3. Plan de control

Para asegurar un funcionamiento adecuado del proceso de desarrollo, se llevará a cabo un plan de control para el que se tendrán en cuenta tanto los objetivos iniciales como los métodos de realización de los mismos, de forma que estos puedan ser adaptados en función de las necesidades y el contexto de cada fase por la que va a ir pasando el proyecto.

Para llevar a cabo esta tarea se realizará una lista de requisitos previamente analizados, ordenados por su prioridad de implementación. Esto facilitará la organización del proyecto en el caso de que en el futuro algún requisito deba ser modificado, o si se debe añadir algún requisito nuevo al sistema, en cuyo caso, podrá hacerse de manera sencilla introduciéndolo en esta lista teniendo en cuenta su prioridad y realizando una replanificación temporal en el caso de que fuera necesario.

Respecto a la planificación y control temporal de todo el proceso, al final de cada actividad planteada se llevarán a cabo un análisis y un seguimiento de la actividad finalizada, para poder verificar que se adapta a la planificación temporal realizada o que por el contrario, se sale de la planificación prevista y se requiere una adaptación de dicho plan. Éste método permitirá detectar posibles contratiempos de forma temprana, permitiendo dividir el proyecto en fragmentos de tiempo más reducidos que serán más sencillos de corregir, ayudando a que el proyecto pueda terminar en la fecha estipulada.

4.2.4. Plan de gestión de riesgos

R01 - Enfermedad

Dada la situación de crisis sanitaria en las que se ha desarrollado el proyecto y que la elaboración del mismo cuenta con un sólo desarrollador, se ha debido tener en cuenta esta posibilidad, ya que imposibilitaría el progreso adecuado del proyecto.

Impacto Leve

Probabilidad Baja

Exposición Alta

Plan de Protección No se puede elaborar un plan de contención para este suceso.

Plan de Contingencia Estudiar y replanificar el plan de proyecto en función del tiempo perdido.

R02 - Desconocimiento de las herramientas a utilizar para el desarrollo

Falta de conocimiento de los frameworks que van a ser utilizados para el desarrollo del front-end y del back-end de la aplicación

Impacto Crítico

Probabilidad Alta

Exposición Media

Plan de Protección Reservar un período de tiempo en el plan de proyecto para poder realizar un aprendizaje previo de las herramientas.

Plan de Contingencia En el caso de que se produjera algún retraso temporal, ajustar la planificación del proyecto como se vea pertinente.

R03 - Desconocimiento de las herramientas a utilizar para el despliegue

Falta de conocimiento de las herramientas que van a ser utilizadas para el despliegue del sistema desarrollado.

Impacto Crítico

Probabilidad Alta

Exposición Media

Plan de Protección Reservar un período de tiempo en el plan de proyecto para poder realizar un aprendizaje previo de las herramientas.

Plan de Contingencia En el caso de que se produjera algún retraso temporal, ajustar la planificación del proyecto como se vea pertinente.

R04 - Retraso en alguna de las fases

Retraso temporal de alguna de las fases anteriormente planificadas.

Impacto Crítico

Probabilidad Media

Exposición Media

Plan de Protección Ajustarse a la planificación planteada lo máximo posible.

Plan de Contingencia En el caso de que se produjera algún retraso temporal, ajustar la planificación del proyecto como se vea pertinente.

R05 - Falta de alguna funcionalidad

Que el sistema carezca de alguna funcionalidad que debería haberse cubierto.

Impacto Crítico

Probabilidad Baja

Exposición Alta

Plan de Protección Ajustarse a los requisitos que fueron elicitados durante la fase de análisis para llevar a cabo la implementación.

Plan de Contingencia En el caso de que alguna funcionalidad faltase, se añadiría como fuese pertinente.

R06 - Diseño erróneo

El diseño no cumple con los requisitos del sistema.

Impacto Crítico

Probabilidad Baja

Exposición Baja

Plan de Protección Realizar el diseño ajustándose a el análisis del sistema realizado en la fase de análisis.

Plan de Contingencia En el caso de que el diseño fuese incorrecto, rehacerlo y ajustar la planificación para compensar el tiempo utilizado.

R07 - Cambio de los requisitos

Durante el desarrollo se añaden requisitos nuevos o se quita alguno de los existentes.

Impacto Crítico

Probabilidad Baja

Exposición Media

Plan de Protección Realizar un estudio de los requisitos con el fin de cubrir una gran mayoría de las funciones necesarias para el sistema.

Plan de Contingencia En el caso de que apareciera o se modificara algún requisito, analizarlo, incorporar los cambios y realizar una reevaluación de los riesgos del proyecto, así como una reestructuración del plan del mismo.

4.2.5. Seguimiento del proyecto

Teniendo en cuenta la planificación realizada a lo largo de este capítulo, no ha surgido ninguna situación excepcional que haya supuesto un retraso en el desarrollo del proyecto, por lo que el plan se ha podido cumplir en su totalidad y la ejecución del mismo ha transcurrido con normalidad.

4.3. Presupuesto

En base a la planificación temporal anteriormente descrita, la duración del proyecto será de 4 meses. A continuación se muestra una tabla que cuenta con la información necesaria para el planteamiento del presupuesto total simulado del proyecto.

Presupuesto del Proyecto	
Duración del proyecto	4 meses.
Salario según convenio de un Ingeniero Informático en España	25.600€.
Costes sociales	8.450€.
Beneficio Industrial (7%)	794,5€.
Total	12.144,5€.

Tabla 4.34: Presupuesto (IVA no incluido)

Dadas las características del proyecto, no se incurrirá en ningún gasto relacionado con hardware u otro tipo de materiales para el desarrollo del mismo, así como tampoco será necesario el pago de ningún tipo de licencia o servicio.

Además, sólo se han tenido en cuenta los costes directos de desarrollo y el beneficio industrial. No se han tenido en cuenta los costes indirectos debido a que su porcentaje de repercusión depende de la estructura de costes de la empresa concreta en la que el trabajador desarrolle su labor.

Tras finalizar el plan de desarrollo del proyecto, nos centraremos en el análisis del problema planteado con una mayor profundidad en el capítulo siguiente.

Capítulo 5

Modelo de análisis

5.1. Casos de uso del sistema

A continuación se detallarán los casos de uso del sistema a desarrollar.

5.1.1. Diagrama general de casos de uso

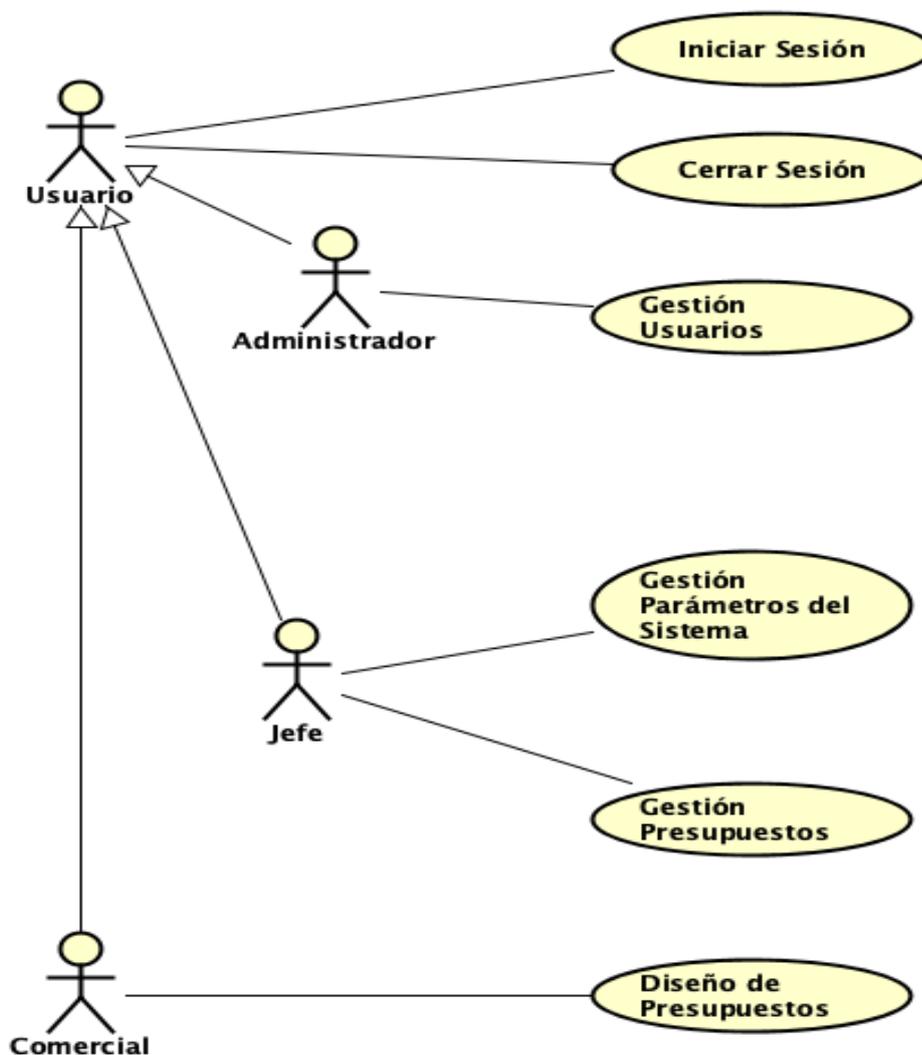


Figura 5.1: Diagrama general de casos de uso

5.1.2. Caso de uso 1: Iniciar sesión

CU-1	Iniciar sesión
Actor	Usuario
Descripción	El sistema deberá permitir al usuario iniciar sesión.
Precondición	
Secuencia Normal	
Paso	Acción
1	El actor introduce usuario y contraseña.
2	El sistema comprueba que las credenciales introducidas son correctas.
3	El actor inicia sesión correctamente en el sistema. El caso de uso termina.
Post-condición	El actor consigue acceder al sistema.
Excepciones	
Variación	Acción
1b, 2b, 3b	El actor sale del sistema y el caso de uso queda sin efecto.
2c	El sistema determina que las credenciales no son correctas, se lo notifica al usuario y el caso de uso continúa en el paso 1.

Tabla 5.1: Caso de Uso 1 - Iniciar sesión

5.1.3. Caso de uso 2: Cerrar sesión

CU-2	Cerrar sesión
Actor	Usuario
Descripción	El sistema deberá permitir al usuario cerrar sesión.
Precondición	El actor ha iniciado sesión.
Secuencia Normal	
Paso	Acción
1	El actor pulsa el botón de cerrar sesión.
2	El sistema muestra un mensaje informando de la acción que se va a realizar y pidiendo confirmación.
3	El actor pulsa el botón de cerrar sesión. El caso de uso termina.
Post-condición	El actor ha cerrado sesión en el sistema.
Excepciones	
Variación	Acción
1b, 2b	El actor sale del sistema y el caso de uso queda sin efecto.
3b	El actor pulsa el botón de cancelar y el caso de uso queda sin efecto.

Tabla 5.2: Caso de Uso 2 - Cerrar sesión

5.1.4. Caso de uso 3: Gestión de Usuarios

Los casos de uso de la gestión de usuarios permiten al administrador visualizar los usuarios del sistema, registrar usuarios nuevos y eliminar usuarios existentes.

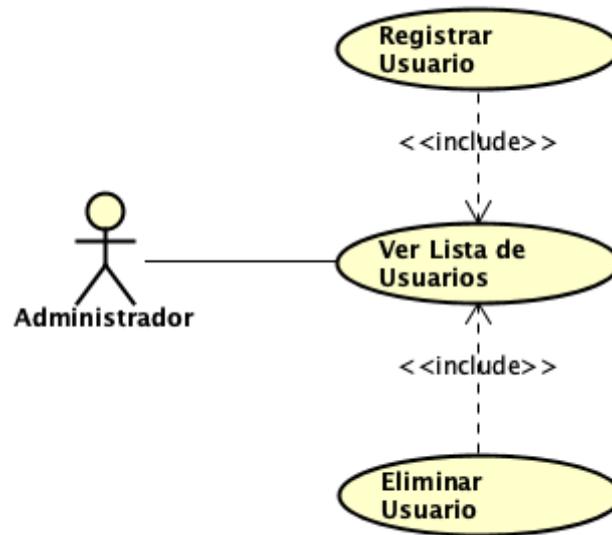


Figura 5.2: Diagrama en detalle de gestión de usuarios

CU-3.1	Ver lista de usuarios
Actor	Administrador
Descripción	El sistema deberá permitir al administrador visualizar un listado de todos los usuarios del sistema.
Precondición	El actor ha iniciado sesión en el sistema.
Secuencia Normal	
Paso	Acción
1	El actor entra en la página principal.
2	El sistema muestra un listado con los usuarios del sistema. El caso de uso termina.
Post-condición	
Excepciones	
Variación	Acción
1b, 2b	El actor sale del sistema. El caso de uso queda sin efecto.
2c	El sistema no contiene usuarios que mostrar, notificándose al actor. El caso de uso termina.

Tabla 5.3: Caso de Uso 3.1 - Ver lista de usuarios

CU-3.2	Registrar usuario
Actor	Administrador
Descripción	El sistema deberá permitir al administrador registrar usuarios en el sistema.
Precondición	El actor ha iniciado sesión en el sistema.
Secuencia Normal	
Paso	Acción
1	Se realiza el caso de uso 'Ver lista de usuarios'.
2	El actor selecciona registrar usuario.
3	El sistema solicita los datos necesarios.
4	El actor rellena los datos necesarios para registrar un nuevo usuario y pulsa el botón de registrar.
5	El sistema comprueba que los datos son válidos.
6	El sistema muestra un mensaje de éxito. El caso de uso termina.
Post-condición	El usuario se ha añadido al sistema.
Excepciones	
Variación	Acción
2b, 4b	El actor sale del sistema y el caso de uso queda sin efecto.
5b	El sistema detecta que los datos no son válidos, muestra un mensaje de error y el caso de uso continua en el paso 4.

Tabla 5.4: Caso de Uso 3.2 - Registrar usuario

CU-3.3	Eliminar usuario
Actor	Administrador
Descripción	El sistema deberá permitir al administrador eliminar usuarios del sistema.
Precondición	El actor ha iniciado sesión en el sistema.
Secuencia Normal	
Paso	Acción
1	Se realiza el caso de uso 'Ver lista de usuarios'.
2	El actor pulsa el botón eliminar en el usuario que desee.
3	El sistema muestra un mensaje informando de la acción que se va a realizar y solicitando confirmación.
4	El actor pulsa el botón eliminar.
5	El sistema elimina el usuario del sistema. El caso de uso termina.
Post-condición	El usuario se elimina del sistema.
Excepciones	
Variación	Acción
4b	El actor pulsa el botón cancelar y el caso de uso queda sin efecto.

Tabla 5.5: Caso de Uso 3.3 - Eliminar usuario

5.1.5. Caso de uso 4: Gestión de parámetros del sistema

Los casos de uso de la gestión de parámetros del sistema permiten al jefe editar los parámetros del sistema, visualizar los cargos existentes en el sistema, crear nuevos, editarlos y eliminarlos del sistema.

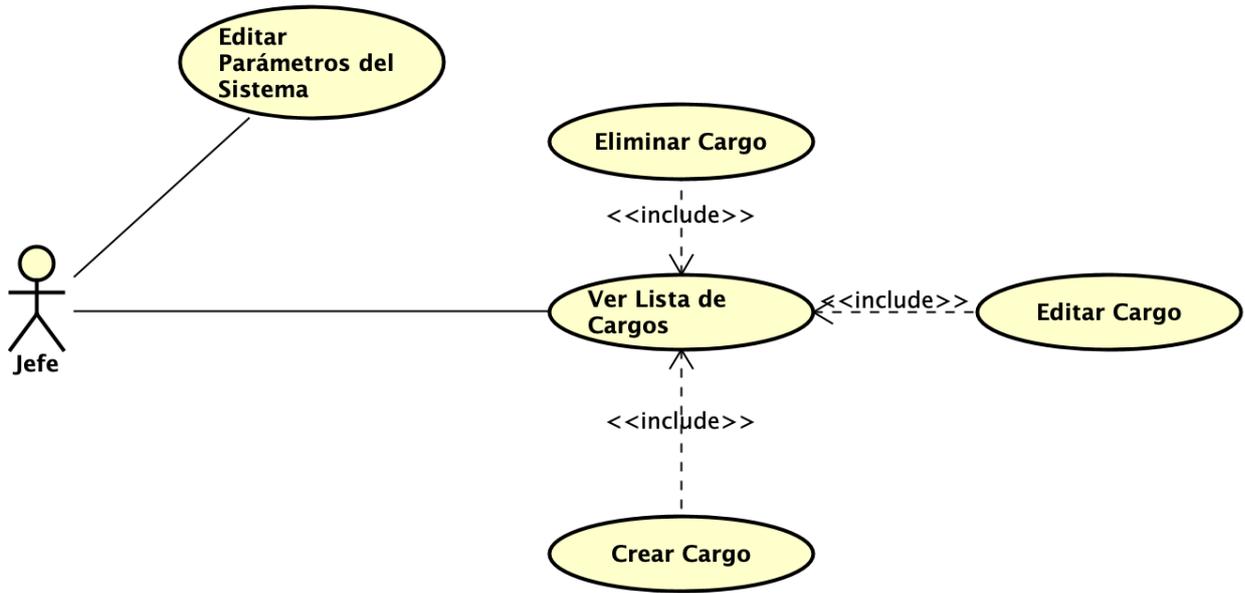


Figura 5.3: Diagrama en detalle de gestión de parámetros

CU-4.1	Editar parámetros del sistema
Actor	Jefe
Descripción	El sistema deberá permitir al jefe editar los parámetros del sistema.
Precondición	El actor ha iniciado sesión en el sistema.
Secuencia Normal	
Paso	Acción
1	El actor entra en la página principal.
2	El actor pulsa el botón de ajustes del sistema.
3	El sistema muestra los parámetros actuales del sistema para modificarlos.
4	El actor modifica los datos y pulsa el botón de guardar.
5	El sistema comprueba que los datos con válidos.
6	El sistema muestra un mensaje de éxito. El caso de uso termina.
Post-condición	Los parámetros del sistema se modifican.
Excepciones	
Variación	Acción
1b, 2b, 4b	El actor sale del sistema y el caso de uso queda sin efecto.
3b, 4c	El actor selecciona la opción cancelar y el caso de uso queda sin efecto.
5b	El sistema detecta que los datos no son válidos, muestra un mensaje de error y el caso de uso continua en el paso 4.

Tabla 5.6: Caso de Uso 4.1 - Editar parámetros del sistema

CU-4.2	Ver lista de cargos
Actor	Jefe
Descripción	El sistema deberá permitir al jefe visualizar un listado de todos los cargos del sistema.
Precondición	El actor ha iniciado sesión en el sistema.
Secuencia Normal	
Paso	Acción
1	El actor entra en la página principal.
2	El actor pulsa el botón de ajustes del sistema.
3	El sistema muestra un listado con los cargos del sistema. El caso de uso termina.
Post-condición	
Excepciones	
Variación	Acción
1b, 2b	El actor sale del sistema. El caso de uso queda sin efecto.
2c	El sistema no contiene cargos que mostrar, notificándose al actor. El caso de uso termina.

Tabla 5.7: Caso de Uso 4.2 - Ver lista de cargos

CU-4.3	Crear cargo
Actor	Jefe
Descripción	El sistema deberá permitir al jefe crear cargos en el sistema.
Precondición	El actor ha iniciado sesión en el sistema.
Secuencia Normal	
Paso	Acción
1	Se realiza el caso de uso 'Ver lista de cargos'.
2	El actor selecciona crear cargo.
3	El sistema solicita los datos necesarios.
4	El actor rellena los datos necesarios para crear un nuevo cargo y pulsa el botón de guardar.
5	El sistema comprueba que los datos son válidos.
6	El sistema muestra un mensaje de éxito. El caso de uso termina.
Post-condición	El cargo se ha añadido al sistema.
Excepciones	
Variación	Acción
2b, 4b	El actor sale del sistema y el caso de uso queda sin efecto.
5b	El sistema detecta que los datos no son válidos, muestra un mensaje de error y el caso de uso continua en el paso 4.

Tabla 5.8: Caso de Uso 4.3 - Crear cargo

CU-4.4	Eliminar cargo
Actor	Jefe
Descripción	El sistema deberá permitir al jefe eliminar cargos del sistema.
Precondición	El actor ha iniciado sesión en el sistema.
Secuencia Normal	
Paso	Acción
1	Se realiza el caso de uso 'Ver lista de cargos'.
2	El actor pulsa el botón eliminar en el cargo que desee.
3	El sistema muestra un mensaje informando de la acción que se va a realizar y solicitando confirmación.
4	El actor pulsa el botón eliminar.
5	El sistema elimina el cargo del sistema. El caso de uso termina.
Post-condición	El cargo se elimina del sistema.
Excepciones	
Variación	Acción
4b	El actor pulsa el botón cancelar y el caso de uso queda sin efecto.

Tabla 5.9: Caso de Uso 4.4 - Eliminar cargo

CU-4.5	Editar cargo
Actor	Jefe
Descripción	El sistema deberá permitir al jefe editar los cargos del sistema.
Precondición	El actor ha iniciado sesión en el sistema.
Secuencia Normal	
Paso	Acción
1	Se realiza el caso de uso 'Ver lista de cargos'.
2	El actor pulsa el botón de editar en el cargo que desee.
3	El sistema muestra los datos actuales del cargo para modificarlos.
4	El actor modifica los datos y pulsa el botón de guardar.
5	El sistema comprueba que los datos son válidos.
6	El sistema muestra un mensaje de éxito. El caso de uso termina.
Post-condición	Los datos del cargo se modifican.
Excepciones	
Variación	Acción
2b, 4b	El actor sale del sistema y el caso de uso queda sin efecto.
4c	El actor selecciona la opción cancelar y el caso de uso queda sin efecto.
5b	El sistema detecta que los datos no son válidos, muestra un mensaje de error y el caso de uso continúa en el paso 4.

Tabla 5.10: Caso de Uso 4.5 - Editar cargo

5.1.6. Caso de uso 5: Gestión de presupuestos

Los casos de uso de la gestión de presupuestos permiten al jefe visualizar los presupuestos aprobados y pendientes de aprobación, aceptar presupuestos, rechazar presupuestos y visualizar archivos de los presupuestos.

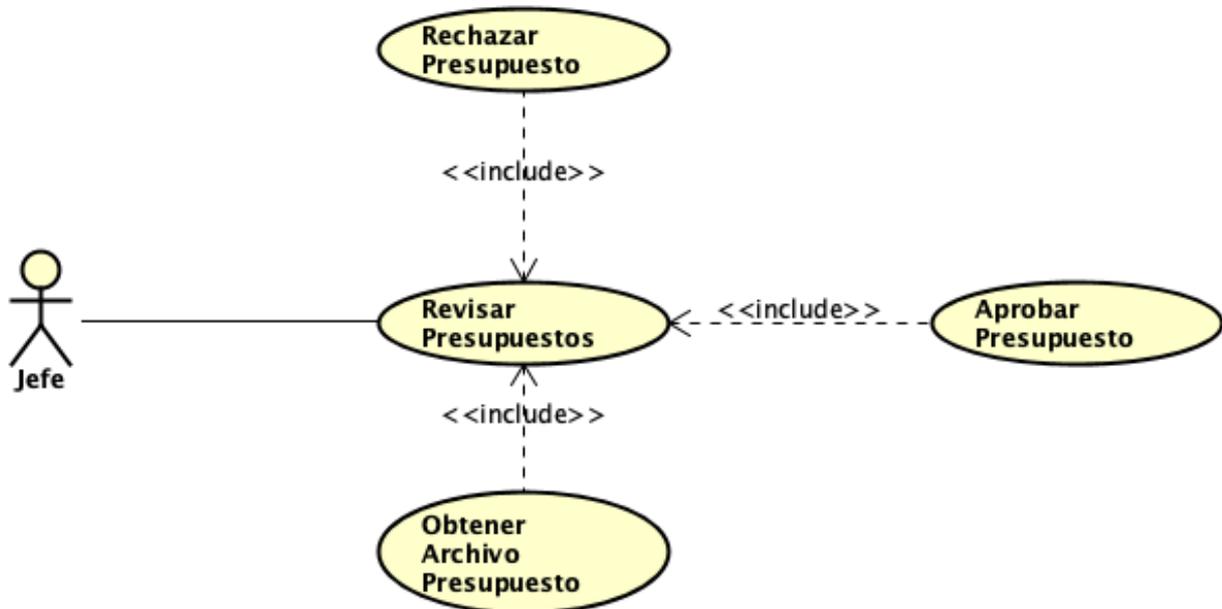


Figura 5.4: Diagrama en detalle de gestión de presupuestos

CU-5.1	Revisar presupuestos
Actor	Jefe
Descripción	El sistema deberá permitir al jefe visualizar un listado de todos los presupuestos aprobados y pendientes de aprobación del sistema.
Precondición	El actor ha iniciado sesión en el sistema.
Secuencia Normal	
Paso	Acción
1	El actor entra en la página principal.
2	El sistema muestra un listado con los presupuestos aprobados y pendientes de aprobación del sistema. El caso de uso termina.
Post-condición	
Excepciones	
Variación	Acción
1b	El actor sale del sistema. El caso de uso queda sin efecto.
2b	El sistema no contiene presupuestos que mostrar, notificándose al actor. El caso de uso termina.

Tabla 5.11: Caso de Uso 5.1 - Revisar presupuestos

CU-5.2	Aprobar presupuesto
Actor	Jefe
Descripción	El sistema deberá permitir al jefe aprobar un presupuesto.
Precondición	El actor ha iniciado sesión en el sistema.
Secuencia Normal	
Paso	Acción
1	Se realiza el caso de uso 'Revisar presupuestos'.
2	El actor pulsa el botón aprobar en el presupuesto pendiente de aprobación que desee.
3	El sistema muestra un mensaje informando de la acción que se va a realizar y solicitando confirmación.
4	El actor pulsa el botón aprobar.
5	El sistema cambia el estado del presupuesto a aprobado. El caso de uso termina.
Post-condición	El estado del presupuesto ha cambiado a aprobado.
Excepciones	
Variación	Acción
2b	El actor sale del sistema. El caso de uso queda sin efecto.
4b	El actor pulsa el botón cancelar y el caso de uso queda sin efecto.

Tabla 5.12: Caso de Uso 5.2 - Aprobar presupuesto

CU-5.3	Rechazar presupuesto
Actor	Jefe
Descripción	El sistema deberá permitir al jefe rechazar un presupuesto.
Precondición	El actor ha iniciado sesión en el sistema.
Secuencia Normal	
Paso	Acción
1	Se realiza el caso de uso 'Revisar presupuestos'.
2	El actor pulsa el botón rechazar en el presupuesto pendiente de aprobación que desee.
3	El sistema muestra un mensaje informando de la acción que se va a realizar y solicitando confirmación.
4	El actor pulsa el botón rechazar.
5	El sistema cambia el estado del presupuesto a editable. El caso de uso termina.
Post-condición	El estado del presupuesto ha cambiado a editable.
Excepciones	
Variación	Acción
2b	El actor sale del sistema. El caso de uso queda sin efecto.
4b	El actor pulsa el botón cancelar y el caso de uso queda sin efecto.

Tabla 5.13: Caso de Uso 5.3 - Rechazar presupuesto

CU-5.4	Obtener archivo presupuesto
Actor	Jefe
Descripción	El sistema deberá permitir al jefe obtener un archivo de un presupuesto.
Precondición	El actor ha iniciado sesión en el sistema.
Secuencia Normal	
Paso	Acción
1	Se realiza el caso de uso 'Revisar presupuestos'.
2	El actor pulsa el botón de obtener archivo en el presupuesto que desee.
3	El sistema muestra el archivo. El caso de uso termina.
Post-condición	
Excepciones	
Variación	Acción
2b	El actor sale del sistema. El caso de uso queda sin efecto.

Tabla 5.14: Caso de Uso 5.4 - Obtener archivo presupuesto

5.1.7. Caso de uso 6: Diseño de presupuestos

Los casos de uso del diseño de presupuestos permiten al comercial visualizar los presupuestos del sistema y crear, editar, eliminar, duplicar y obtener archivos de los mismos.

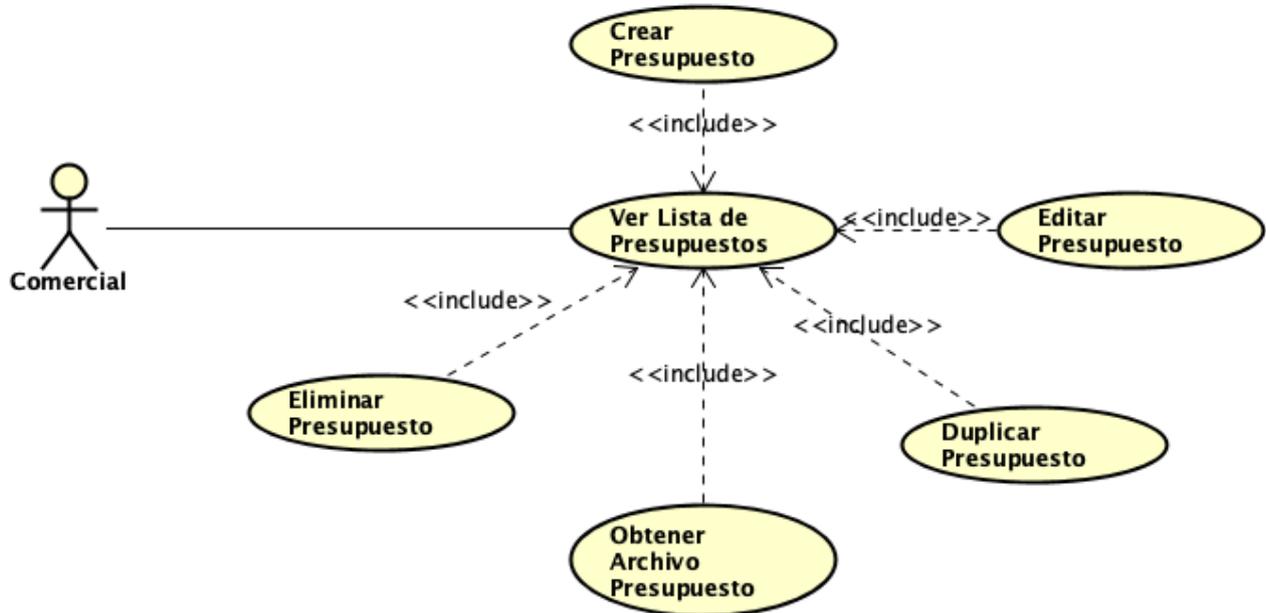


Figura 5.5: Diagrama en detalle de diseño de presupuestos

CU-6.1	Ver lista de presupuestos
Actor	Comercial
Descripción	El sistema deberá permitir al comercial visualizar un listado de todos los presupuestos del sistema.
Precondición	El actor ha iniciado sesión en el sistema.
Secuencia Normal	
Paso	Acción
1	El actor entra en la página principal.
2	El sistema muestra un listado con los presupuestos del sistema. El caso de uso termina.
Post-condición	
Excepciones	
Variación	Acción
1b	El actor sale del sistema. El caso de uso queda sin efecto.
2b	El sistema no contiene presupuestos que mostrar, notificándose al actor. El caso de uso termina.

Tabla 5.15: Caso de Uso 6.1 - Ver lista de presupuestos

CU-6.2	Crear presupuesto
Actor	Comercial
Descripción	El sistema deberá permitir al comercial crear presupuestos en el sistema.
Precondición	El actor ha iniciado sesión en el sistema.
Secuencia Normal	
Paso	Acción
1	Se realiza el caso de uso 'Ver lista de presupuestos'.
2	El actor selecciona crear presupuesto.
3	El sistema solicita los datos necesarios.
4	El actor rellena los datos necesarios para crear un nuevo presupuesto y pulsa el botón de guardar.
5	El sistema comprueba que los datos son válidos.
6	El sistema muestra un mensaje de éxito. El caso de uso termina.
Post-condición	El presupuesto se ha añadido al sistema.
Excepciones	
Variación	Acción
2b, 4b	El actor sale del sistema y el caso de uso queda sin efecto.
4c	El actor rellena los datos y pulsa el botón de guardar y finalizar. El sistema comprueba los datos y si son válidos, se muestra un mensaje de éxito, se cambia el estado del presupuesto a 'pendiente de aprobación' y el caso de uso termina.
5b	El sistema detecta que los datos no son válidos, muestra un mensaje de error y el caso de uso continua en el paso 4.

Tabla 5.16: Caso de Uso 6.2 - Crear presupuesto

CU-6.3	Eliminar presupuesto
Actor	Comercial
Descripción	El sistema deberá permitir al comercial eliminar presupuestos del sistema.
Precondición	El actor ha iniciado sesión en el sistema.
Secuencia Normal	
Paso	Acción
1	Se realiza el caso de uso 'Ver lista de presupuestos'.
2	El actor pulsa el botón eliminar en el presupuesto que desee.
3	El sistema muestra un mensaje informando de la acción que se va a realizar y solicitando confirmación.
4	El actor pulsa el botón eliminar.
5	El sistema elimina el presupuesto del sistema. El caso de uso termina.
Post-condición	El presupuesto se elimina del sistema.
Excepciones	
Variación	Acción
4b	El actor pulsa el botón cancelar y el caso de uso queda sin efecto.

Tabla 5.17: Caso de Uso 6.3 - Eliminar presupuesto

CU-6.4	Editar presupuesto
Actor	Comercial
Descripción	El sistema deberá permitir al comercial editar los presupuestos del sistema.
Precondición	El actor ha iniciado sesión en el sistema.
Secuencia Normal	
Paso	Acción
1	Se realiza el caso de uso 'Ver lista de presupuestos'.
2	El actor pulsa el botón editar en el presupuesto que desee.
3	El sistema muestra los datos actuales del presupuesto para modificarlos.
4	El actor modifica los datos y pulsa el botón de guardar.
5	El sistema comprueba que los datos con válidos.
6	El sistema muestra un mensaje de éxito. El caso de uso termina.
Post-condición	Los datos del cargo se modifican.
Excepciones	
Variación	Acción
2b, 4b	El actor sale del sistema y el caso de uso queda sin efecto.
4c	El actor selecciona la opción cancelar y el caso de uso queda sin efecto.
4d	El actor rellena los datos y pulsa el botón de guardar y finalizar. El sistema comprueba los datos y si son válidos, se muestra un mensaje de éxito, se cambia el estado del presupuesto a 'pendiente de aprobación' y el caso de uso termina.
5b	El sistema detecta que los datos no son válidos, muestra un mensaje de error y el caso de uso continua en el paso 4.

Tabla 5.18: Caso de Uso 6.4 - Editar presupuesto

CU-6.5	Duplicar presupuesto
Actor	Comercial
Descripción	El sistema deberá permitir al comercial duplicar presupuestos del sistema.
Precondición	El actor ha iniciado sesión en el sistema.
Secuencia Normal	
Paso	Acción
1	Se realiza el caso de uso 'Ver lista de presupuestos'.
2	El actor pulsa el botón duplicar en el presupuesto que desee.
3	El sistema muestra un mensaje informando de la acción que se va a realizar y solicitando confirmación.
4	El actor pulsa el botón duplicar.
5	El sistema crea un duplicado del presupuesto en el sistema. El caso de uso termina.
Post-condición	Se ha creado un duplicado del presupuesto en el sistema.
Excepciones	
Variación	Acción
2b	El actor sale del sistema y el caso de uso queda sin efecto.
4b	El actor pulsa el botón cancelar y el caso de uso queda sin efecto.

Tabla 5.19: Caso de Uso 6.5 - Duplicar presupuesto

CU-5.4	Obtener archivo presupuesto
Actor	Comercial
Descripción	El sistema deberá permitir al comercial obtener un archivo de un presupuesto.
Precondición	El actor ha iniciado sesión en el sistema.
Secuencia Normal	
Paso	Acción
1	Se realiza el caso de uso 'Ver lista de presupuestos'.
2	El actor pulsa el botón de obtener archivo en el presupuesto que desee.
3	El sistema muestra el archivo. El caso de uso termina.
Post-condición	
Excepciones	
Variación	Acción
2b	El actor sale del sistema. El caso de uso queda sin efecto.

Tabla 5.20: Caso de Uso 6.6 - Obtener archivo presupuesto

A continuación se muestran diagramas detallados de los casos de uso 6.2, 'Crear presupuesto' y 6.4, 'Editar presupuesto'.

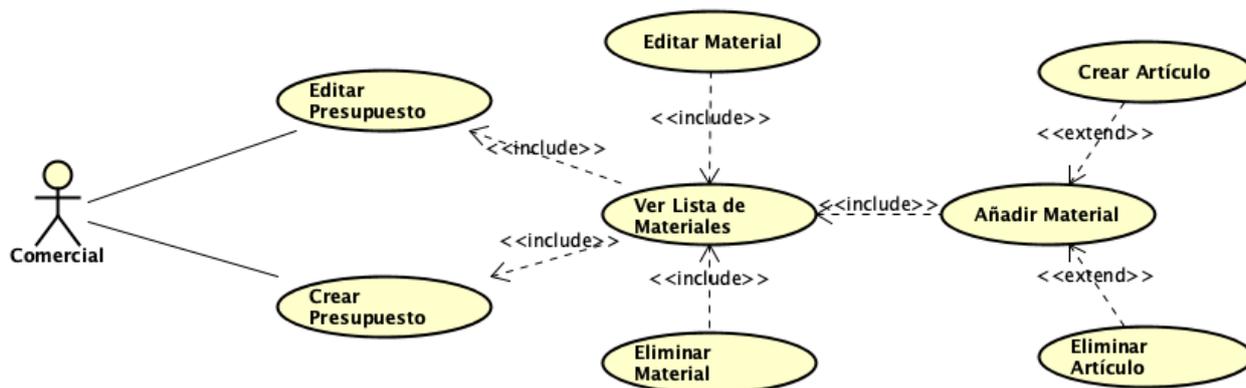


Figura 5.6: Gestión de materiales de un presupuesto

CU-6.7	Ver lista de materiales
Actor	Comercial
Descripción	El sistema deberá permitir al comercial visualizar un listado de todos los materiales de un presupuesto.
Precondición	El actor ha iniciado sesión en el sistema.
Secuencia Normal	
Paso	Acción
1	Se realiza el caso de uso 'Crear presupuesto'.
2	El usuario pulsa el botón de materiales.
3	El sistema muestra un listado con los materiales del presupuesto. El caso de uso termina.
Post-condición	
Excepciones	
Variación	Acción
1b	Se realiza el caso de uso 'Editar presupuesto'. El caso de uso continúa en el paso 2.
3b	El sistema no contiene materiales que mostrar, notificándose al actor. El caso de uso termina.

Tabla 5.21: Caso de Uso 6.7 - Ver lista de materiales

CU-6.8	Añadir material
Actor	Comercial
Descripción	El sistema deberá permitir al comercial añadir materiales a un presupuesto.
Precondición	El actor ha iniciado sesión en el sistema.
Secuencia Normal	
Paso	Acción
1	Se realiza el caso de uso 'Ver lista de materiales'.
2	El actor selecciona añadir material.
3	El sistema muestra el historial de artículos del sistema.
4	El actor pulsa el botón de añadir de uno de los artículos del historial.
5	El sistema solicita los datos necesarios para añadir el material.
6	El actor rellena los datos necesarios para añadir un material y pulsa el botón de guardar.
7	El sistema comprueba que los datos son válidos.
8	El sistema muestra un mensaje de éxito. El caso de uso termina.
Post-condición	El material se ha añadido al presupuesto.
Excepciones	
Variación	Acción
2b, 4b, 6b	El actor sale del sistema y el caso de uso queda sin efecto.
7b	El sistema detecta que los datos no son válidos, muestra un mensaje de error y el caso de uso continua en el paso 4.
4c	El actor pulsa el botón de crear artículo. Se ejecuta el caso de uso 6.11 ¡Crear Artículo!. El caso de uso continua en el paso 5.
4d	El actor pulsa el botón de eliminar de uno de los artículos del historial. Se ejecuta el caso de uso 6.12 ¡Eliminar Artículo!. El caso de uso continua en el paso 3.

Tabla 5.22: Caso de Uso 6.8 - Añadir material

CU-6.12	Eliminar material
Actor	Comercial
Descripción	El sistema deberá permitir al comercial eliminar materiales de un presupuesto.
Precondición	El actor ha iniciado sesión en el sistema.
Secuencia Normal	
Paso	Acción
1	Se realiza el caso de uso 'Ver lista de materiales'.
2	El actor pulsa el botón eliminar en el material que desee.
3	El sistema muestra un mensaje informando de la acción que se va a realizar y solicitando confirmación.
4	El actor pulsa el botón eliminar.
5	El sistema elimina el material del presupuesto. El caso de uso termina.
Post-condición	El material se elimina del presupuesto.
Excepciones	
Variación	Acción
4b	El actor pulsa el botón cancelar y el caso de uso queda sin efecto.

Tabla 5.23: Caso de Uso 6.9 - Eliminar material

CU-6.13	Editar material
Actor	Comercial
Descripción	El sistema deberá permitir al comercial editar los materiales de un presupuesto.
Precondición	El actor ha iniciado sesión en el sistema.
Secuencia Normal	
Paso	Acción
1	Se realiza el caso de uso 'Ver lista de materiales'.
2	El actor pulsa el botón editar en el material que desee.
3	El sistema muestra los datos actuales del material para modificarlos.
4	El actor modifica los datos y pulsa el botón de guardar.
5	El sistema comprueba que los datos con válidos.
6	El sistema muestra un mensaje de éxito. El caso de uso termina.
Post-condición	Los datos del cargo se modifican.
Excepciones	
Variación	Acción
2b, 4b	El actor sale del sistema y el caso de uso queda sin efecto.
4c	El actor selecciona la opción cancelar y el caso de uso queda sin efecto.
5b	El sistema detecta que los datos no son válidos, muestra un mensaje de error y el caso de uso continua en el paso 4.

Tabla 5.24: Caso de Uso 6.10 - Editar material

CU-6.11	Crear artículo
Actor	Comercial
Descripción	El sistema deberá permitir al comercial crear artículos en el historial del sistema.
Precondición	El actor ha iniciado sesión en el sistema.
Secuencia Normal	
Paso	Acción
1	El sistema solicita los datos necesarios para crear un artículo.
2	El actor rellena los datos necesarios y pulsa el botón de guardar.
3	El sistema comprueba que los datos son válidos.
4	El sistema muestra un mensaje de éxito. El caso de uso termina.
Post-condición	El artículo se ha añadido al historial del sistema.
Excepciones	
Variación	Acción
2b	El actor sale del sistema y el caso de uso queda sin efecto.
5b	El sistema detecta que los datos no son válidos, muestra un mensaje de error y el caso de uso continua en el paso 1.

Tabla 5.25: Caso de Uso 6.11 - Crear artículo

CU-6.12	Eliminar artículo
Actor	Comercial
Descripción	El sistema deberá permitir al comercial eliminar artículos del historial del sistema.
Precondición	El actor ha iniciado sesión en el sistema.
Secuencia Normal	
Paso	Acción
1	El sistema muestra un mensaje informando del artículo que se va a eliminar y solicita confirmación.
2	El actor pulsa el botón eliminar.
3	El sistema elimina el artículo del historial. El caso de uso termina.
Post-condición	El artículo se elimina del historial del sistema.
Excepciones	
Variación	Acción
2b	El actor pulsa el botón cancelar y el caso de uso queda sin efecto.

Tabla 5.26: Caso de Uso 6.12 - Eliminar artículo

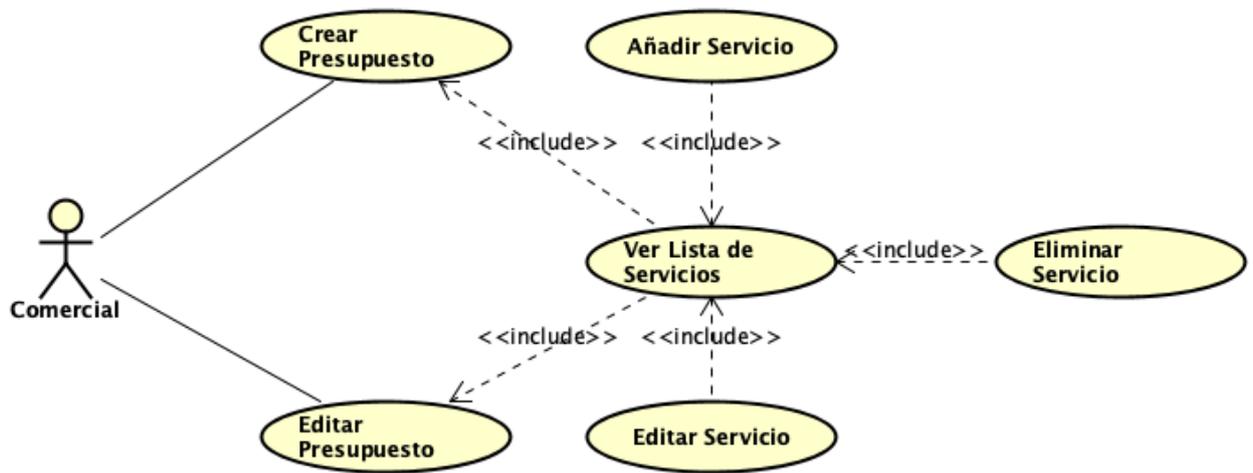


Figura 5.7: Gestión de servicios de un presupuesto

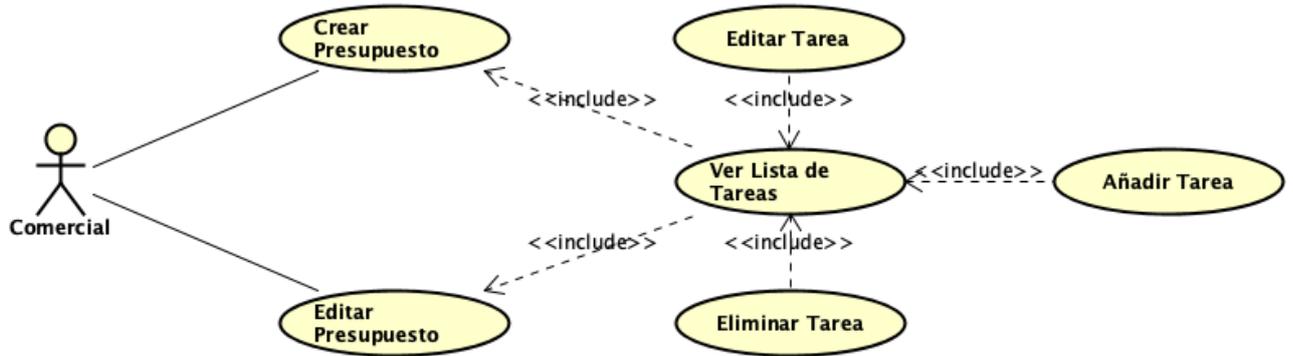


Figura 5.8: Gestión de tareas de un presupuesto

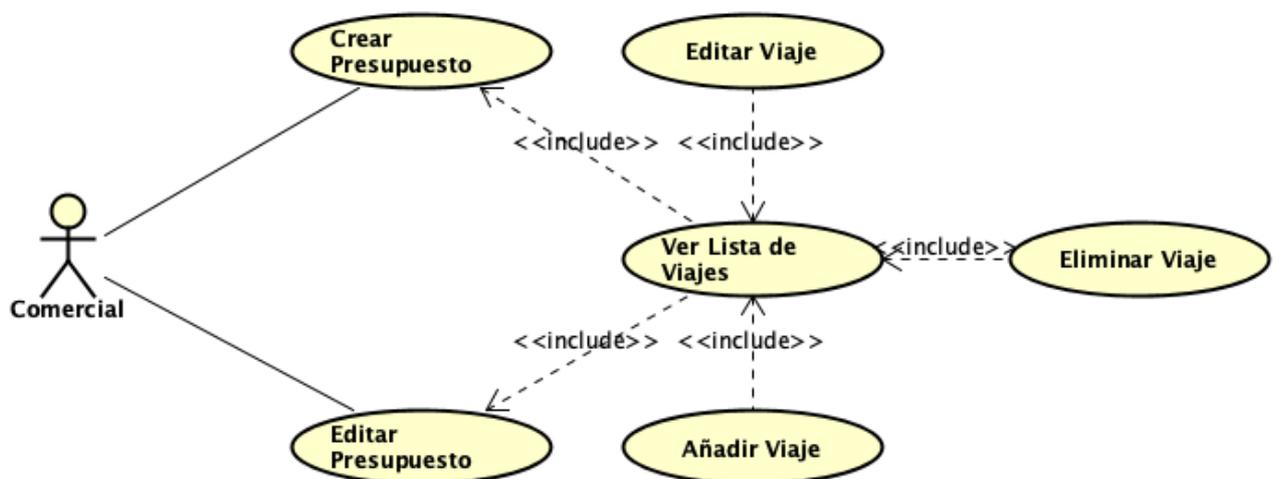


Figura 5.9: Gestión de viajes de un presupuesto

No se incluye descripción detallada de los casos de uso relacionados con la gestión de servicios, tareas y viajes de un presupuesto, debido a que son análogos a los casos de uso previamente descritos para la gestión de materiales de un presupuesto del sistema.

5.2. Modelo de dominio

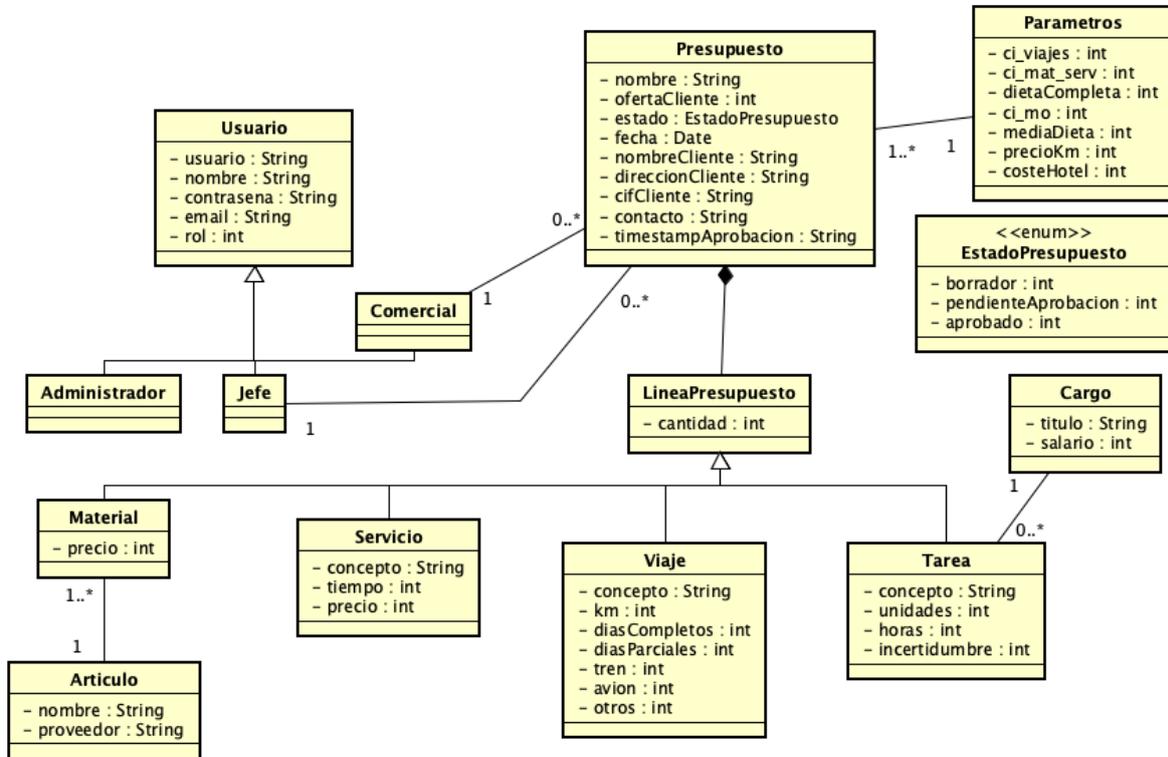


Figura 5.10: Modelo de dominio

Capítulo 6

Diseño

En este capítulo se detallarán los modelos generados durante la fase de diseño del sistema. Estos modelos han sido realizados a partir de los diagramas creados durante la fase de análisis y con el fin de expandir y clarificar los mismos, mostrando así la arquitectura final del sistema desarrollado, así como los servicios que proporciona y las distintas interacciones con el usuario.

6.1. Arquitectura del sistema

En esta sección se detallará la arquitectura elegida para llevar a cabo el desarrollo de cada una de las partes del sistema. Se ha dividido la aplicación en dos partes, utilizando una arquitectura cliente-servidor.

6.1.1. Arquitectura del Back-end

Para realizar la implementación de la parte de servidor, se ha optado por una arquitectura de capas, con 3 estrictas y una relajada. [30] Esto quiere decir que cada capa del sistema se comunica sólo con la que está inmediatamente por debajo, exceptuando la capa relajada, con la que se pueden comunicar el resto de las capas del sistema. Se ha escogido este patrón arquitectónico con el fin de simplificar la implementación todo lo posible.

Las capas en las que se ha dividido el sistema son las siguientes:

- **Routes:** Esta capa se ocupará de recibir y gestionar tanto las peticiones HTTP como las respuestas que se le enviarán al cliente. Estará compuesta por los router, que reciben las peticiones y envían las respuestas, y los `HttpHandlers`, que parsean las peticiones para poder enviarlas a la capa de servicio, y le dan formato a las respuestas recibidas de los servicios para poder enviarlas como respuesta a las peticiones HTTP.
- **Services:** Esta capa se ocupa de toda la lógica de negocio de la aplicación y la comunicación con la base de datos.
- **Model:** Esta capa define los modelos del dominio del sistema necesarios para la comunicación con la base de datos utilizando el ORM.
- **Util:** Esta es la única capa relajada del sistema. Contiene utilidades variadas que pueden ser utilizadas por el resto de las capas.

A continuación se mostrará el diagrama de capas de la arquitectura del back-end del sistema. Debido a su tamaño, se mostrarán dos versiones, una con el detalle de las operaciones y otra sin el detalle, para que se pueda apreciar la organización de las distintas capas.

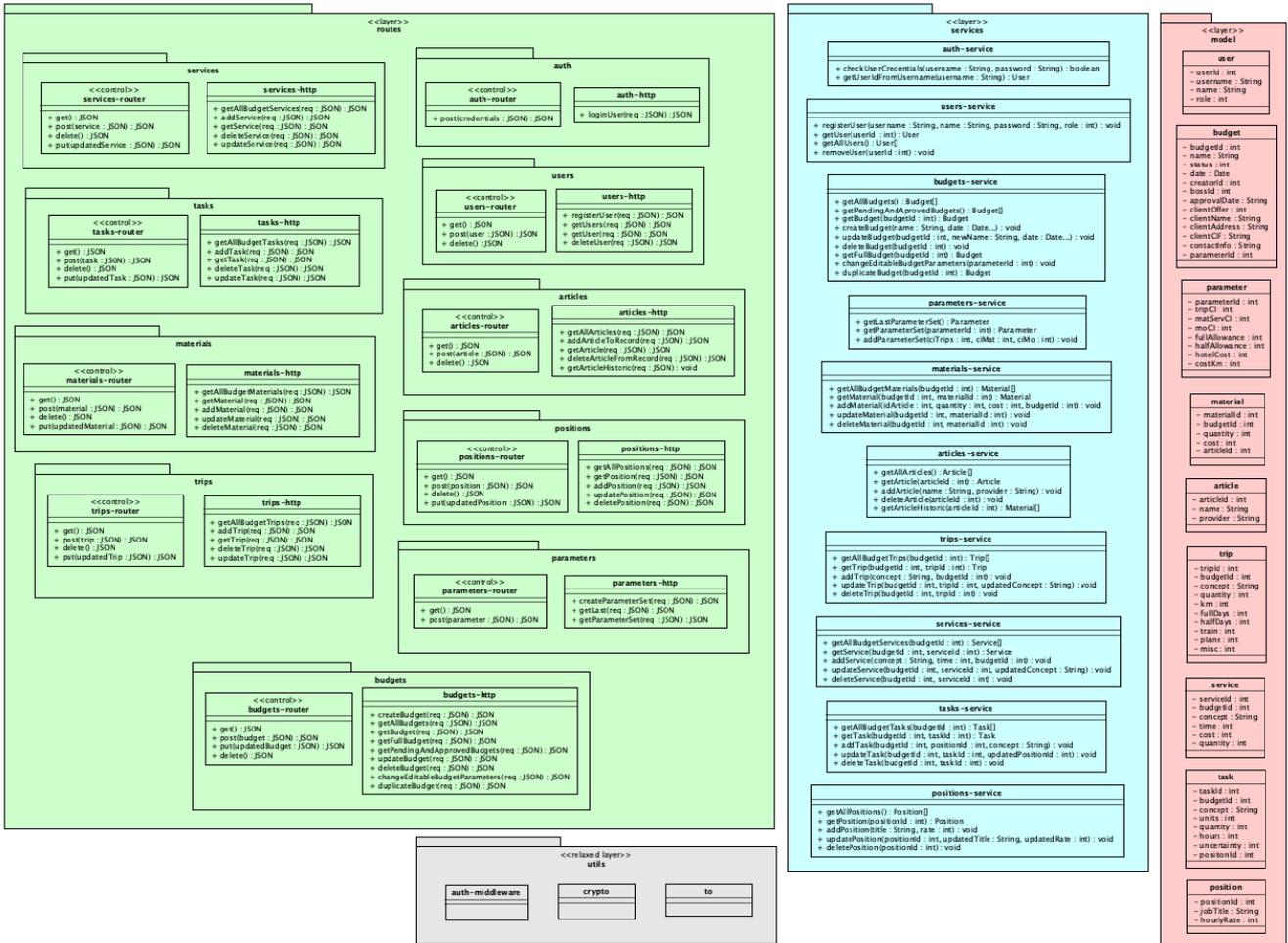


Figura 6.1: Arquitectura usada para el Back-end

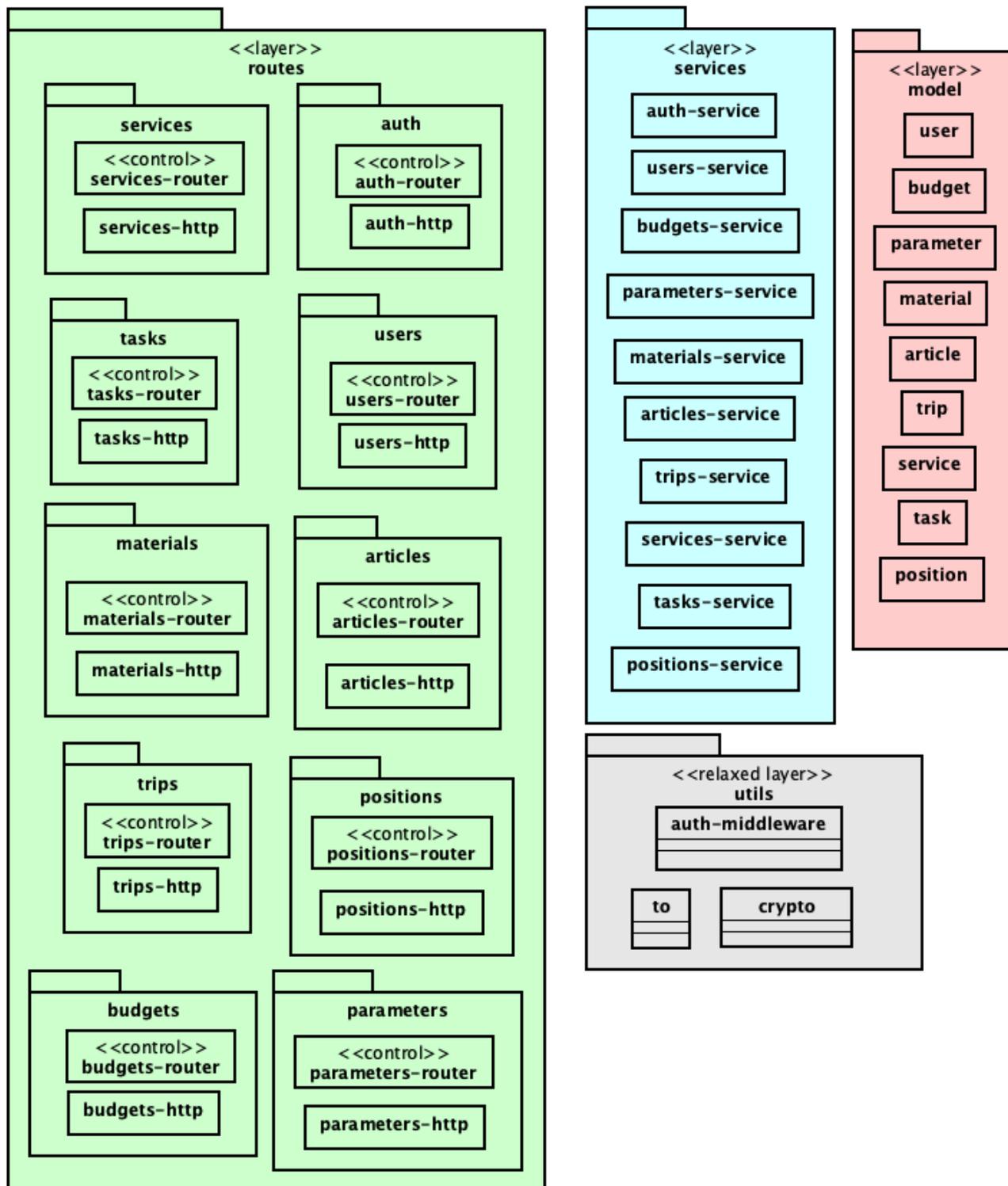


Figura 6.2: Arquitectura usada para el Back-end (Simplificada)

6.1.2. Arquitectura Front-end

La arquitectura del front-end del sistema ha sido diseñada utilizando una simplificación del patrón MVVM [29], ya que React como librería no implementa ningún patrón arquitectónico por sí misma. El patrón MVVM (Model-View-ViewModel, o Modelo-Vista-VistaModelo) ayuda a separar la lógica de negocio de la interfaz de usuario. Se compone de tres partes principales:

- **Modelo:** Contiene el modelo de datos y la lógica de negocio de nuestra aplicación.
- **VistaModelo:** Contiene toda la lógica de presentación, e implementará propiedades y funciones que nos permitan definir las funcionalidades que va a tener nuestra aplicación, y que se enlazarán (también conocido como binding) con la Vista. Es decir, se encarga de realizar la comunicación entre el Modelo y la Vista de nuestro sistema.
- **Vista:** Es la interfaz gráfica de nuestra aplicación, con la que interactuará el usuario.

Para explicar como se adaptará esta arquitectura al diseño de nuestro sistema, es necesario explicar algunos de los conceptos principales de React:

- **Componentes:** Los componentes [26] nos permiten separar la interfaz de usuario en piezas independientes. Dichos componentes aceptarán entradas arbitrarias (llamadas "props." propiedades) y devolverán a React elementos que describen aquello que debe aparecer en pantalla.
- **Estado:** El estado de un componente [28] son aquellas propiedades que van asociado a él. Cuando estas propiedades cambian, el componente se re-renderiza para mostrar la información apropiada dependiendo de los valores de dicho estado. El estado en React se considera inmutable y sólo se puede cambiar utilizando la función especializada `setState()`. Un componente puede tener tantos estados como sea necesario.
- **Context API:** El Contexto en React [27] proporciona una forma de pasar datos a través del árbol de componentes sin tener que pasar props manualmente entre niveles. En React el flujo de datos entre componentes va desde arriba hacia abajo en el árbol de componentes. Es decir, los componentes padres pueden pasar propiedades a sus hijos, y así sucesivamente, pero nunca al revés. El Context API proporciona una forma de crear un contexto global formado por propiedades a las que podrán acceder por igual todos aquellos componentes que se encuentren dentro del Provider de dicho contexto, sin necesidad de que esas propiedades hayan pasado previamente por los componentes que están por encima de ellos.

Una vez explicados estos conceptos básicos, los componentes principales de la arquitectura del sistema son los siguientes [16]:

- **Modelo:** Se considerará modelo todos los API services del sistema, que serán los que se comuniquen con la API REST y obtengan la información necesaria para poder pasársela a los componentes.
- **VistaModelo:** Incluye todos los contextos del sistema, así como el estado local y la lógica almacenada en cada uno de los componentes.
- **Vista:** Parte visual de los componentes (JSX, CSS).

La comunicación entre componentes y API services se llevará a cabo de la siguiente forma:

- El usuario interactúa con la interfaz. Dicha interacción causa un cambio en el estado del componente.

- Cuando React detecta que el estado ha cambiado, provoca un re-renderizado del componente correspondiente.
- Si es necesario, el componente interactuará con el service correspondiente para recibir o enviar la información correspondiente al API REST.

A continuación se mostrará un diagrama que pretende ilustrar la interacción entre el usuario y los componentes con sus respectivos estados, inspirado en uno similar pero con la gestión de estados utilizando el Context API [21]:

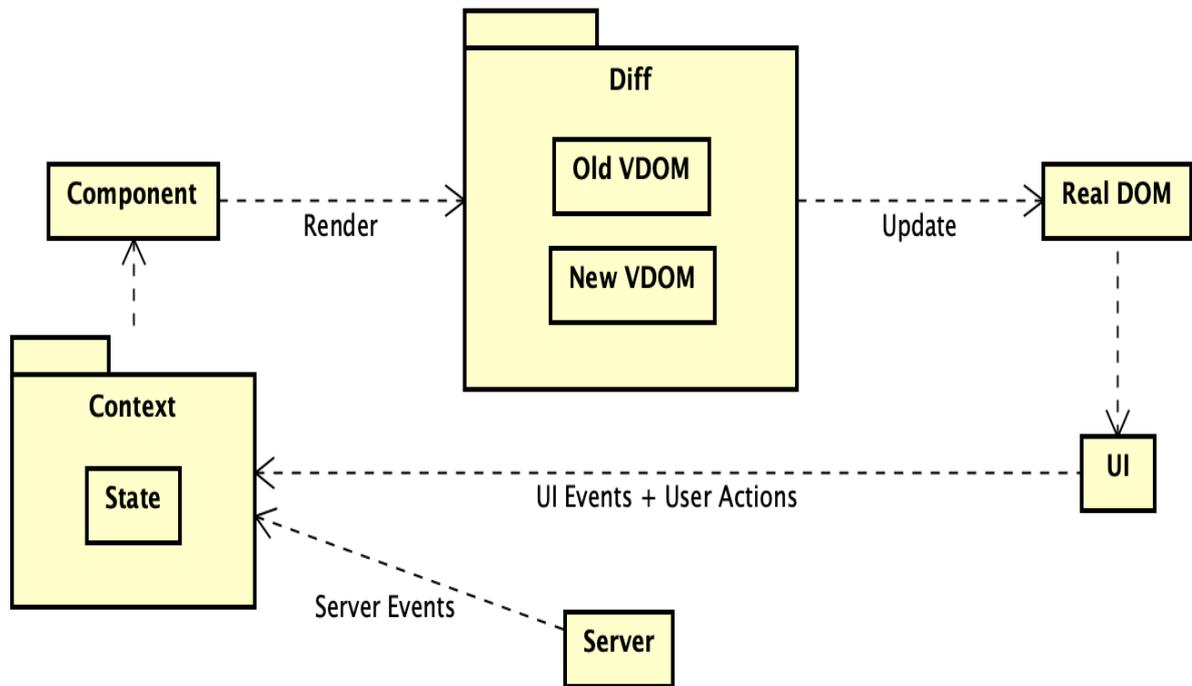


Figura 6.3: Funcionamiento React + Context API

Finalmente, se mostrará un diagrama de paquetes que ilustra como están organizados los distintos componentes y paquetes del sistema:

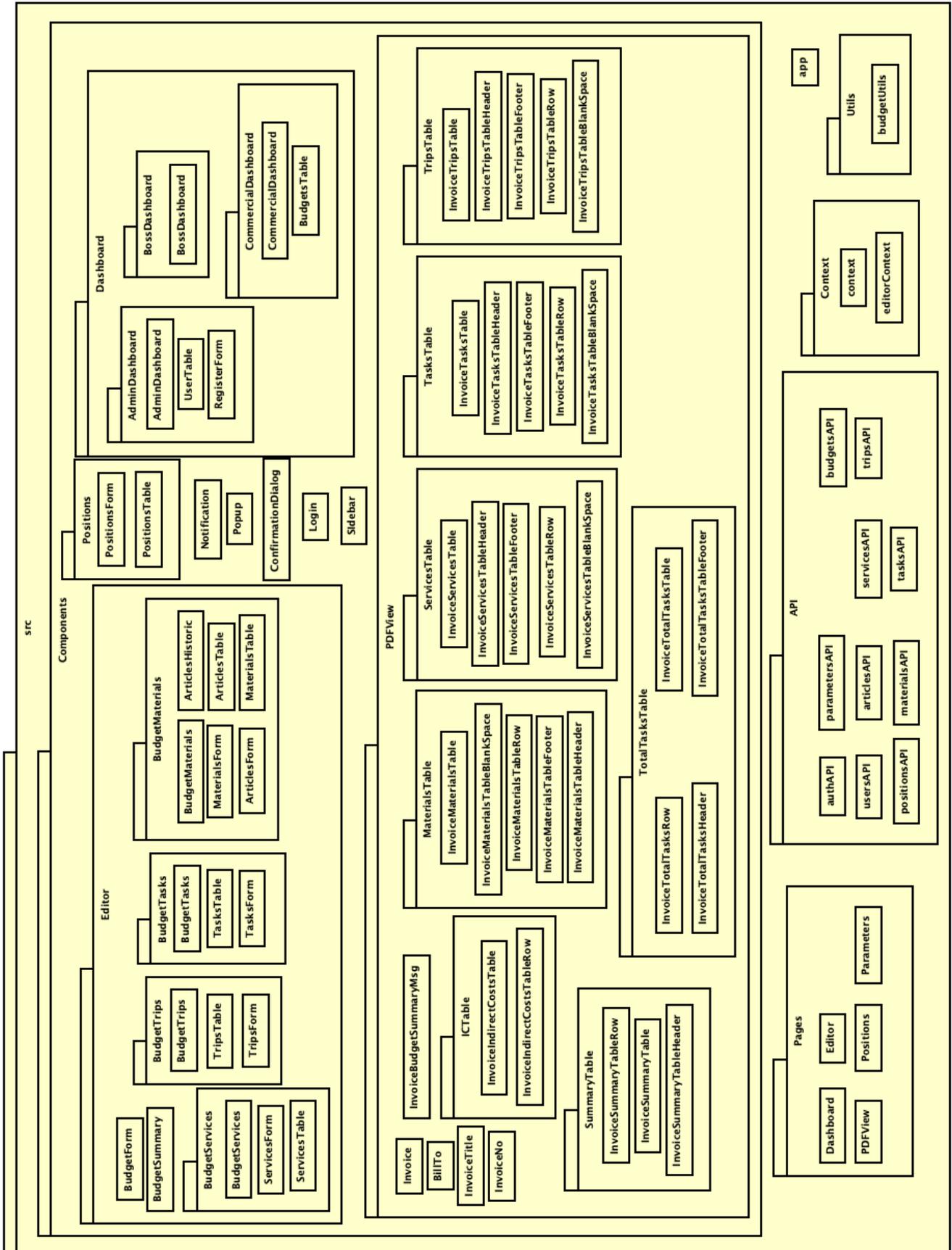


Figura 6.4: Arquitectura usada para el Front-end

6.2. Interfaz de usuario

Los bocetos de esta sección se han realizado utilizando la herramienta web Moqup [7]. Se han incluido las vistas que se consideraban más relevantes a la hora de mostrar el funcionamiento general de la aplicación, obviando aquellas vistas que resulten análogas a alguna de las ya mostradas.

6.2.1. Vista sobre inicio de sesión

Se incluye la vista para iniciar sesión en la aplicación.



Figura 6.5: Inicio de Sesión

6.2.2. Vistas sobre diseño y gestión de presupuestos

Se incluyen las vistas involucradas con la visualización, creación, modificación y eliminación de los presupuestos del sistema por parte de los comerciales.

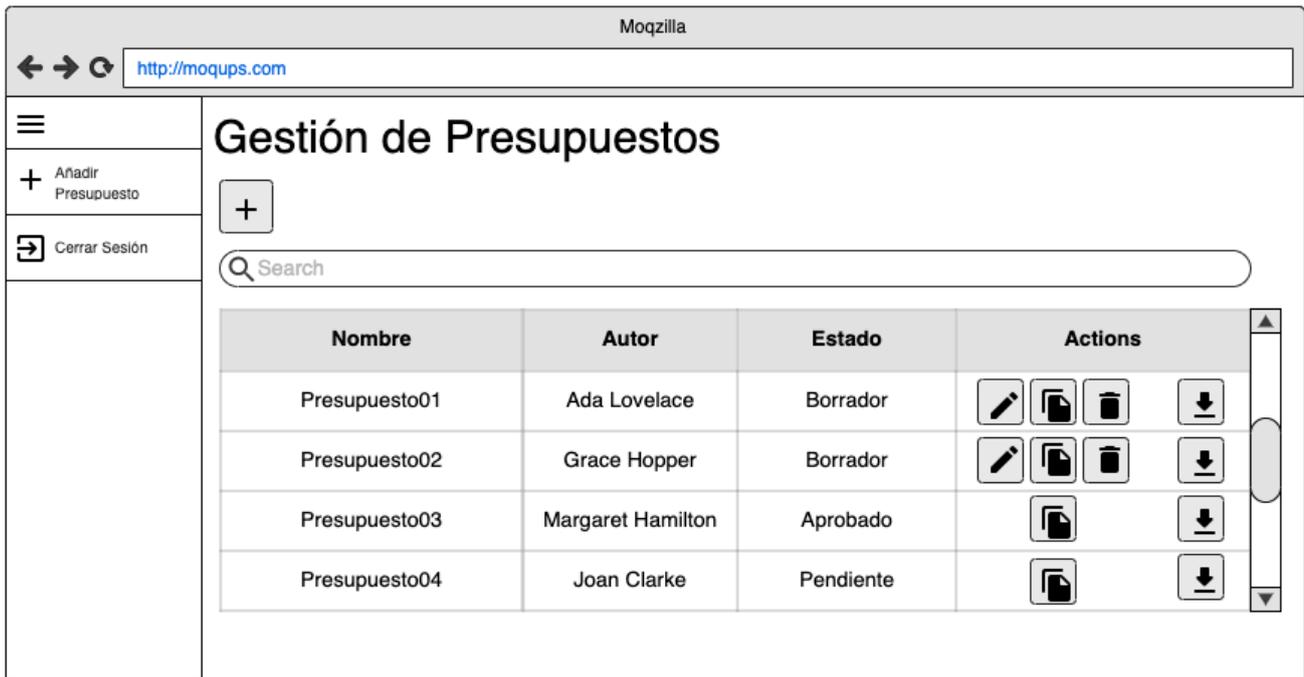


Figura 6.6: Panel de Gestión de Presupuestos

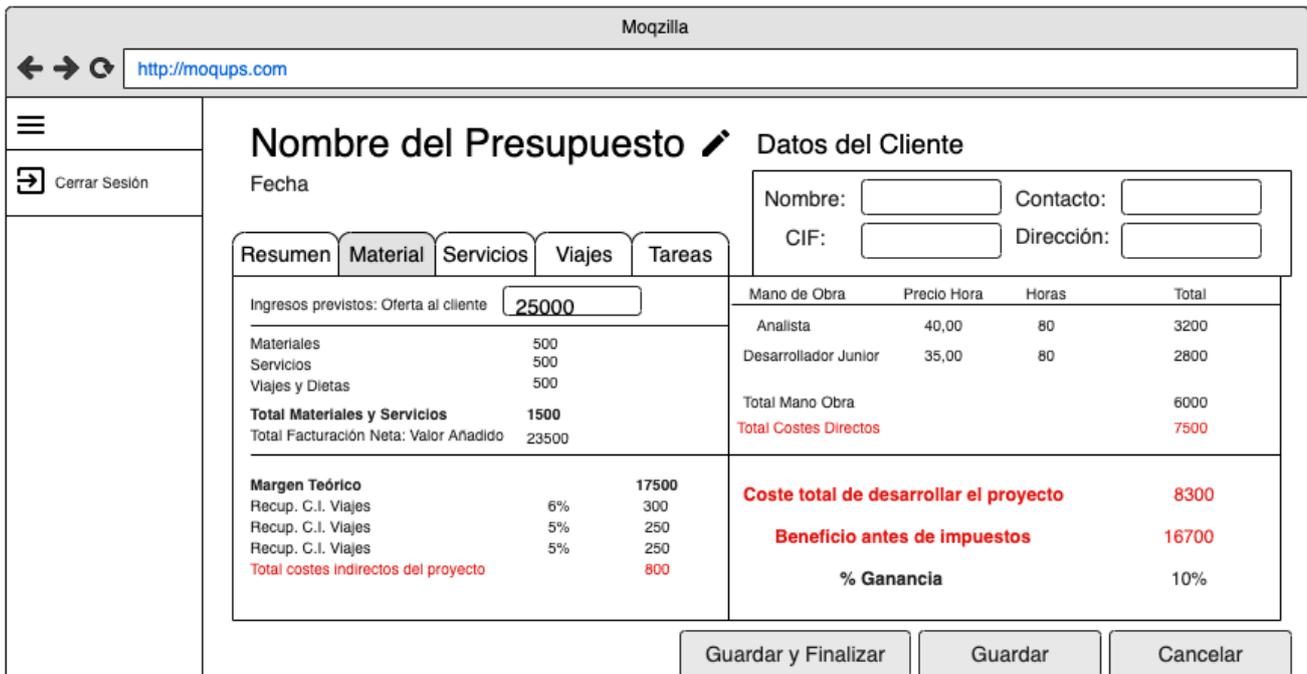


Figura 6.7: Creación (o edición) de un presupuesto - Vista General

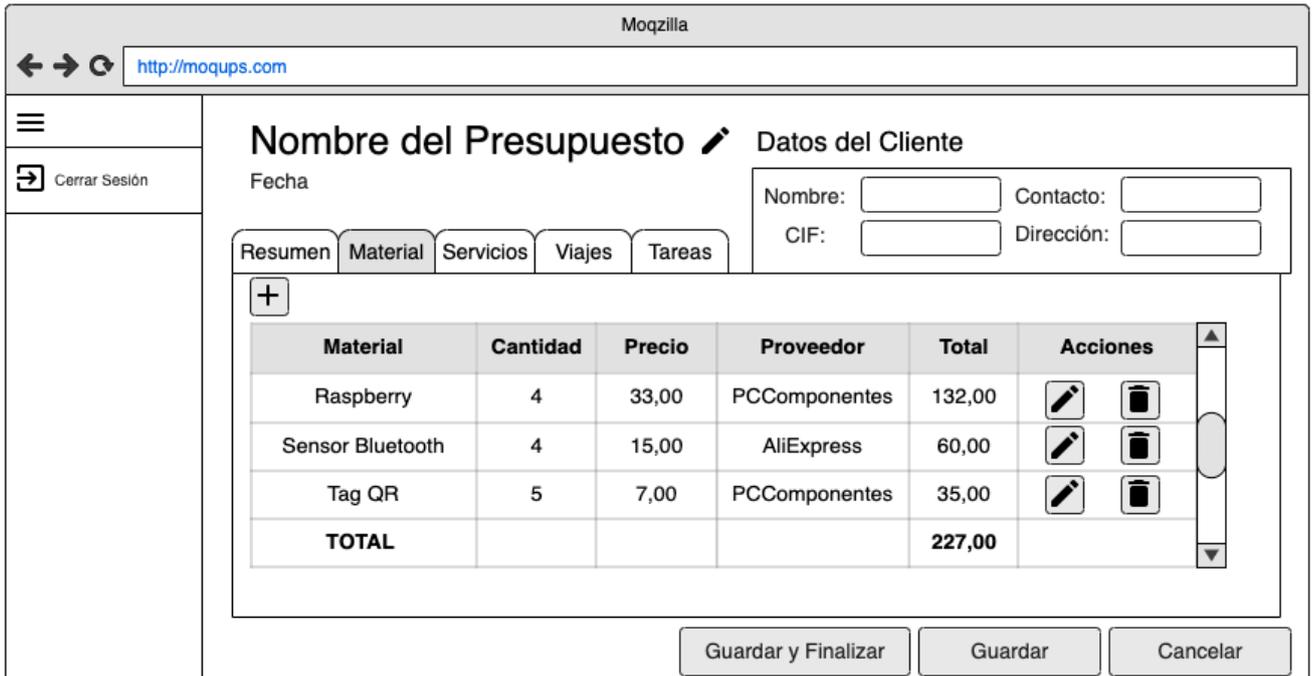


Figura 6.8: Creación (o edición) de un presupuesto - Gestión de Materiales

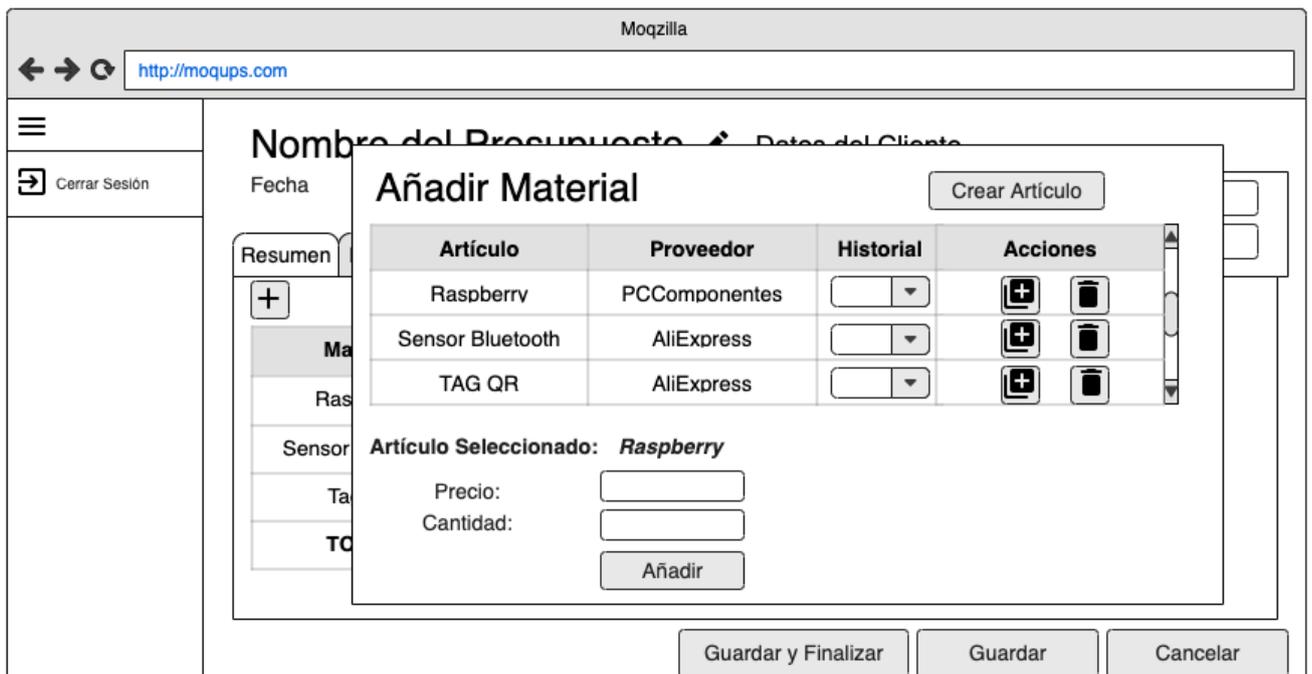


Figura 6.9: Añadir material a un presupuesto

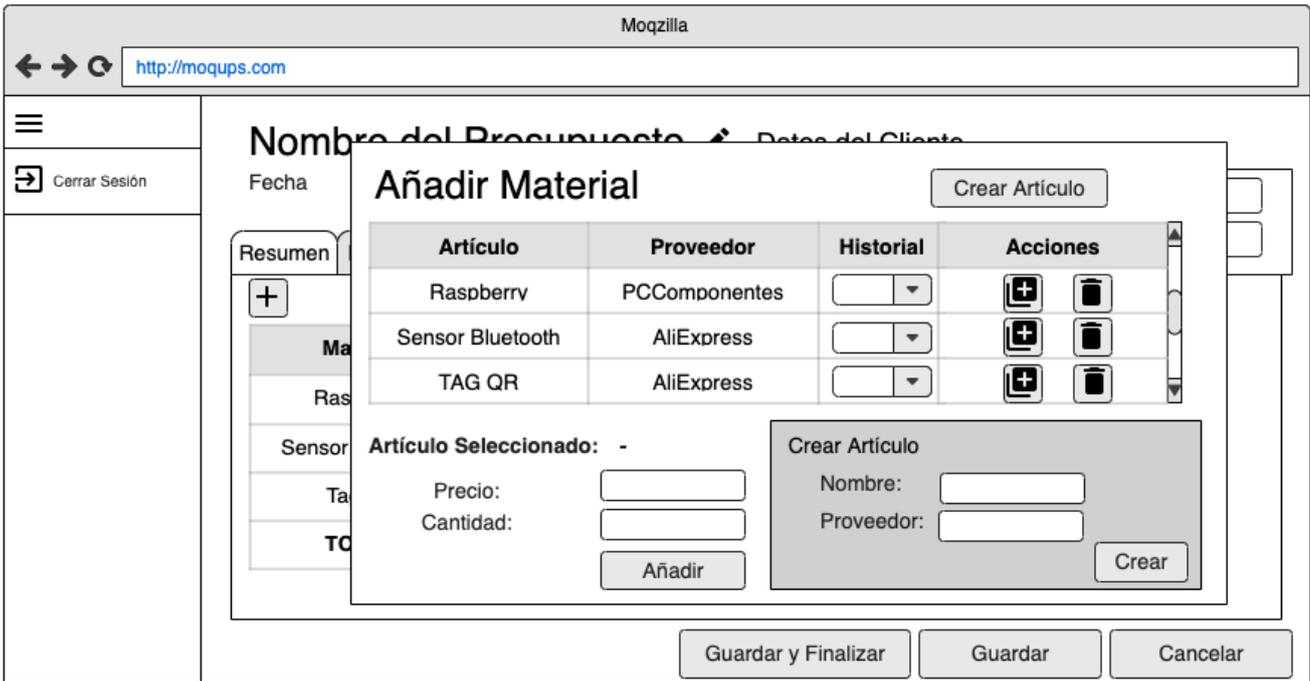


Figura 6.10: Añadir artículo al histórico del sistema

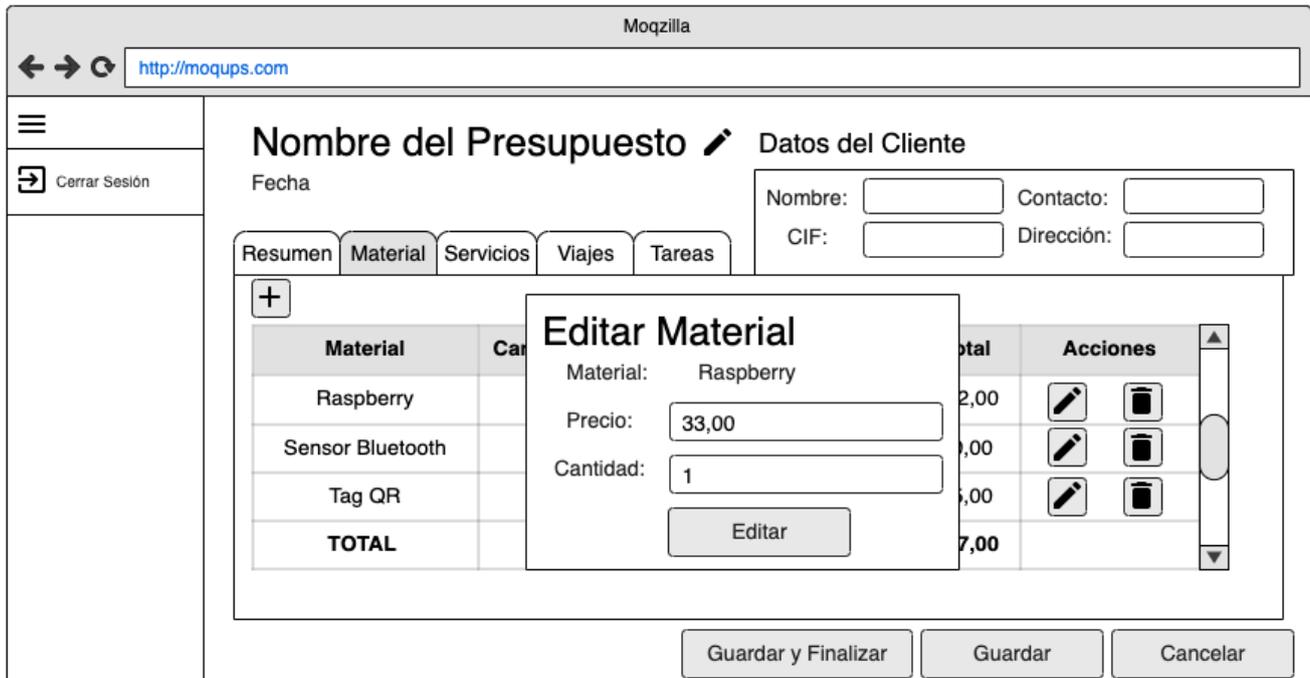


Figura 6.11: Editar material de un presupuesto



Figura 6.12: Creación (o edición) de un presupuesto - Gestión de Servicios

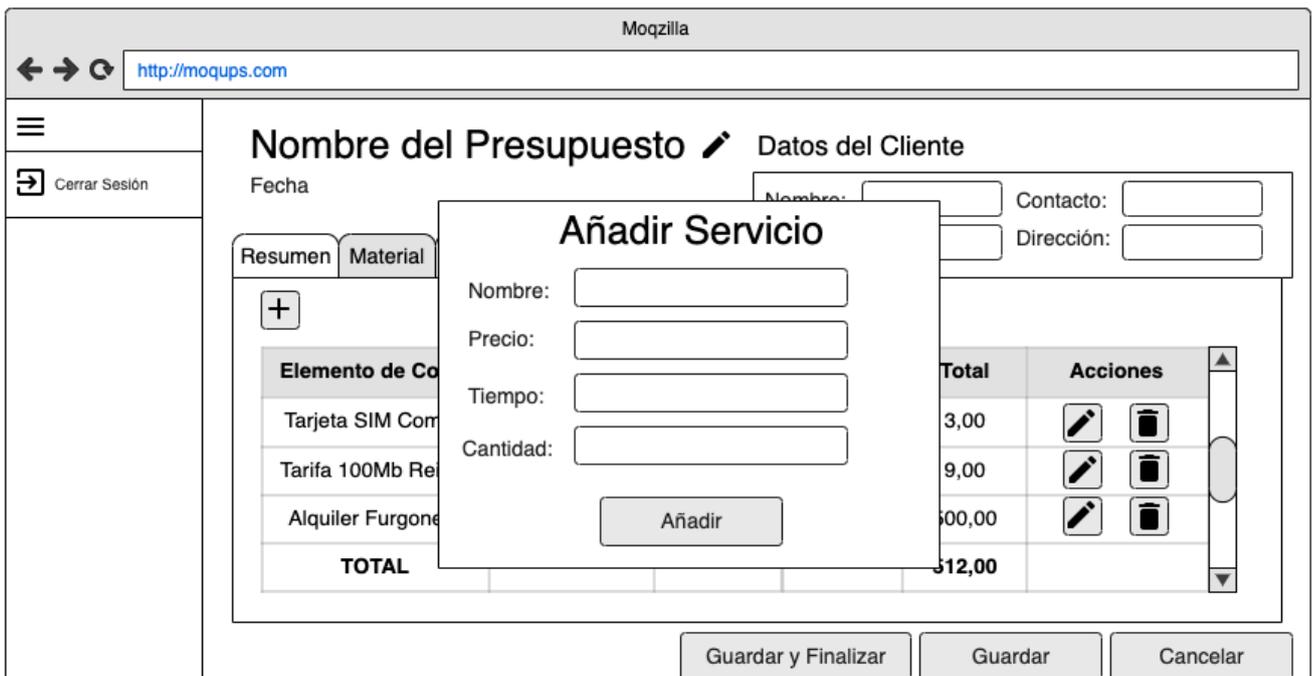


Figura 6.13: Añadir un servicio a un presupuesto

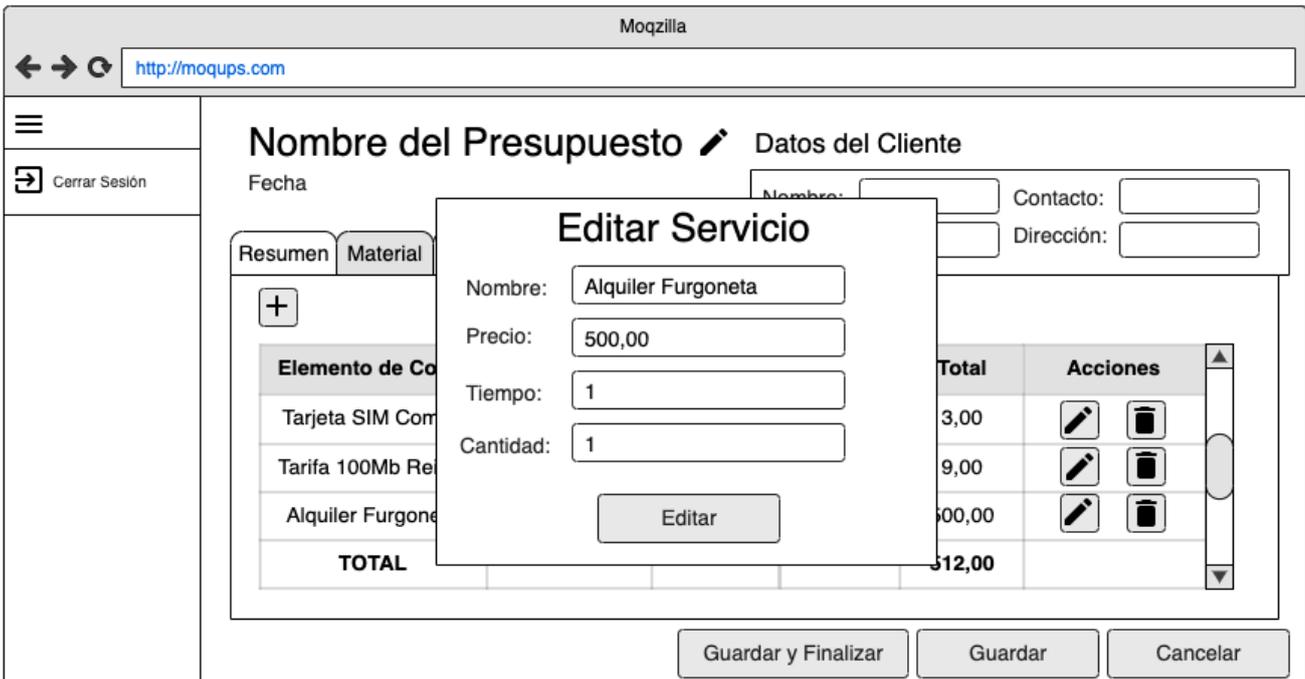


Figura 6.14: Editar servicio de un presupuesto

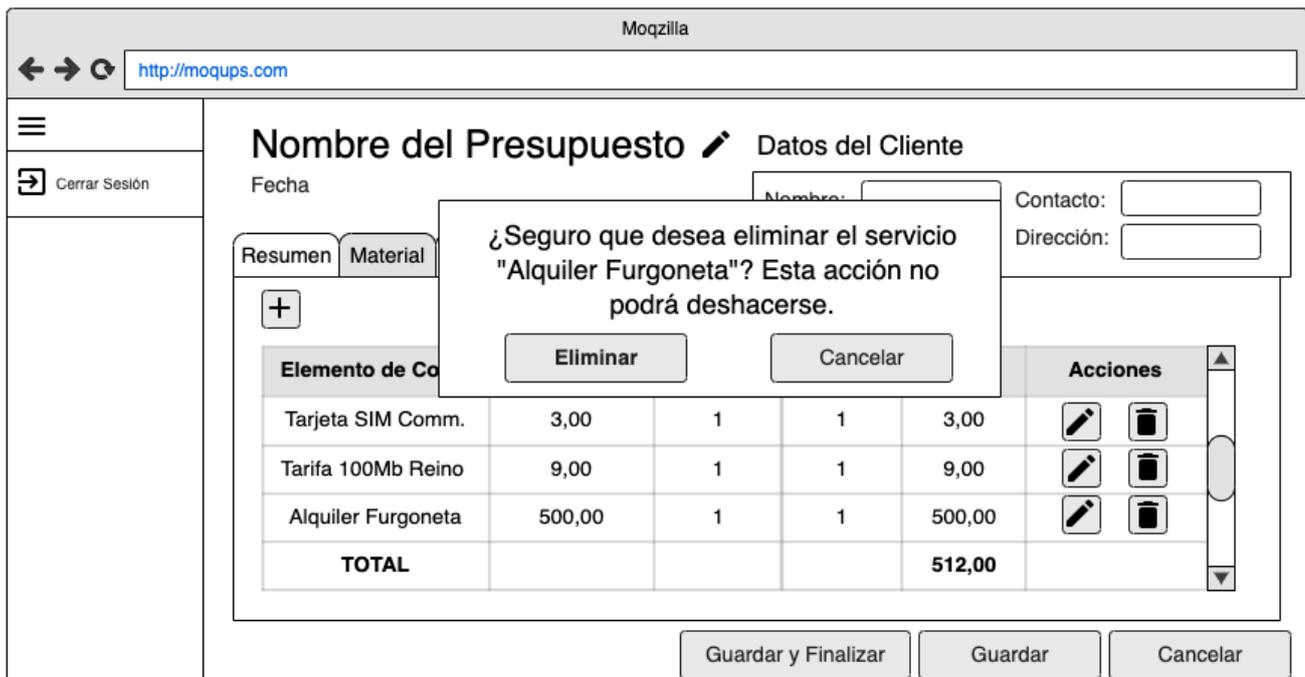


Figura 6.15: Eliminar servicio de un presupuesto

Mozilla

← → ↻ <http://moqups.com>

☰

🔑 Cerrar Sesión

Nombre del Presupuesto

Fecha:

Nombre:

CIF:

Contacto:

Dirección:

Resumen
Material
Servicios
Viajes
Tareas

Dieta Completa Nacional: 30,00 Coste Km: 0,22

1/2 Dieta Nacional: 15,00 Coste Hotel: 70,00

+

Concepto	Km	Total	Trenes	Aviones	Otros	Dias	1/2	Total	Acciones
Viaje 1	100	20,00	-	-	-	5	0	520	
Viaje 2	200	44,00	-	-	-	2	0	244	
Viaje 3	50	10,00	-	-	-	5	1	525	
TOTAL								1279	

Guardar y Finalizar
Guardar
Cancelar

Figura 6.16: Creación (o edición) de un presupuesto - Gestión de viajes

Mozilla

← → ↻ <http://moqups.com>

☰

🔑 Cerrar Sesión

Nombre del Presupuesto

Fecha:

Nombre:

CIF:

Contacto:

Dirección:

Resumen
Material
Servicios
Viajes
Tareas

+

Tarea	Unidades	h/unidad	Incert.	Perfil	Total h	Actions
Desarrollo API	8	40	1	Des. Junior	320	
Analisis	7	40	1	Analista	280	
Arquitectura	4	30	2	Arq. Software	240	
Pruebas	1	5	1	Des. Junior	5	

Guardar y Finalizar
Guardar
Cancelar

Figura 6.17: Creación (o edición) de un presupuesto - Gestión de Tareas

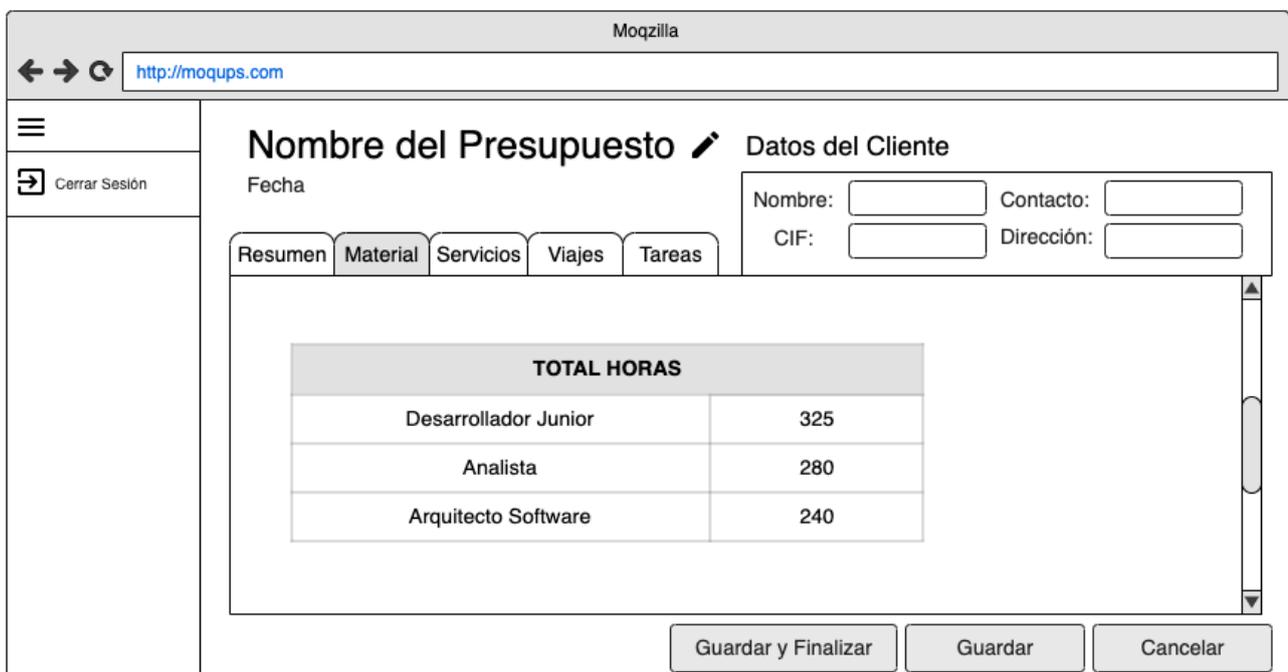


Figura 6.18: Creación (o edición) de un presupuesto - Gestión de Tareas

6.2.3. Revisión de presupuestos y ajustes de parámetros del sistema

Se incluyen las vistas para visualizar, aceptar y rechazar presupuestos por parte de los jefes, así como la vista que permitirá la gestión de parámetros del sistema y la visualización, creación, edición y eliminación de los distintos cargos o perfiles profesionales del mismo.

The screenshot shows the 'Ajustes del Sistema' page. The browser address bar is 'http://moqups.com'. The left sidebar contains 'Revisión Presupuestos' and 'Cerrar Sesión'. The main content area is divided into two sections: 'Costes Desplazamientos (€)' and 'Recup. Costes Indirectos'. The first section has input fields for 'Dieta Completa Nacional' (30), 'Media Dieta Nacional' (15), 'Coste Noche Hotel' (70), and 'Precio/Kilómetro' (0,22). The second section has input fields for '% C.I. Viajes' (6), '% C.I. Materiales y Servicios' (5), and '% C.I. M/O' (6). Below these is a 'Cargos' section with a '+' button and a table of job titles.

Título	€/h	Acciones
Analista	40	
Desarrollador Junior	35	

Figura 6.19: Gestión de parámetros y cargos del sistema

The screenshot shows the 'Revisión de Presupuestos' page. The browser address bar is 'http://moqups.com'. The left sidebar contains 'Ajustes' and 'Cerrar Sesión'. The main content area features a search bar and a table of budget items.

Nombre	Autor	Estado	Acciones
Presupuesto01	Comercial01	Pendiente	
Presupuesto02	Comercial02	Pendiente	
Presupuesto03	Comercial03	Aprobado	
Presupuesto04	Comercial04	Aprobado	

Figura 6.20: Panel de revisión de presupuestos

6.2.4. Gestión de usuarios

Se incluyen las vistas para la visualización, creación y eliminación de usuarios del sistema por parte del administrador.

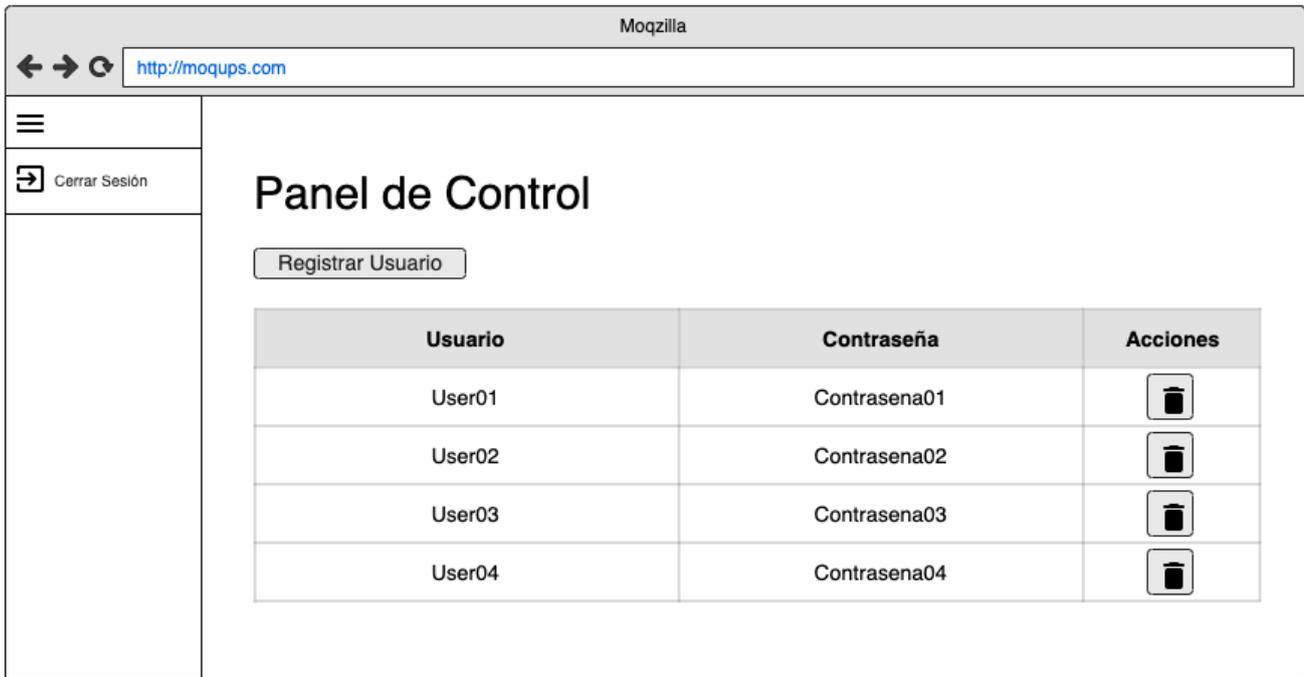


Figura 6.21: Panel de control para la gestión de usuarios



Figura 6.22: Registro de un Usuario

6.3. Diagrama de paths de la aplicación

En esta sección se mostrará el diagrama de paths del sistema. Este diagrama permite ver en detalle las rutas y recursos de los que dispondrá el sistema, así como los distintos endpoints a los que se podrán realizar peticiones HTTP. A su vez, esto nos permite tener una vista de alto nivel del tipo de operaciones que el usuario podrá llevar a cabo al utilizar el sistema.

Debido al gran tamaño del diagrama, se proporciona una versión con los endpoints y otra sin ellos que permita poder visualizar mejor la organización de las rutas del sistema.

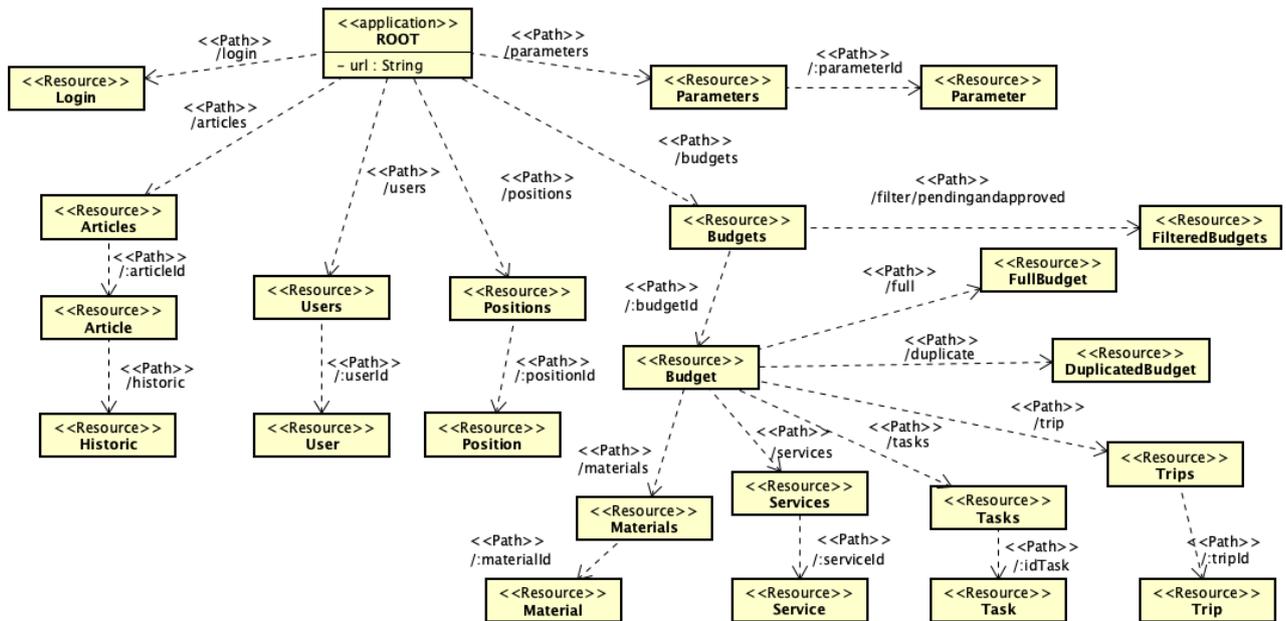


Figura 6.23: Diagrama de Paths simplificado de la aplicación

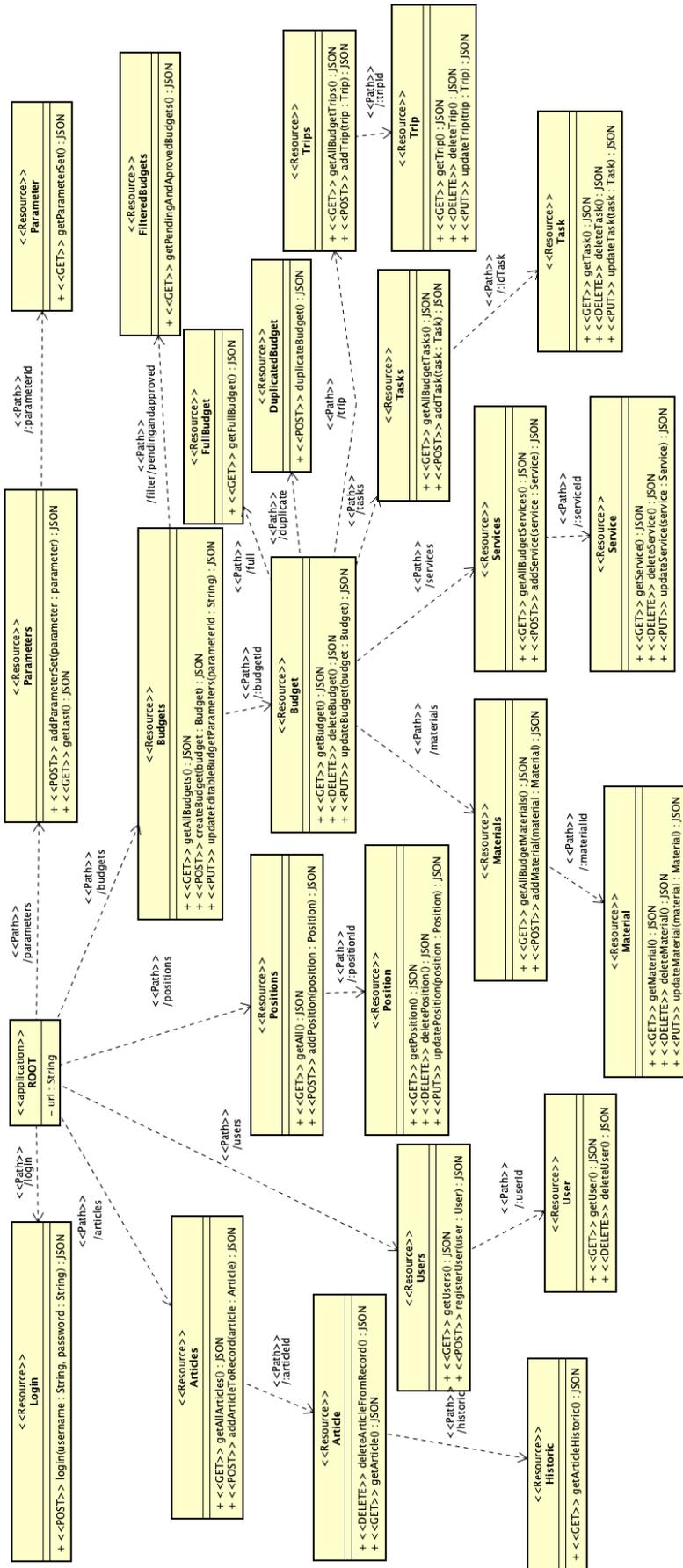


Figura 6.24: Diagrama de Paths de la aplicación

6.4. Diagramas de secuencia en diseño

En esta sección se mostrarán los diagramas de secuencia en diseño para los casos de uso que se han considerado más relevantes para la comprensión del funcionamiento global del sistema. Debido a que el sistema a desarrollar está formado por una parte de servidor o back-end y una parte de cliente o front-end, esta sección se dividirá en dos subsecciones para explicar con más detalle cada una de las partes del sistema implementado.

Cabe destacar que que todos los casos de uso son muy similares entre sí, por lo que sólo se van a detallar algunos de ellos, haciendo hincapié en aquellos casos de uso que tengan alguna cosa distinta al resto.

6.4.1. Diagramas de secuencia en diseño del Back-end

En este apartado se mostrarán los diagramas de secuencia en diseño para los casos de uso más relevantes de la parte de servidor o back-end de la aplicación.

Para la correcta comprensión de esta parte del sistema se va a detallar un caso de uso de cada tipo. Los casos de uso estándares y que se ven repetidos a lo largo de todo el sistema son los de visualización, creación, edición y eliminación de datos. Por tanto, se mostrará uno de cada tipo y se hará especial hincapié en el caso de uso de inicio de sesión por parte de los usuarios, al ser este un caso de uso completamente distinto al resto.

En el caso de las interacciones del sistema con la base de datos, no se detallarán demasiado ya que se ha utilizado un ORM para facilitar esta tarea, lo cual simplifica todo bastante

Se utilizará color verde para representar la capa de las rutas, azul para la capa de los servicios, rojo para los modelos y finalmente gris para aquellas entidades que pertenezcan a la capa relajada de utilidades del sistema o *utils*.

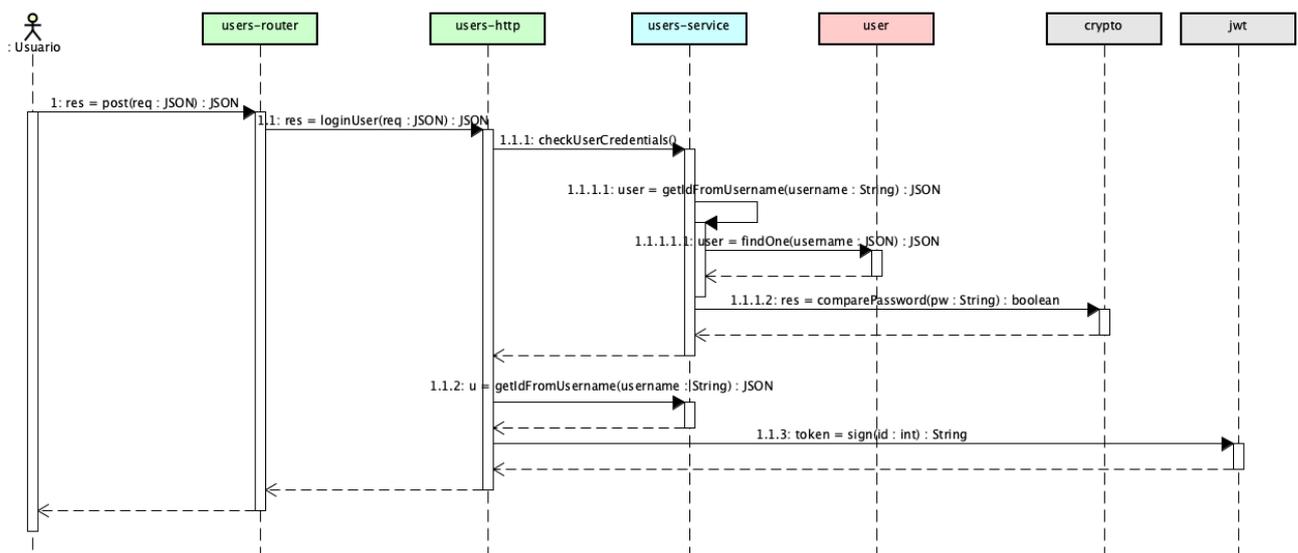


Figura 6.25: Inicio de Sesión

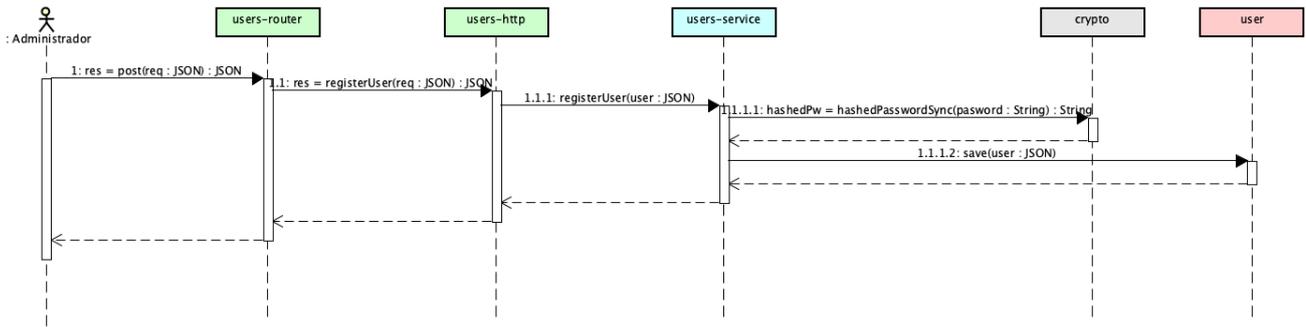


Figura 6.26: Registrar Usuario

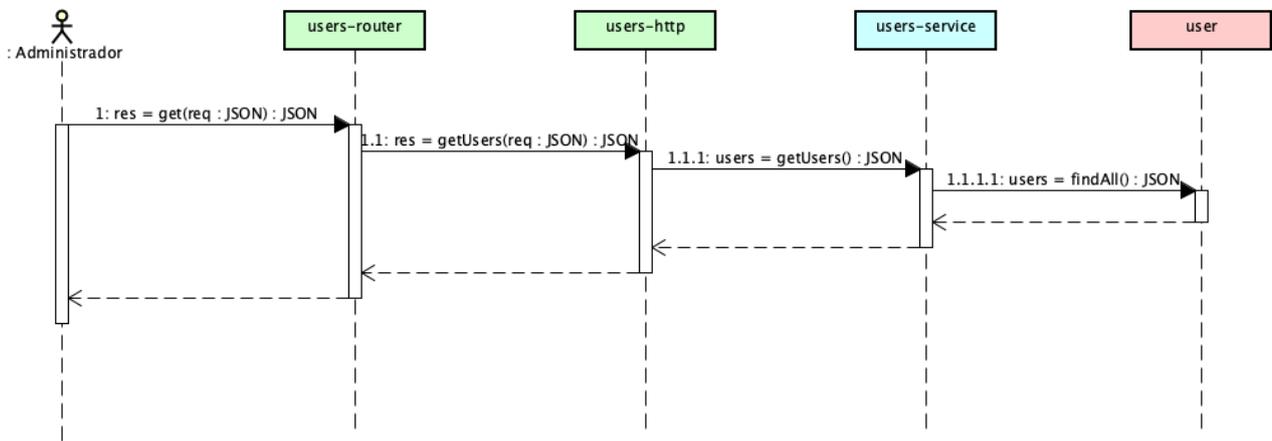


Figura 6.27: Visualizar usuarios del sistema

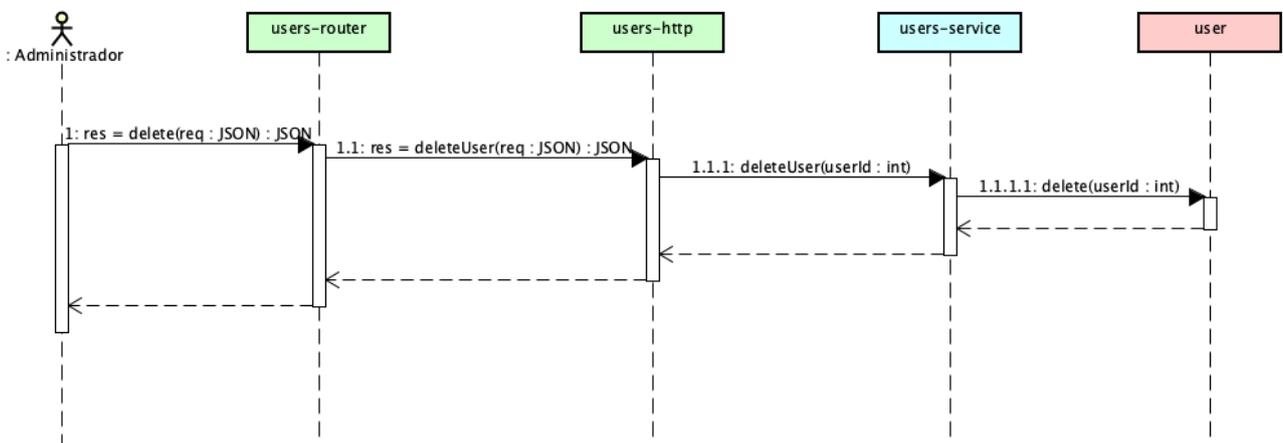


Figura 6.28: Eliminar usuarios del sistema

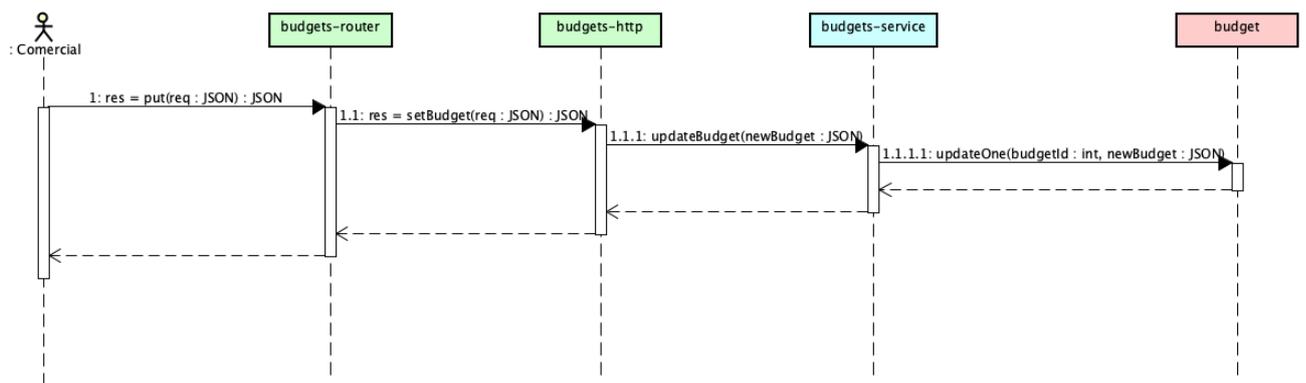


Figura 6.29: Editar Presupuesto

6.4.2. Diagramas de secuencia en diseño del Front-end

En ese apartado se mostrarán los diagramas de secuencia en diseño para los casos de uso que se han considerado más relevantes a la hora de ilustrar la funcionalidad del front-end del sistema. Se omitirán aquellos casos de uso que sean similares a los ya mostrados, para evitar la redundancia de los diagramas. Para diferenciar los distintos tipos de archivos, ya que el sistema no sigue una orientación a objetos, se ha decidido pintar de rosa los archivos correspondientes a componentes, en verde las clases que se comunican con la API, en amarillo los contextos y en azul las librerías externas utilizadas. Las interacciones entre las distintas líneas de vida reflejarán el paso de información entre los distintos componentes que forman el sistema.

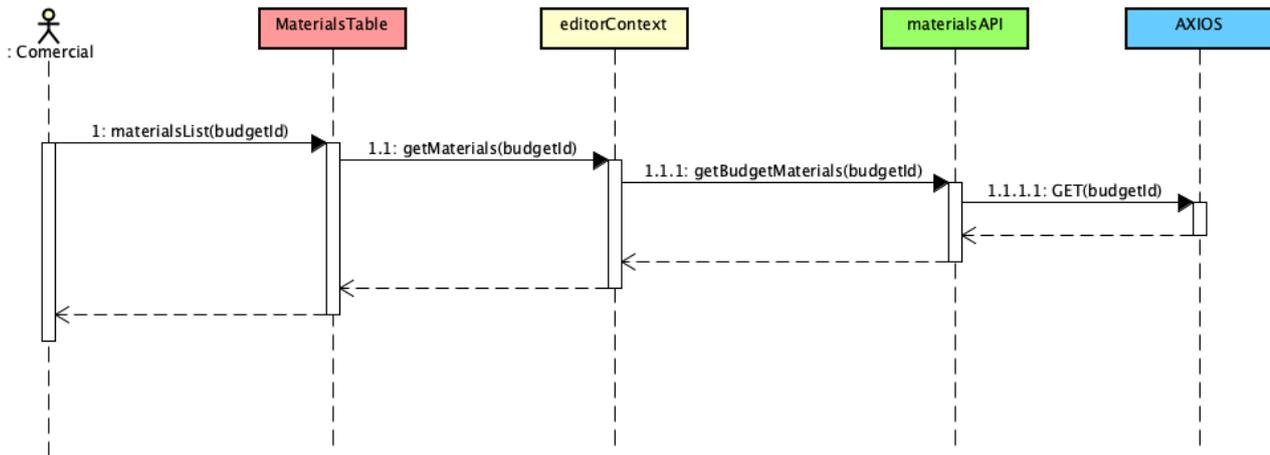


Figura 6.30: Ver Materiales de un Presupuesto

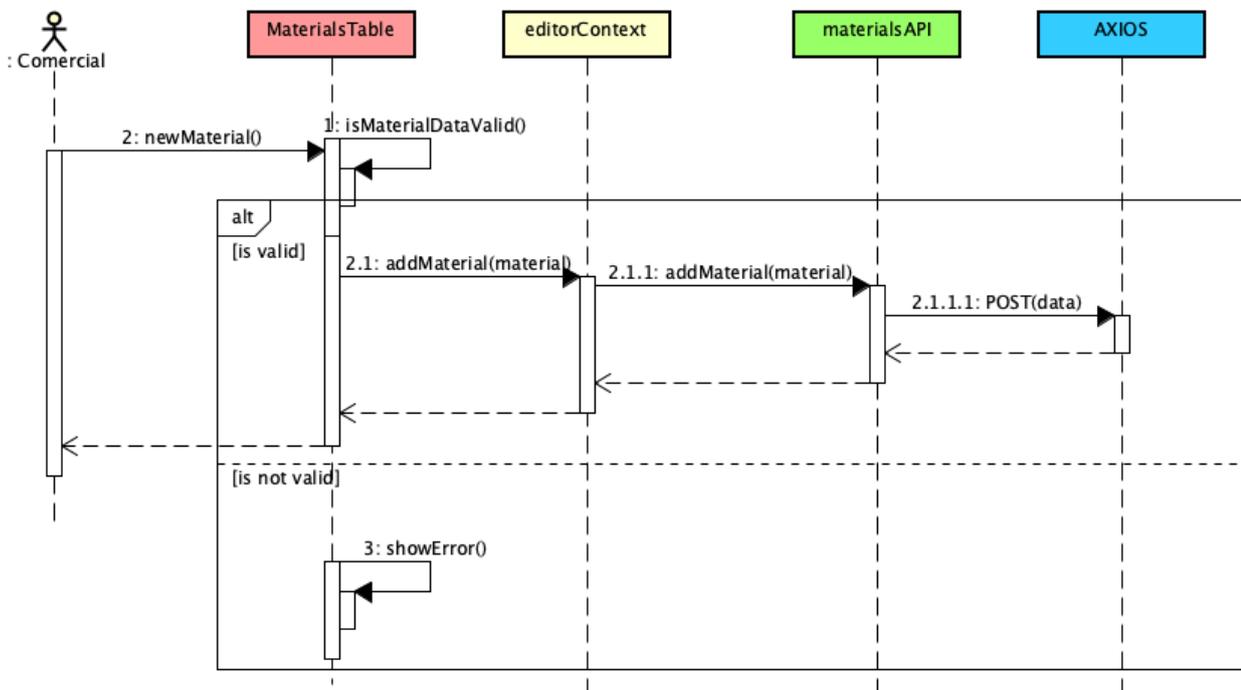


Figura 6.31: Añadir Material a un Presupuesto

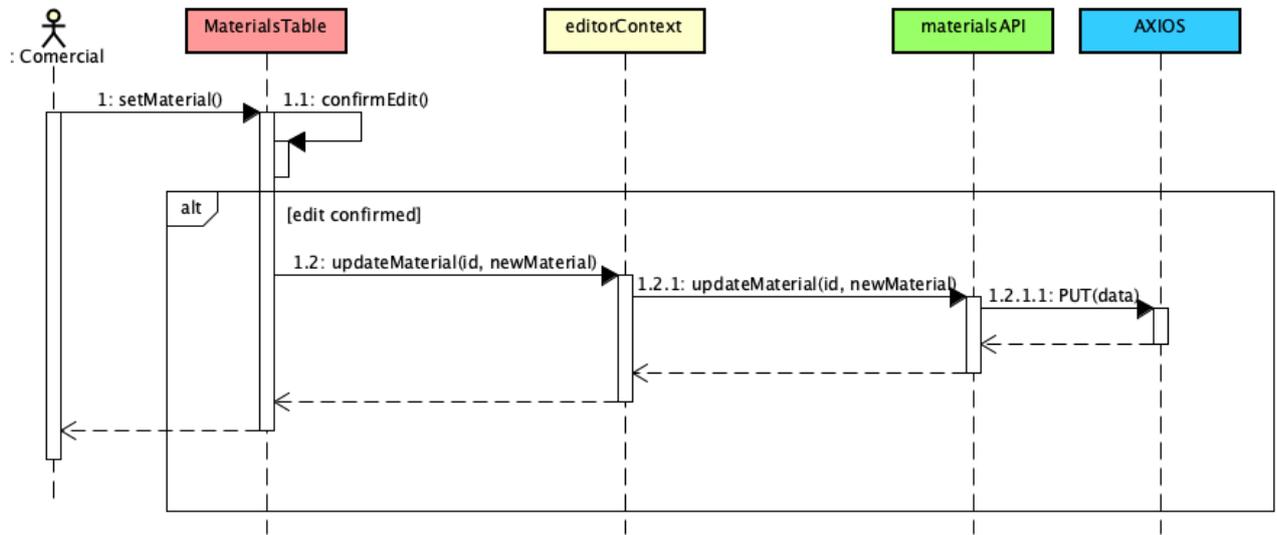


Figura 6.32: Editar un material

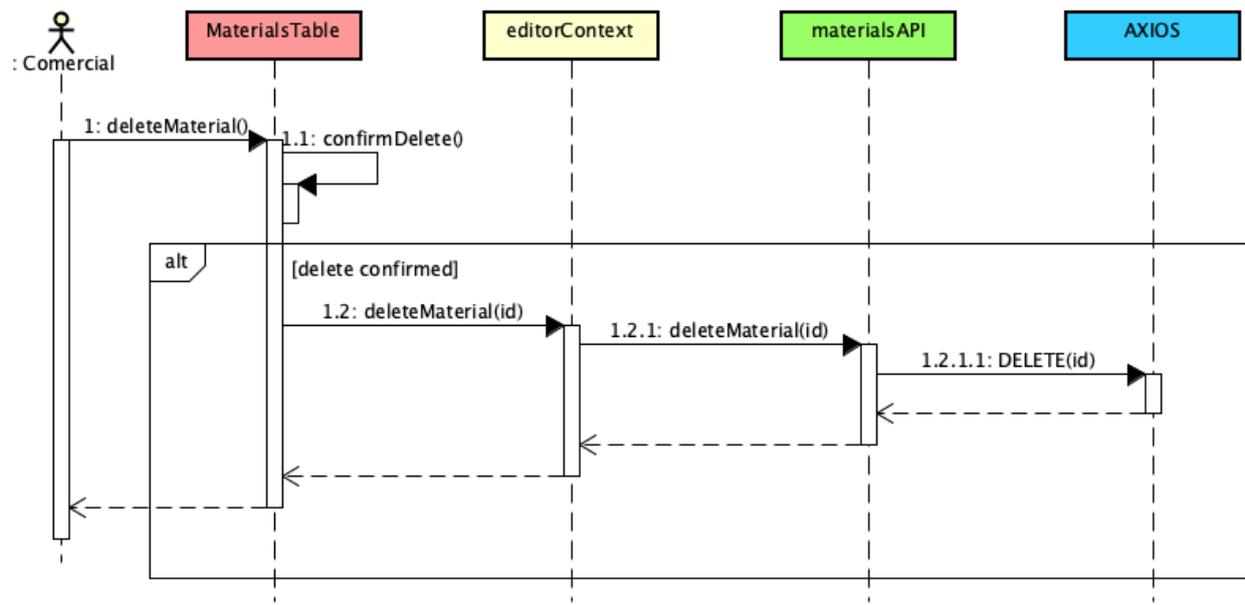


Figura 6.33: Eliminar un material

Capítulo 7

Implementación

7.1. Diagrama de despliegue

En el siguiente diagrama se puede observar el despliegue del sistema mediante contenedores de Docker [3].

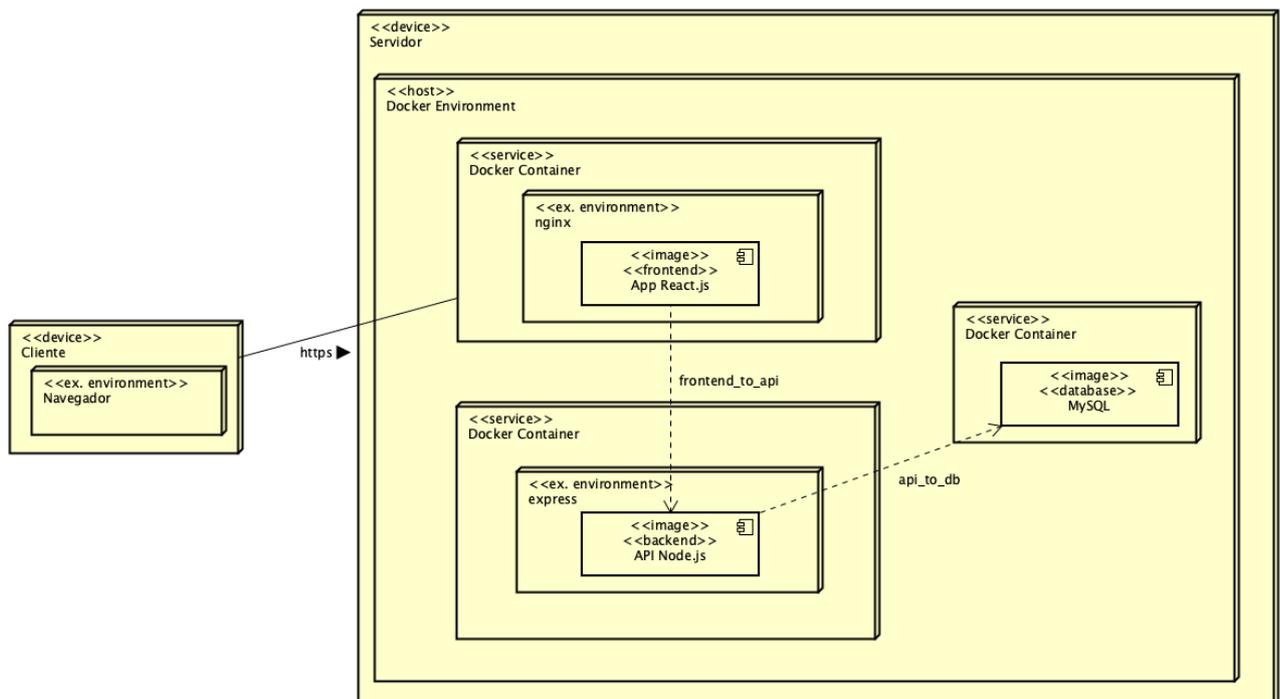


Figura 7.1: Diagrama de Despliegue de la aplicación

7.2. Entorno de desarrollo

Para el desarrollo tanto del back-end como del front-end se ha utilizado Visual Studio Code, un editor de texto gratuito desarrollado por Microsoft, altamente personalizable debido a la gran cantidad de extensiones que existen para el mismo, las cuales permiten adaptar el entorno a los gustos y necesidades del programador, dependiendo de la tarea que se vaya a realizar. Las extensiones utilizadas han sido varias relacionadas con JavaScript, Node.js y React.js.

7.3. Versiones necesarias de software

Al tratarse de una aplicación web, para acceder a ella sólo será necesario un equipo con conexión a Internet y uno de los siguientes navegadores:

- Google Chrome.
- Mozilla Firefox.
- Safari.
- Opera.
- Microsoft Edge.
- Internet Explorer.

7.4. Herramientas utilizadas

Para el desarrollo del sistema se han utilizado varias herramientas. A continuación se indicarán aquellas más relevantes, dividiendo la presente sección en herramientas utilizadas para la parte de servidor (back-end) y herramientas utilizadas para la interfaz web (front-end).

7.4.1. Front-end

- React.js: Biblioteca de JavaScript basada en componentes para construir interfaces de usuario. Permite crear componentes encapsulados que manejen su propio estado y se re-renderizarlos de forma apropiada cuando los datos de la aplicación cambien. [13]
- Material-UI: Proyecto open-source que consiste en una serie de componentes pre-diseñados para React que adaptan los estándares de diseño de Material Design de Google. [6]
- Axios: Cliente HTTP basado en promesas de JavaScript. Ha sido empleado como gestor para realizar las peticiones al API REST desarrollado para el Backend. [23]
- React-PDF: Librería open-source que contiene primitivas que permiten renderizar y crear archivos de tipo PDF para mostrarlos en una aplicación React y permitir al usuario visualizarlos y descargarlos. [12]

7.4.2. Back-end

- Node.js: Entorno en tiempo de ejecución de JavaScript para la capa del servidor, diseñado para crear aplicaciones network escalables. Es asíncrono, con E/S de datos y una arquitectura orientada a eventos. [24]
- Express: Es una infraestructura de aplicaciones web Node.js de código abierto diseñada para crear aplicaciones web y API. Se suele utilizar como servidor estándar para Node.js. [4]
- Passport.js: Middleware de autenticación para Node.js. Soporta un amplio número de estrategias para la autenticación de usuarios utilizando credenciales como usuario y contraseña, Twitter, Facebook, etc. [10]

- Sequelize: Es un ORM para Node.js basado en promesas compatible con MySQL, Postgres, SQLite y otros servicios de bases de datos. Simplifica la utilización de bases de datos relacionales, permitiendo realizar consultas, crear tablas, asociaciones, etc. [14]
- MySQL: Sistema de gestión de bases de datos relacionales de código abierto con un modelo cliente-servidor. [8]

7.4.3. Despliegue

Tanto para el front-end como para el back-end se ha utilizado Docker [3], un proyecto de código abierto que permite automatizar el despliegue de aplicaciones dentro de contenedores de software. Utiliza el kernel de Linux para segregar los procesos y permitir que estos se ejecuten de forma independiente, lo que proporciona una capa de abstracción y automatización que permite que un sistema sea portable y fácil de desplegar en cualquier máquina con esta tecnología instalada.

7.5. Problemas encontrados y soluciones aportadas

En esta sección se hablará sobre los distintos problemas que han surgido a lo largo del desarrollo, así como las soluciones aportadas para solventar cada uno de ellos.

7.5.1. Back-end

Los problemas más destacables que han surgido durante el desarrollo del servidor han sido los siguientes:

- **Problema:** Despliegue en Docker del back-end. Al levantar el contenedor, se caía inmediatamente por un error de compilación con una de las dependencias. El problema sucedía porque la compilación de dicha dependencia, llamada 'bcrypt' variaba entre sistemas operativos. Por tanto, se estaban copiando los node_modules del sistema operativo de origen, y a la hora de ejecutarlo en el entorno del contenedor fallaba.
- **Solución:** Crear un fichero .dockerignore al que añadir el directorio 'node_modules' y crear un script llamado 'docker-entrypoint.sh' para ejecutarlo al levantar el contenedor y que antes de arrancar el servidor desinstalase y volviese a instalar la dependencia conflictiva.

7.5.2. Front-end

Los problemas más destacables que han surgido durante el desarrollo de la aplicación web han sido los siguientes:

- **Problema:** Gestión de los tokens de usuario. Cuando un usuario iniciaba sesión en el sistema, dicha sesión no persistía en el tiempo, causando que el usuario tuviese que iniciar sesión con una mayor frecuencia de la que se consideraba óptima.
- **Solución:** Almacenar el token de sesión en el localStorage del navegador, para permitir que la sesión se mantuviese abierta en el navegador.

7.6. Control de versiones

Como control de versiones se ha utilizado git, concretamente en la página de GitLab, utilizando un repositorio privado para la parte de servidor y otro para la parte de la interfaz web.

El flujo de trabajo utilizado ha sido el marcado por Gitflow, basado principalmente en dos ramas, la rama

develop, a la que convergerán las distintas ramas de desarrollo que incluyan las distintas features, y la rama *master*, que será la rama principal o de producción, desde la que sólo se podrán crear otras ramas para arreglos de errores de forma rápida, también llamados *hotfixes*.

7.7. Servidor de la aplicación

En este apartado se explicará por separado la parte del servidor para front-end y para back-end.

7.7.1. Back-end

El API REST del servicio se encuentra desplegado en un contenedor Docker que contiene todas las dependencias necesarias y permite su comunicación con la base de datos y la aplicación web.

7.7.2. Front-end

La parte de la aplicación web se encuentra desplegada en un contenedor Docker que permite su comunicación con la API, utilizando nginx [20] como servidor web.

Capítulo 8

Plan de pruebas y evaluación

En este apartado se detallarán las pruebas a realizar para comprobar que todo lo implementado funciona de la forma esperada, y que por tanto se cumplen con todos los objetivos planteados.

Se llevarán a cabo pruebas unitarias que comprueben detalladamente los requisitos planteados en el capítulo 3, de forma que a través de ellas se verifique que tanto la interfaz web como el servidor desarrollados permiten realizar todas las funcionalidades de la forma esperada.

8.1. Pruebas para el back-end

En este apartado se detallarán las pruebas realizadas para comprobar el funcionamiento de la API REST desarrollada, comprobando el correcto funcionamiento de todas las operaciones CRUD implementadas para la gestión de presupuestos, usuarios, cargos, parámetros, materiales, servicios, viajes, artículos y tareas.

Las pruebas se realizarán utilizando Postman [25], una aplicación que sirve para realizar peticiones HTTP al servidor una vez desplegado. Para ello, simplemente hace falta conocer la URL de la API y las URL de los distintos endpoints de la misma. Además, habrá que tener en cuenta la estructura de los JSON que el servidor espera recibir, así como las cabeceras necesarias en las peticiones, para poder realizar las pruebas de forma exitosa. La URL para hacer las pruebas desplegando en local será **http://localhost:8080/**.

8.1.1. Pruebas sobre gestión de usuarios

En este apartado se realizarán las pruebas correspondientes al correcto funcionamiento de la gestión de usuarios del sistema, que incluyen la comprobación de los credenciales de usuario así como la creación, eliminación y visualización de los mismos (RF-01, RF-03, RF-04 y RF-05).

Prueba 1.1 (RF-01)	
Descripción	Esta prueba está destinada a comprobar el correcto funcionamiento de la autenticación de un usuario.
Endpoint	/login
Tipo de petición	POST
Parámetros del Body	username y password
Acción	Autenticar un usuario.
Resultado Esperado	El servidor contesta con un 200, y retorna el token de identificación del usuario, así como su id y su rol en el sistema.

Tabla 8.1: Prueba 1.1

Prueba 1.2 (RF-03)	
Descripción	Esta prueba está destinada a comprobar el correcto funcionamiento de la creación de un usuario.
Endpoint	/users
Tipo de petición	POST
Parámetros del Body	Nombre, username, password, email y rol.
Acción	Registrar un usuario en el sistema.
Resultado Esperado	El servidor contesta con un 201, indicando que se ha creado correctamente.

Tabla 8.2: Prueba 1.2

Prueba 1.3 (RF-04)	
Descripción	Esta prueba está destinada a comprobar el correcto funcionamiento del listado de los usuarios.
Endpoint	/users
Tipo de petición	GET
Parámetros del Body	-
Acción	Listar los usuarios existentes en el sistema.
Resultado Esperado	El servidor responde devolviendo un JSON que contiene todos los usuarios del sistema.

Tabla 8.3: Prueba 1.3

Prueba 1.4 (RF-05)	
Descripción	Esta prueba está destinada a comprobar el correcto funcionamiento de la eliminación de un usuario.
Endpoint	/users/userid
Tipo de petición	DELETE
Parámetros del Body	-
Acción	Eliminar un usuario indicando su id en la petición.
Resultado Esperado	El servidor contesta con un 200, indicando que se ha eliminado correctamente.

Tabla 8.4: Prueba 1.4

8.1.2. Pruebas sobre gestión de cargos

En este apartado se realizarán las pruebas correspondientes al correcto funcionamiento de la gestión de cargos del sistema, que incluyen la creación, edición, visualización y eliminación de los mismos (RF-08, RF-09, RF-10 y RF-11).

Prueba 2.1 (RF-08)	
Descripción	Esta prueba está destinada a comprobar el correcto funcionamiento del listado de los cargos del sistema.
Endpoint	/positions
Tipo de petición	GET
Parámetros del Body	-
Acción	Listar los cargos existentes en el sistema.
Resultado Esperado	El servidor contesta con un JSON que contiene todos los cargos existentes en el sistema.

Tabla 8.5: Prueba 2.1

Prueba 2.2 (RF-09)	
Descripción	Esta prueba está destinada a comprobar el correcto funcionamiento de la creación de un cargo.
Endpoint	/positions
Tipo de petición	POST
Parámetros del Body	Nombre, salario
Acción	Añadir un nuevo cargo.
Resultado Esperado	El servidor contesta con un 201, indicando que se ha creado correctamente.

Tabla 8.6: Prueba 2.2

Prueba 2.3 (RF-10)	
Descripción	Esta prueba está destinada a comprobar el correcto funcionamiento de la modificación de un cargo.
Endpoint	/positions/positionid
Tipo de petición	PUT
Parámetros del Body	Nombre y/o salario actualizados
Acción	Modificar un cargo existente, indicando su id en la petición y pasando los datos actualizados en el body.
Resultado Esperado	El servidor contesta con un 200, indicando que se ha modificado correctamente.

Tabla 8.7: Prueba 2.3

Prueba 2.4 (RF-11)	
Descripción	Esta prueba está destinada a comprobar el correcto funcionamiento de la eliminación de un cargo.
Endpoint	/positions/positionid
Tipo de petición	DELETE
Parámetros del Body	-
Acción	Eliminar un cargo concreto indicando su id en la petición.
Resultado Esperado	El servidor contesta con un 200, indicando que se ha eliminado correctamente.

Tabla 8.8: Prueba 2.4

8.1.3. Pruebas sobre gestión de parámetros

En este apartado se realizarán las pruebas correspondientes al correcto funcionamiento de la gestión de parámetros del sistema, que incluyen la creación, edición, visualización y eliminación de los mismos (RF-06 y RF-07).

Prueba 3.1 (RF-06)	
Descripción	Esta prueba está destinada a comprobar el correcto funcionamiento de la visualización de los parámetros actuales del sistema.
Endpoint	/parameters
Tipo de petición	GET
Parámetros del Body	-
Acción	Visualizar los parámetros más recientes registrados en el sistema.
Resultado Esperado	El servidor contesta con un JSON que contiene el ultimo set de parámetros registrados en el sistema.

Tabla 8.9: Prueba 3.1

Prueba 3.2 (RF-07)	
Descripción	Esta prueba está destinada a comprobar el correcto funcionamiento de la creación de set de parámetros.
Endpoint	/parameters
Tipo de petición	POST
Parámetros del Body	costes indirectos de viajes, materiales, servicios y MO, costes de dietas, hotel y precio del km
Acción	Añadir un nuevo set de parámetros al sistema.
Resultado Esperado	El servidor contesta con un 201, indicando que se ha creado correctamente.

Tabla 8.10: Prueba 3.2

8.1.4. Pruebas sobre gestión de artículos

En este apartado se realizarán las pruebas correspondientes al correcto funcionamiento de la gestión de artículos del sistema, que incluyen la creación, visualización, eliminación y visualización de un histórico de uso de los mismos (RF-22, RF-23, RF-24 y RF-25).

Prueba 4.1 (RF-22)	
Descripción	Esta prueba está destinada a comprobar el correcto funcionamiento del listado de los artículos del sistema.
Endpoint	/articles
Tipo de petición	GET
Parámetros del Body	-
Acción	Listar los artículos existentes en el sistema.
Resultado Esperado	El servidor contesta con un JSON que contiene todos los artículos existentes en el sistema.

Tabla 8.11: Prueba 4.1

Prueba 4.2 (RF-24)	
Descripción	Esta prueba está destinada a comprobar el correcto funcionamiento de la creación de un artículo.
Endpoint	/articles
Tipo de petición	POST
Parámetros del Body	Nombre, proveedor
Acción	Añadir un nuevo artículo.
Resultado Esperado	El servidor contesta con un 201, indicando que se ha creado correctamente.

Tabla 8.12: Prueba 4.2

Prueba 4.3 (RF-23)	
Descripción	Esta prueba está destinada a comprobar el correcto funcionamiento de la visualización de un listado de presupuestos donde se ha utilizado un artículo.
Endpoint	/articles/articleid/historic
Tipo de petición	GET
Parámetros del Body	-
Acción	Obtener el histórico de un artículo existente, indicando su id en la petición.
Resultado Esperado	El servidor contesta con un JSON que contiene un listado de los presupuestos en los que ese artículo ha sido utilizado.

Tabla 8.13: Prueba 4.3

Prueba 4.4 (RF-25)	
Descripción	Esta prueba está destinada a comprobar el correcto funcionamiento de la eliminación de un artículo.
Endpoint	/articles/articleid
Tipo de petición	DELETE
Parámetros del Body	-
Acción	Eliminar un artículo concreto indicando su id en la petición.
Resultado Esperado	El servidor contesta con un 200, indicando que se ha eliminado correctamente.

Tabla 8.14: Prueba 4.4

8.1.5. Pruebas sobre gestión de presupuestos

En este apartado se realizarán las pruebas correspondientes al correcto funcionamiento de la gestión de presupuestos del sistema, que incluyen la creación, visualización, edición, eliminación y duplicación de los mismos (RF-12, RF-19, RF-20, RF-42, RF-43 y RF-44).

Prueba 5.1 (RF-12)	
Descripción	Esta prueba está destinada a comprobar el correcto funcionamiento del listado de los presupuestos aprobados y pendientes de aprobación.
Endpoint	/budgets/filter/pendingandapproved
Tipo de petición	GET
Parámetros del Body	-
Acción	Listar los presupuestos aprobados y pendientes de aprobación del sistema.
Resultado Esperado	El servidor contesta con un JSON que contiene todos los presupuestos aprobados y pendientes de aprobación del sistema.

Tabla 8.15: Prueba 5.1

Prueba 5.2 (RF-19)	
Descripción	Esta prueba está destinada a comprobar el correcto funcionamiento de la creación de un presupuesto.
Endpoint	/budgets
Tipo de petición	POST
Parámetros del Body	Nombre, id del creador, id de los parámetros, estado, oferta del cliente, nombre del cliente, dirección del cliente, CIF del cliente, info. de contacto
Acción	Añadir un nuevo presupuesto.
Resultado Esperado	El servidor contesta con un 201, indicando que se ha creado correctamente.

Tabla 8.16: Prueba 5.2

Prueba 5.3 (RF-20)	
Descripción	Esta prueba está destinada a comprobar el correcto funcionamiento de la modificación de un presupuesto.
Endpoint	/budgets/budgetid
Tipo de petición	PUT
Parámetros del Body	Nombre, estado, oferta del cliente, nombre del cliente, dirección del cliente, CIF del cliente, info. de contacto actualizados
Acción	Modificar un presupuesto existente, indicando su id en la petición y pasando los datos actualizados en el body.
Resultado Esperado	El servidor contesta con un 200, indicando que se ha modificado correctamente.

Tabla 8.17: Prueba 5.3

Prueba 5.4 (RF-42)	
Descripción	Esta prueba está destinada a comprobar el correcto funcionamiento del listado de todos los presupuestos del sistema.
Endpoint	/budgets
Tipo de petición	GET
Parámetros del Body	-
Acción	Listar todos los presupuestos del sistema.
Resultado Esperado	El servidor contesta con un JSON que contiene todos los presupuestos del sistema.

Tabla 8.18: Prueba 5.4

Prueba 5.5 (RF-43)	
Descripción	Esta prueba está destinada a comprobar el correcto funcionamiento de la duplicación de un presupuesto.
Endpoint	/budgets/budgetid/duplicate
Tipo de petición	POST
Parámetros del Body	-
Acción	Crear una copia de un presupuesto del sistema.
Resultado Esperado	El servidor contesta con un 201, que indica que el presupuesto duplicado se ha creado correctamente.

Tabla 8.19: Prueba 5.5

Prueba 5.6 (RF-44)	
Descripción	Esta prueba está destinada a comprobar el correcto funcionamiento de la eliminación de un presupuesto.
Endpoint	/budgets/budgetid
Tipo de petición	DELETE
Parámetros del Body	-
Acción	Eliminar un presupuesto concreto indicando su id en la petición.
Resultado Esperado	El servidor contesta con un 200, indicando que se ha eliminado correctamente.

Tabla 8.20: Prueba 5.6

8.1.6. Pruebas sobre gestión de materiales

En este apartado se realizarán las pruebas correspondientes al correcto funcionamiento de la gestión de materiales de los presupuestos del sistema, que incluyen la creación, edición, visualización y eliminación de los mismos (RF-26, RF-27, RF-28 y RF-29).

Prueba 6.1 (RF-26)	
Descripción	Esta prueba está destinada a comprobar el correcto funcionamiento del listado de todos los materiales de un presupuesto.
Endpoint	/budgets/budgetid/materials
Tipo de petición	GET
Parámetros del Body	-
Acción	Listar todos los materiales de un presupuesto.
Resultado Esperado	El servidor contesta con un JSON que contiene todos los materiales del presupuesto cuyo id indicamos en la petición.

Tabla 8.21: Prueba 6.1

Prueba 6.2 (RF-27)	
Descripción	Esta prueba está destinada a comprobar el correcto funcionamiento de la creación de un material para un presupuesto.
Endpoint	/budgets/budgetid/materials
Tipo de petición	POST
Parámetros del Body	Id del artículo, precio, cantidad.
Acción	Crear un material asociado a un presupuesto del sistema.
Resultado Esperado	El servidor contesta con un 201, que indica que el material se ha creado correctamente.

Tabla 8.22: Prueba 6.2

Prueba 6.3 (RF-28)	
Descripción	Esta prueba está destinada a comprobar el correcto funcionamiento de la eliminación de un material asociado a un presupuesto.
Endpoint	/budgets/budgetid/materials/materialid
Tipo de petición	DELETE
Parámetros del Body	-
Acción	Eliminar un material concreto indicando su id y el id del presupuesto al que pertenece en la petición.
Resultado Esperado	El servidor contesta con un 200, indicando que se ha eliminado correctamente.

Tabla 8.23: Prueba 6.3

Prueba 6.4 (RF-29)	
Descripción	Esta prueba está destinada a comprobar el correcto funcionamiento de la modificación de un material.
Endpoint	/budgets/budgetid/materials/materialid
Tipo de petición	PUT
Parámetros del Body	Precio y cantidad actualizados
Acción	Modificar un material existente, indicando su id y el del presupuesto al que pertenece en la petición y pasando los datos actualizados en el body.
Resultado Esperado	El servidor contesta con un 200, indicando que se ha modificado correctamente.

Tabla 8.24: Prueba 6.4

8.1.7. Pruebas sobre gestión de servicios

En este apartado se realizarán las pruebas correspondientes al correcto funcionamiento de la gestión de servicios de presupuestos del sistema, que incluyen la creación, edición, visualización y eliminación de los mismos (RF-38, RF-39, RF-40 y RF-41).

Prueba 7.1 (RF-38)	
Descripción	Esta prueba está destinada a comprobar el correcto funcionamiento del listado de todos los servicios de un presupuesto.
Endpoint	/budgets/budgetid/services
Tipo de petición	GET
Parámetros del Body	-
Acción	Listar todos los servicios de un presupuesto.
Resultado Esperado	El servidor contesta con un JSON que contiene todos los servicios del presupuesto cuyo id indicamos en la petición.

Tabla 8.25: Prueba 7.1

Prueba 7.2 (RF-39)	
Descripción	Esta prueba está destinada a comprobar el correcto funcionamiento de la creación de un servicio para un presupuesto.
Endpoint	/budgets/budgetid/services
Tipo de petición	POST
Parámetros del Body	Concepto, tiempo, precio, cantidad.
Acción	Crear un servicio asociado a un presupuesto del sistema.
Resultado Esperado	El servidor contesta con un 201, que indica que el servicio se ha creado correctamente.

Tabla 8.26: Prueba 7.2

Prueba 7.3 (RF-40)	
Descripción	Esta prueba está destinada a comprobar el correcto funcionamiento de la eliminación de un servicio asociado a un presupuesto.
Endpoint	/budgets/budgetid/services/serviceid
Tipo de petición	DELETE
Parámetros del Body	-
Acción	Eliminar un servicio concreto indicando su id y el id del presupuesto al que pertenece en la petición.
Resultado Esperado	El servidor contesta con un 200, indicando que se ha eliminado correctamente.

Tabla 8.27: Prueba 7.3

Prueba 7.4 (RF-41)	
Descripción	Esta prueba está destinada a comprobar el correcto funcionamiento de la modificación de un servicio.
Endpoint	/budgets/budgetid/services/serviceid
Tipo de petición	PUT
Parámetros del Body	concepto, tiempo, precio y cantidad actualizados.
Acción	Modificar un servicio existente, indicando su id y el del presupuesto al que pertenece en la petición y pasando los datos actualizados en el body.
Resultado Esperado	El servidor contesta con un 200, indicando que se ha modificado correctamente.

Tabla 8.28: Prueba 7.4

8.1.8. Pruebas sobre gestión de tareas

En este apartado se realizarán las pruebas correspondientes al correcto funcionamiento de la gestión de tareas de presupuestos del sistema, que incluyen la creación, edición, visualización y eliminación de los mismos (RF-34, RF-35, RF-36 y RF-37).

Prueba 8.1 (RF-34)	
Descripción	Esta prueba está destinada a comprobar el correcto funcionamiento del listado de todas las tareas de un presupuesto.
Endpoint	/budgets/budgetid/tasks
Tipo de petición	GET
Parámetros del Body	-
Acción	Listar todas las tareas de un presupuesto.
Resultado Esperado	El servidor contesta con un JSON que contiene todas las tareas del presupuesto cuyo id indicamos en la petición.

Tabla 8.29: Prueba 8.1

Prueba 8.2 (RF-35)	
Descripción	Esta prueba está destinada a comprobar el correcto funcionamiento de la creación de una tarea para un presupuesto.
Endpoint	/budgets/budgetid/tasks
Tipo de petición	POST
Parámetros del Body	Concepto, unidades, incertidumbre, id del cargo, cantidad.
Acción	Crear una tarea asociada a un presupuesto del sistema.
Resultado Esperado	El servidor contesta con un 201, que indica que la tarea se ha creado correctamente.

Tabla 8.30: Prueba 8.2

Prueba 8.3 (RF-36)	
Descripción	Esta prueba está destinada a comprobar el correcto funcionamiento de la eliminación de una tarea asociada a un presupuesto.
Endpoint	/budgets/budgetid/tasks/taskid
Tipo de petición	DELETE
Parámetros del Body	-
Acción	Eliminar una tarea concreta indicando su id y el id del presupuesto al que pertenece en la petición.
Resultado Esperado	El servidor contesta con un 200, indicando que se ha eliminado correctamente.

Tabla 8.31: Prueba 8.3

Prueba 8.4 (RF-37)	
Descripción	Esta prueba está destinada a comprobar el correcto funcionamiento de la modificación de una tarea.
Endpoint	/budgets/budgetid/tasks/taskid
Tipo de petición	PUT
Parámetros del Body	Concepto, id del cargo, unidades, incertidumbre y cantidad actualizados.
Acción	Modificar una tarea existente, indicando su id y el del presupuesto al que pertenece en la petición y pasando los datos actualizados en el body.
Resultado Esperado	El servidor contesta con un 200, indicando que se ha modificado correctamente.

Tabla 8.32: Prueba 8.4

8.1.9. Pruebas sobre gestión de viajes

En este apartado se realizarán las pruebas correspondientes al correcto funcionamiento de la gestión de viajes de presupuestos del sistema, que incluyen la creación, edición, visualización y eliminación de los mismos (RF-30, RF-31, RF-32 y RF-33).

Prueba 9.1 (RF-30)	
Descripción	Esta prueba está destinada a comprobar el correcto funcionamiento del listado de todos los viajes de un presupuesto.
Endpoint	/budgets/budgetid/trips
Tipo de petición	GET
Parámetros del Body	-
Acción	Listar todos los viajes de un presupuesto.
Resultado Esperado	El servidor contesta con un JSON que contiene todos los viajes del presupuesto cuyo id indicamos en la petición.

Tabla 8.33: Prueba 9.1

Prueba 9.2 (RF-31)	
Descripción	Esta prueba está destinada a comprobar el correcto funcionamiento de la creación de un viaje para un presupuesto.
Endpoint	/budgets/budgetid/trips
Tipo de petición	POST
Parámetros del Body	Concepto, km, cantidad, días completos y parciales, coste de tren, avión y otros gastos.
Acción	Crear un viaje asociado a un presupuesto del sistema.
Resultado Esperado	El servidor contesta con un 201, que indica que el viaje se ha creado correctamente.

Tabla 8.34: Prueba 9.2

Prueba 9.3 (RF-33)	
Descripción	Esta prueba está destinada a comprobar el correcto funcionamiento de la eliminación de un viaje asociado a un presupuesto.
Endpoint	/budgets/budgetid/trips/tripid
Tipo de petición	DELETE
Parámetros del Body	-
Acción	Eliminar un viaje concreto indicando su id y el id del presupuesto al que pertenece en la petición.
Resultado Esperado	El servidor contesta con un 200, indicando que se ha eliminado correctamente.

Tabla 8.35: Prueba 9.3

Prueba 9.4 (RF-32)	
Descripción	Esta prueba está destinada a comprobar el correcto funcionamiento de la modificación de un viaje.
Endpoint	/budgets/budgetid/trips/tripid
Tipo de petición	PUT
Parámetros del Body	Concepto, km, cantidad, días completos y parciales, coste de tren, avión y otros gastos actualizados.
Acción	Modificar un viaje existente, indicando su id y el del presupuesto al que pertenece en la petición y pasando los datos actualizados en el body.
Resultado Esperado	El servidor contesta con un 200, indicando que se ha modificado correctamente.

Tabla 8.36: Prueba 9.4

8.2. Pruebas para el front-end

En este punto se detallarán las pruebas correspondientes a la parte de la interfaz del sistema desarrollado, detallando los pasos a dar para probar el correcto funcionamiento de la aplicación en base a los requisitos.

8.2.1. Pruebas sobre autenticación de usuarios

En este apartado se detallarán las pruebas correspondientes al inicio y cierre de sesión por parte de los usuarios (RF-01 y RF-02).

Prueba 10.1 (RF-01)	
Descripción	Esta prueba está destinada a comprobar el funcionamiento de inicio de sesión del usuario en el sistema.
Acción	Introducir las credenciales del usuario en la ventana de inicio de sesión.
Resultado Esperado	El usuario ha conseguido identificarse correctamente en el sistema y está en la vista de dashboard, que dependerá de su rol en el sistema.

Tabla 8.37: Prueba 10.1

Prueba 10.2 (RF-02)	
Descripción	Esta prueba está destinada a comprobar el funcionamiento de cierre de sesión del usuario en el sistema.
Acción	Hacer click en el botón de cerrar sesión que se encuentra en la barra lateral de la interfaz.
Resultado Esperado	El usuario ha terminado su sesión y se encuentra en la página de inicio de sesión.

Tabla 8.38: Prueba 10.2

8.2.2. Pruebas sobre gestión de usuarios

En este apartado se realizarán las pruebas correspondientes al correcto funcionamiento de la gestión de usuarios del sistema, que incluyen la creación, eliminación y visualización de los mismos (RF-03, RF-04 y RF-05).

Prueba 11.1 (RF-03)	
Descripción	Esta prueba está destinada a comprobar el funcionamiento de la creación de usuarios en el sistema.
Acción	Introducir el nombre, el nombre de usuario, el correo electrónico y el rol en el formulario de registro que se encuentra en la pantalla principal del administrador. Pulsar el botón de registrar.
Resultado Esperado	El usuario aparece en la lista de usuarios del sistema.

Tabla 8.39: Prueba 11.1

Prueba 11.2 (RF-04)	
Descripción	Esta prueba está destinada a comprobar el funcionamiento del listado de los usuarios del sistema.
Acción	Iniciar sesión en el sistema con un usuario en el rol de administrador.
Resultado Esperado	El usuario ha accedido a su página principal y puede visualizar un listado de los usuarios del sistema.

Tabla 8.40: Prueba 11.2

Prueba 11.3 (RF-05)	
Descripción	Esta prueba está destinada a comprobar el funcionamiento de la eliminación de un usuario del sistema.
Acción	Hacer click en el icono de eliminar al lado del nombre de un usuario en el listado de usuarios del sistema. Pulsar eliminar en el dialogo de confirmación.
Resultado Esperado	El usuario ha desaparecido del listado de usuarios del sistema.

Tabla 8.41: Prueba 11.3

8.2.3. Pruebas sobre gestión de cargos

En este apartado se realizarán las pruebas correspondientes al correcto funcionamiento de la gestión de cargos del sistema, que incluyen la creación, edición, visualización y eliminación de los mismos (RF-08, RF-09, RF-10 y RF-11).

Prueba 12.1 (RF-08)	
Descripción	Esta prueba está destinada a comprobar el funcionamiento de la visualización del listado de los perfiles profesionales o cargos del sistema.
Acción	Hacer click en el botón de la barra lateral "Cargos".
Resultado Esperado	El usuario puede visualizar un listado de los cargos actuales del sistema.

Tabla 8.42: Prueba 12.1

Prueba 12.2 (RF-09)	
Descripción	Esta prueba está destinada a comprobar el funcionamiento de añadir cargos nuevos al sistema.
Acción	Hacer click en el botón de la barra lateral "Cargos". Pulsar el botón de añadir. Rellenar el formulario y confirmar.
Resultado Esperado	El cargo nuevo aparece en la lista de cargos del sistema.

Tabla 8.43: Prueba 12.2

Prueba 12.3 (RF-10)	
Descripción	Esta prueba está destinada a comprobar el funcionamiento de la modificación de un cargo del sistema.
Acción	Hacer click en el icono de editar al lado del cargo en el listado de cargos del sistema. Rellenar el formulario con los datos nuevos y confirmar.
Resultado Esperado	Los datos del cargo se muestran modificados en el listado de cargos del sistema.

Tabla 8.44: Prueba 12.3

Prueba 12.4 (RF-11)	
Descripción	Esta prueba está destinada a comprobar el funcionamiento de la eliminación de un cargo del sistema.
Acción	Hacer click en el icono de eliminar al lado del cargo en el listado de cargos del sistema. Pulsar el botón de eliminar en el diálogo de confirmación.
Resultado Esperado	El cargo ya no aparece en el listado de cargos del sistema.

Tabla 8.45: Prueba 12.4

8.2.4. Pruebas sobre gestión de parámetros

En este apartado se realizarán las pruebas correspondientes al correcto funcionamiento de la gestión de parámetros del sistema, que incluyen la creación y visualización de los mismos (RF-06 y RF-07).

Prueba 13.1 (RF-06)	
Descripción	Esta prueba está destinada a comprobar el funcionamiento de la visualización de los últimos parámetros del sistema.
Acción	Hacer click en el botón de la barra lateral "Parámetros".
Resultado Esperado	El usuario puede visualizar los últimos parámetros con los que cuenta el sistema.

Tabla 8.46: Prueba 13.1

Prueba 13.2 (RF-07)	
Descripción	Esta prueba está destinada a comprobar el funcionamiento de añadir parámetros nuevos al sistema.
Acción	Hacer click en el botón de la barra lateral "Parámetros". Editar los valores necesarios en el formulario de la pantalla y pulsar el botón de confirmar.
Resultado Esperado	Los parámetros del sistema cambian en la pantalla. Todos los presupuestos en estado editable ahora utilizarán esos parámetros.

Tabla 8.47: Prueba 13.2

8.2.5. Pruebas sobre gestión de artículos

En este apartado se realizarán las pruebas correspondientes al correcto funcionamiento de la gestión de artículos del sistema, que incluyen la creación, visualización, eliminación y visualización de un histórico de uso de los mismos (RF-22, RF-23, RF-24 y RF-25).

Prueba 14.1 (RF-23)	
Descripción	Esta prueba está destinada a comprobar el funcionamiento de la visualización del histórico de presupuestos en los que se ha utilizado un material.
Acción	Pulsar el botón de edición de un presupuesto o crear uno nuevo. En la vista de editor de presupuestos, pulsar el botón de histórico de al lado de un artículo, que se encuentra en la tabla con histórico de artículos de sistema, en la sección de materiales.
Resultado Esperado	El usuario puede visualizar un listado de todos los presupuestos en los que ese artículo ha sido utilizado previamente.

Tabla 8.48: Prueba 14.1

Prueba 14.2 (RF-24)	
Descripción	Esta prueba está destinada a comprobar el funcionamiento de añadir artículos nuevos al sistema.
Acción	En la sección de materiales del editor de presupuestos, hacer click en el botón de añadir en la tabla del histórico de artículos del sistema. Rellenar el formulario y pulsar el botón de confirmar.
Resultado Esperado	El artículo nuevo aparece en la lista de artículos del sistema.

Tabla 8.49: Prueba 14.2

Prueba 14.3 (RF-22)	
Descripción	Esta prueba está destinada a comprobar el funcionamiento de la visualización de artículos del sistema.
Acción	Acceder al editor de presupuestos. Navegar hasta la sección de materiales.
Resultado Esperado	El usuario puede visualizar una tabla con un listado del histórico de artículos del sistema.

Tabla 8.50: Prueba 14.3

Prueba 14.4 (RF-25)	
Descripción	Esta prueba está destinada a comprobar el funcionamiento de la eliminación de un artículo del sistema.
Acción	Hacer click en el icono de eliminar al lado del artículo en el listado de artículos del sistema. Pulsar el botón de eliminar en el diálogo de confirmación.
Resultado Esperado	El artículo ya no aparece en el listado de artículos del sistema.

Tabla 8.51: Prueba 14.4

8.2.6. Pruebas sobre gestión de presupuestos

En este apartado se realizarán las pruebas correspondientes al correcto funcionamiento de la gestión de presupuestos del sistema, que incluyen la creación, visualización, edición, eliminación, ordenación, duplicación y visualización en PDF de los mismos (RF-12, RF-13, RF-14, RF-15, RF-16, RF-17, RF-18, RF-19, RF-20, RF-21, RF-42, RF-43, RF-44, RF-45, RF-46, RF-47, RF-48 y RF-49).

Prueba 15.1 (RF-12)	
Descripción	Esta prueba está destinada a comprobar el funcionamiento de la visualización de un listado con los presupuestos aprobados y pendientes de aprobación del sistema.
Acción	Iniciar sesión en el sistema con el rol de jefe.
Resultado Esperado	El usuario visualizará un listado con los presupuestos aprobados y pendientes de aprobación del sistema.

Tabla 8.52: Prueba 15.1

Prueba 15.2 (RF-13)	
Descripción	Esta prueba está destinada a comprobar el funcionamiento de la aprobación del presupuesto del sistema.
Acción	Hacer click en el icono de aprobación al lado del presupuesto pendiente de aprobación en el listado de presupuestos aprobados y pendientes de aprobación del sistema.
Resultado Esperado	El estado del presupuesto ha pasado a ser ".aprobado".

Tabla 8.53: Prueba 15.2

Prueba 15.3 (RF-14)	
Descripción	Esta prueba está destinada a comprobar el funcionamiento del rechazo de un presupuesto del sistema.
Acción	Hacer click en el icono de rechazo al lado del presupuesto pendiente de aprobación en el listado de presupuestos aprobados y pendientes de aprobación del sistema.
Resultado Esperado	El estado del presupuesto ha pasado a ser ".editable". Ya no se muestra en el listado.

Tabla 8.54: Prueba 15.3

Prueba 15.4 (RF-15)	
Descripción	Esta prueba está destinada a comprobar el funcionamiento de la visualización del archivo de un presupuesto del sistema.
Acción	Hacer click en el icono de descarga al lado del presupuesto pendiente de aprobación en el listado de presupuestos aprobados y pendientes de aprobación del sistema.
Resultado Esperado	El usuario puede visualizar un archivo PDF con los datos del presupuesto.

Tabla 8.55: Prueba 15.4

Prueba 15.5 (RF-16)	
Descripción	Esta prueba está destinada a comprobar el funcionamiento de la ordenación del listado de presupuestos por nombre.
Acción	Hacer click en el la cabecera de la tabla de la columna de nombres en el listado de presupuestos aprobados y pendientes de aprobación del sistema.
Resultado Esperado	Los presupuestos pasan a estar ordenados por nombre en la tabla.

Tabla 8.56: Prueba 15.5

Prueba 15.6 (RF-17)	
Descripción	Esta prueba está destinada a comprobar el funcionamiento de la ordenación del listado de presupuestos por estado.
Acción	Hacer click en el la cabecera de la tabla de la columna de estados en el listado de presupuestos aprobados y pendientes de aprobación del sistema.
Resultado Esperado	Los presupuestos pasan a estar ordenados por estado en la tabla.

Tabla 8.57: Prueba 15.6

Prueba 15.7 (RF-18)	
Descripción	Esta prueba está destinada a comprobar el funcionamiento de la ordenación del listado de presupuestos por autor.
Acción	Hacer click en el la cabecera de la tabla de la columna de autores en el listado de presupuestos aprobados y pendientes de aprobación del sistema.
Resultado Esperado	Los presupuestos pasan a estar ordenados por autor en la tabla.

Tabla 8.58: Prueba 15.7

Prueba 15.8 (RF-19)	
Descripción	Esta prueba está destinada a comprobar el funcionamiento de la creación de presupuestos.
Acción	Iniciar sesión en el sistema con rol de comercial. Introducir un nombre para el presupuesto en el formulario de la pantalla principal y pulsar el botón de confirmación.
Resultado Esperado	El presupuesto ha sido creado y se puede modificar desde el editor de presupuestos.

Tabla 8.59: Prueba 15.8

Prueba 15.9 (RF-20)	
Descripción	Esta prueba está destinada a comprobar el funcionamiento de la edición de presupuestos.
Acción	Iniciar sesión en el sistema con rol de comercial. Pulsar el botón de edición al lado del nombre de presupuesto editable en la lista de presupuestos. Cambiar los valores necesarios y pulsar el botón de "guardar y salir".
Resultado Esperado	El usuario ha editado el presupuesto con éxito.

Tabla 8.60: Prueba 15.9

Prueba 15.10 (RF-21)	
Descripción	Esta prueba está destinada a comprobar el funcionamiento de la edición de presupuestos.
Acción	Iniciar sesión en el sistema con rol de comercial. Pulsar el botón de edición al lado del nombre de presupuesto editable en la lista de presupuestos. Pulsar el botón de "guardar y finalizar".
Resultado Esperado	El estado del presupuesto ha cambiado a "pendiente de aprobación".

Tabla 8.61: Prueba 15.10

Prueba 15.11 (RF-42)	
Descripción	Esta prueba está destinada a comprobar el funcionamiento de la visualización de un listado con todos los presupuestos del sistema.
Acción	Iniciar sesión en el sistema con el rol de comercial.
Resultado Esperado	El usuario visualizará un listado con todos los presupuestos del sistema.

Tabla 8.62: Prueba 15.11

Prueba 15.12 (RF-43)	
Descripción	Esta prueba está destinada a comprobar el funcionamiento de la duplicación de un presupuesto.
Acción	Iniciar sesión en el sistema con el rol de comercial. Pulsar el botón de duplicar al lado del nombre del presupuesto que se quiera duplicar.
Resultado Esperado	El presupuesto duplicado aparece en la lista de presupuestos del sistema.

Tabla 8.63: Prueba 15.12

Prueba 15.13 (RF-44)	
Descripción	Esta prueba está destinada a comprobar el funcionamiento de la eliminación de un presupuesto.
Acción	Iniciar sesión en el sistema con el rol de comercial. Pulsar el botón de eliminar al lado del nombre del presupuesto que se quiera eliminar. Pulsar confirmar en el cuadro de diálogo.
Resultado Esperado	El presupuesto ya no aparece en la lista de presupuestos del sistema.

Tabla 8.64: Prueba 15.13

Prueba 15.14 (RF-45 y RF-46)	
Descripción	Esta prueba está destinada a comprobar el funcionamiento de la visualización del archivo de un presupuesto del sistema.
Acción	Hacer click en el icono de descarga al lado del presupuesto en el listado de presupuestos del sistema.
Resultado Esperado	El usuario puede visualizar un archivo PDF con los datos del presupuesto.

Tabla 8.65: Prueba 15.14

Prueba 15.15 (RF-47)	
Descripción	Esta prueba está destinada a comprobar el funcionamiento de la ordenación del listado de presupuestos por nombre.
Acción	Hacer click en el la cabecera de la tabla de la columna de nombres en el listado de presupuestos del sistema.
Resultado Esperado	Los presupuestos pasan a estar ordenados por nombre en la tabla.

Tabla 8.66: Prueba 15.15

Prueba 15.16 (RF-48)	
Descripción	Esta prueba está destinada a comprobar el funcionamiento de la ordenación del listado de presupuestos por estado.
Acción	Hacer click en el la cabecera de la tabla de la columna de estados en el listado de presupuestos del sistema.
Resultado Esperado	Los presupuestos pasan a estar ordenados por estado en la tabla.

Tabla 8.67: Prueba 15.16

Prueba 15.17 (RF-49)	
Descripción	Esta prueba está destinada a comprobar el funcionamiento de la ordenación del listado de presupuestos por autor.
Acción	Hacer click en el la cabecera de la tabla de la columna de autores en el listado de presupuestos del sistema.
Resultado Esperado	Los presupuestos pasan a estar ordenados por autor en la tabla.

Tabla 8.68: Prueba 15.17

8.2.7. Pruebas sobre gestión de materiales

En este apartado se realizarán las pruebas correspondientes al correcto funcionamiento de la gestión de materiales de los presupuestos del sistema, que incluyen la creación, edición, visualización y eliminación de los mismos (RF-26, RF-27, RF-28 y RF-29).

Prueba 16.1 (RF-26)	
Descripción	Esta prueba está destinada a comprobar el funcionamiento de la visualización del listado de los materiales asociados a un presupuesto.
Acción	En el editor de presupuestos, hacer click en la pestaña de "Materiales".
Resultado Esperado	El usuario puede visualizar un listado de los materiales asociados al presupuesto.

Tabla 8.69: Prueba 16.1

Prueba 16.2 (RF-27)	
Descripción	Esta prueba está destinada a comprobar el funcionamiento de añadir materiales a un presupuesto.
Acción	En el editor de presupuestos, hacer click en la pestaña de "Materiales". Pulsar el botón de añadir al lado del artículo que se desee escoger en la tabla de artículos. Rellenar el formulario y confirmar.
Resultado Esperado	El material nuevo aparece en la lista de materiales del presupuesto.

Tabla 8.70: Prueba 16.2

Prueba 16.3 (RF-29)	
Descripción	Esta prueba está destinada a comprobar el funcionamiento de la modificación de un material del presupuesto.
Acción	Hacer click en el icono de editar al lado del material en el listado de materiales del presupuesto. Rellenar el formulario con los datos nuevos y confirmar.
Resultado Esperado	Los datos del material se muestran modificados en el listado de materiales del presupuesto.

Tabla 8.71: Prueba 16.3

Prueba 16.4 (RF-28)	
Descripción	Esta prueba está destinada a comprobar el funcionamiento de la eliminación de un material de un presupuesto.
Acción	Hacer click en el icono de eliminar al lado del material en el listado de materiales del presupuesto. Pulsar el botón de eliminar en el diálogo de confirmación.
Resultado Esperado	El material ya no aparece en el listado de materiales del presupuesto.

Tabla 8.72: Prueba 16.4

8.2.8. Pruebas sobre gestión de servicios

En este apartado se realizarán las pruebas correspondientes al correcto funcionamiento de la gestión de servicios de presupuestos del sistema, que incluyen la creación, edición, visualización y eliminación de los mismos (RF-38, RF-39, RF-40 y RF-41).

Prueba 17.1 (RF-38)	
Descripción	Esta prueba está destinada a comprobar el funcionamiento de la visualización del listado de los servicios asociados a un presupuesto.
Acción	En el editor de presupuestos, hacer click en la pestaña de "Servicios".
Resultado Esperado	El usuario puede visualizar un listado de los servicios asociados al presupuesto.

Tabla 8.73: Prueba 17.1

Prueba 17.2 (RF-39)	
Descripción	Esta prueba está destinada a comprobar el funcionamiento de añadir servicios a un presupuesto.
Acción	En el editor de presupuestos, hacer click en la pestaña de "Servicios". Pulsar el botón de añadir en la tabla de servicios. Rellenar el formulario y confirmar.
Resultado Esperado	El servicio nuevo aparece en la lista de servicios del presupuesto.

Tabla 8.74: Prueba 17.2

Prueba 17.3 (RF-41)	
Descripción	Esta prueba está destinada a comprobar el funcionamiento de la modificación de un servicio del presupuesto.
Acción	Hacer click en el icono de editar al lado del servicio en el listado de servicios del presupuesto. Rellenar el formulario con los datos nuevos y confirmar.
Resultado Esperado	Los datos del servicio se muestran modificados en el listado de servicios del presupuesto.

Tabla 8.75: Prueba 17.3

Prueba 17.4 (RF-40)	
Descripción	Esta prueba está destinada a comprobar el funcionamiento de la eliminación de un servicio de un presupuesto.
Acción	Hacer click en el icono de eliminar al lado del servicio en el listado de servicios del presupuesto. Pulsar el botón de eliminar en el diálogo de confirmación.
Resultado Esperado	El servicio ya no aparece en el listado de servicios del presupuesto.

Tabla 8.76: Prueba 17.4

8.2.9. Pruebas sobre gestión de tareas

En este apartado se realizarán las pruebas correspondientes al correcto funcionamiento de la gestión de tareas de presupuestos del sistema, que incluyen la creación, edición, visualización y eliminación de los mismos (RF-34, RF-35, RF-36 y RF-37).

Prueba 18.1 (RF-34)	
Descripción	Esta prueba está destinada a comprobar el funcionamiento de la visualización del listado de las tareas asociadas a un presupuesto.
Acción	En el editor de presupuestos, hacer click en la pestaña de "Tareas".
Resultado Esperado	El usuario puede visualizar un listado de las tareas asociadas al presupuesto.

Tabla 8.77: Prueba 18.1

Prueba 18.2 (RF-35)	
Descripción	Esta prueba está destinada a comprobar el funcionamiento de añadir tareas a un presupuesto.
Acción	En el editor de presupuestos, hacer click en la pestaña de "Tareas". Pulsar el botón de añadir en la tabla de tareas. Rellenar el formulario y confirmar.
Resultado Esperado	La tarea nueva aparece en la lista de tareas del presupuesto.

Tabla 8.78: Prueba 18.2

Prueba 18.3 (RF-37)	
Descripción	Esta prueba está destinada a comprobar el funcionamiento de la modificación de una tarea del presupuesto.
Acción	Hacer click en el icono de editar al lado de la tarea en el listado de tareas del presupuesto. Rellenar el formulario con los datos nuevos y confirmar.
Resultado Esperado	Los datos de la tarea se muestran modificados en el listado de tareas del presupuesto.

Tabla 8.79: Prueba 18.3

Prueba 18.4 (RF-36)	
Descripción	Esta prueba está destinada a comprobar el funcionamiento de la eliminación de una tarea de un presupuesto.
Acción	Hacer click en el icono de eliminar al lado de la tarea en el listado de tareas del presupuesto. Pulsar el botón de eliminar en el diálogo de confirmación.
Resultado Esperado	La tarea ya no aparece en el listado de tareas del presupuesto.

Tabla 8.80: Prueba 18.4

8.2.10. Pruebas sobre gestión de viajes

En este apartado se realizarán las pruebas correspondientes al correcto funcionamiento de la gestión de viajes de presupuestos del sistema, que incluyen la creación, edición, visualización y eliminación de los mismos (RF-30, RF-31, RF-32 y RF-33).

Prueba 19.1 (RF-30)	
Descripción	Esta prueba está destinada a comprobar el funcionamiento de la visualización del listado de los viajes asociados a un presupuesto.
Acción	En el editor de presupuestos, hacer click en la pestaña de "Viajes".
Resultado Esperado	El usuario puede visualizar un listado de los viajes asociados al presupuesto.

Tabla 8.81: Prueba 19.1

Prueba 19.2 (RF-31)	
Descripción	Esta prueba está destinada a comprobar el funcionamiento de añadir viajes a un presupuesto.
Acción	En el editor de presupuestos, hacer click en la pestaña de "Viajes". Pulsar el botón de añadir en la tabla de viajes. Rellenar el formulario y confirmar.
Resultado Esperado	El viaje nuevo aparece en la lista de viajes del presupuesto.

Tabla 8.82: Prueba 19.2

Prueba 19.3 (RF-32)	
Descripción	Esta prueba está destinada a comprobar el funcionamiento de la modificación de un viaje del presupuesto.
Acción	Hacer click en el icono de editar al lado del viaje en el listado de viajes del presupuesto. Rellenar el formulario con los datos nuevos y confirmar.
Resultado Esperado	Los datos del viaje se muestran modificados en el listado de viajes del presupuesto.

Tabla 8.83: Prueba 19.3

Prueba 19.4 (RF-33)	
Descripción	Esta prueba está destinada a comprobar el funcionamiento de la eliminación de un viaje de un presupuesto.
Acción	Hacer click en el icono de eliminar al lado del viaje en el listado de viajes del presupuesto. Pulsar el botón de eliminar en el diálogo de confirmación.
Resultado Esperado	El viaje ya no aparece en el listado de viajes del presupuesto.

Tabla 8.84: Prueba 19.4

Capítulo 9

Manual de programador

En este capítulo se detallará la información necesaria para un desarrollador que quiera trabajar en el sistema implementado, tanto para su mantenimiento como para su ampliación. Se incluirá toda la información necesaria tanto para el back-end como para el front-end.

9.1. Back-end

En este punto se explicará todo lo necesario acerca de la estructura del back-end del sistema, incluyendo una explicación detallada de cada una de las carpetas y ficheros de interés. Al final de la sección se incluirá una breve explicación de como realizar peticiones HTTP con Postman [25], herramienta utilizada para probar la API REST desarrollada.

9.1.1. Directivas generales

El código fuente desarrollado se divide en cuatro subcarpetas: 'models', 'routes', 'services' y 'tools'.

- **Carpeta routes:** Contiene los distintos endpoints que tiene el sistema. Se encarga de recibir y gestionar las peticiones que la API recibe, así como dar formato y enviar las respuestas pertinentes en formato JSON y con un código HTTP de respuesta en función de la operación realizada.
- **Carpeta services:** Contiene la lógica de negocio de la aplicación. Es decir, las funcionalidades necesarias para operar con la BBDD del sistema (el equivalente a las *queries*, pero utilizando Sequelize como ORM).
- **Carpeta models:** Contiene los modelos de dominio de las entidades del sistema necesarios para la comunicación con la BBDD. Para cada entidad se definen los campos correspondientes a cada una de las columnas de las tablas correspondientes en la BBDD.
- **Carpeta tools:** Contiene una serie de ficheros auxiliares en los que se apoya el sistema para su funcionamiento, incluyendo utilidades para la autenticación de usuarios y para la gestión de promesas Javascript.

9.1.2. Carpeta routes

En la carpeta router se encuentran los endpoints a los que la aplicación realizará peticiones. Esta carpeta se divide en 10 subcarpetas, es decir, una para cada entidad del sistema, y una adicional para aislar el endpoint referente a la autenticación de los usuarios del sistema. Dentro de cada una de esas subcarpetas encontramos dos ficheros, el `httpHandler` y el `router`.

- **Router:** Define los distintos endpoints relacionados con esa entidad del dominio. Redirige las peticiones al `Handler` correspondiente.
- **Handler:** Recibe las peticiones de los distintos endpoints, recoge la información necesaria de las mismas y interactúa con el servicio correspondiente dependiendo de la petición. También le da formato a la información que recibe del servicio para convertirla en una respuesta HTTP válida en formato JSON.

Generalmente, cada una de las entidades del sistema tiene 5 endpoints distintos, salvo ciertas entidades que no necesitan todos ellos y salvo el caso particular de los presupuestos, que tienen algún endpoint más. Aun así, podemos resumir el funcionamiento de los endpoints de cada una de las entidades del sistema de la siguiente forma:

- **GET elements:** Obtener todos los elementos del sistema. En el caso de materiales, servicios, viajes y tareas, obtenemos todos los que pertenecen a un presupuesto dado.
- **GET element:** Obtiene un elemento en concreto dado su identificador.
- **POST create new element:** Crea un nuevo elemento con los datos requeridos.
- **DELETE element:** Elimina un elemento dado su identificador. Si se borra un elemento que tiene otros por debajo, estos elementos también se eliminarán, como sucede en el caso de los presupuestos, que tienen materiales, servicios, viajes y tareas.
- **PUT element:** Edita un elemento mediante su identificador y proporcionando los nuevos datos.

En el caso de los presupuestos, destacamos tres endpoints particulares que otras entidades no tienen:

- **POST duplicate budget:** Recibe el identificador de un presupuesto, y crea una copia en la BBDD.
- **GET pending and approved budgets:** Devuelve todos los presupuestos aprobados y pendientes de aprobación del sistema.
- **PUT update parameter in editable budgets:** Edita el identificador de los parámetros del sistema a los que están asociados los presupuestos editables del sistema.

9.1.3. Carpeta `services`

Contiene los ficheros que proveen de consultas y funciones de tratamiento de los datos al handler correspondiente. Interactúan con la BBDD mediante el ORM. Hay una por cada entidad de dominio (usuarios, artículos, cargos, parámetros, presupuestos, materiales, servicios, viajes y tareas) y una adicional para gestionar la autenticación.

Dentro de cada fichero se encuentran consultas análogas a los endpoints que posee cada una de las entidades. Nos permiten obtener todos los elementos del tipo correspondiente, un elemento concreto utilizando su identificador, crear un elemento, editarlo y eliminarlo. Tenemos dos casos especiales, el de `budget-service` que, como mencionamos en la sección anterior, nos permite duplicar un presupuesto, editar los parámetros de todos los presupuestos editables, y obtener los presupuestos aprobados y pendientes de aprobación, y el caso especial de `auth-service`, que contiene funciones específicas para comprobar las credenciales del usuario.

9.1.4. Carpeta models

En la carpeta model se encuentran definidas las nueve entidades de la BBDD, habiendo una por cada tabla: **users**, **parameters**, **articles**, **positions**, **budgets**, **materials**, **tasks**, **trips** y **services**. Dentro de cada modelo se definen sus atributos, que equivalen a las columnas de la tabla correspondiente a la BBDD.

9.1.5. Carpeta tools

En la carpeta de tools encontramos los siguientes archivos Javascript:

- **to:** Contiene una función que facilita la gestión de promesas Javascript.
- **crypto:** Contiene las funciones de hash que se utilizan para la gestión del encriptado y desencriptado de contraseñas utilizando Passport.js [10].
- **auth-middleware:** Contiene los middleware (funciones que se ejecutan cada vez que se realiza una petición) relacionados con la protección de los endpoints utilizando JWT (JSON Web Token).

9.1.6. Otros ficheros relevantes

En esta sección se indican otros ficheros de interés del sistema, que se encuentran en la carpeta raíz (/) del proyecto.

- **/app.js:** Punto de entrada de la aplicación, donde se instancia el servidor Express [4] y se crean las rutas de la API:
- **/db.js:** Gestiona la conexión y sincronización con la BBDD utilizando Sequelize [14], así como las relaciones entre las distintas tablas.
- **/middleware.js:** Utiliza los middlewares de autenticación para proteger los endpoints del sistema.

9.1.7. Pruebas en Postman

Tal y como se indicó al principio de esta sección, a continuación se indicará como hacer una serie de peticiones que permitan probar el funcionamiento de la API. Para realizar la demostración utilizaremos los endpoints de los cargos del sistema. El resto de peticiones para los demás endpoints se realizarían de forma análoga.

En todas las peticiones a realizar se debe indicar la **URL** y un **endpoint**. En este caso, la URL base sería **http://localhost:8080/** y el endpoint sería **/positions**

Iniciar Sesión

En primer lugar, debemos hacer una petición de login con los credenciales del usuario para obtener el token que necesitamos para hacer peticiones a los endpoints protegidos. Dicho token será incluido en un header con el nombre 'Authorization' en el resto de las peticiones a realizar.

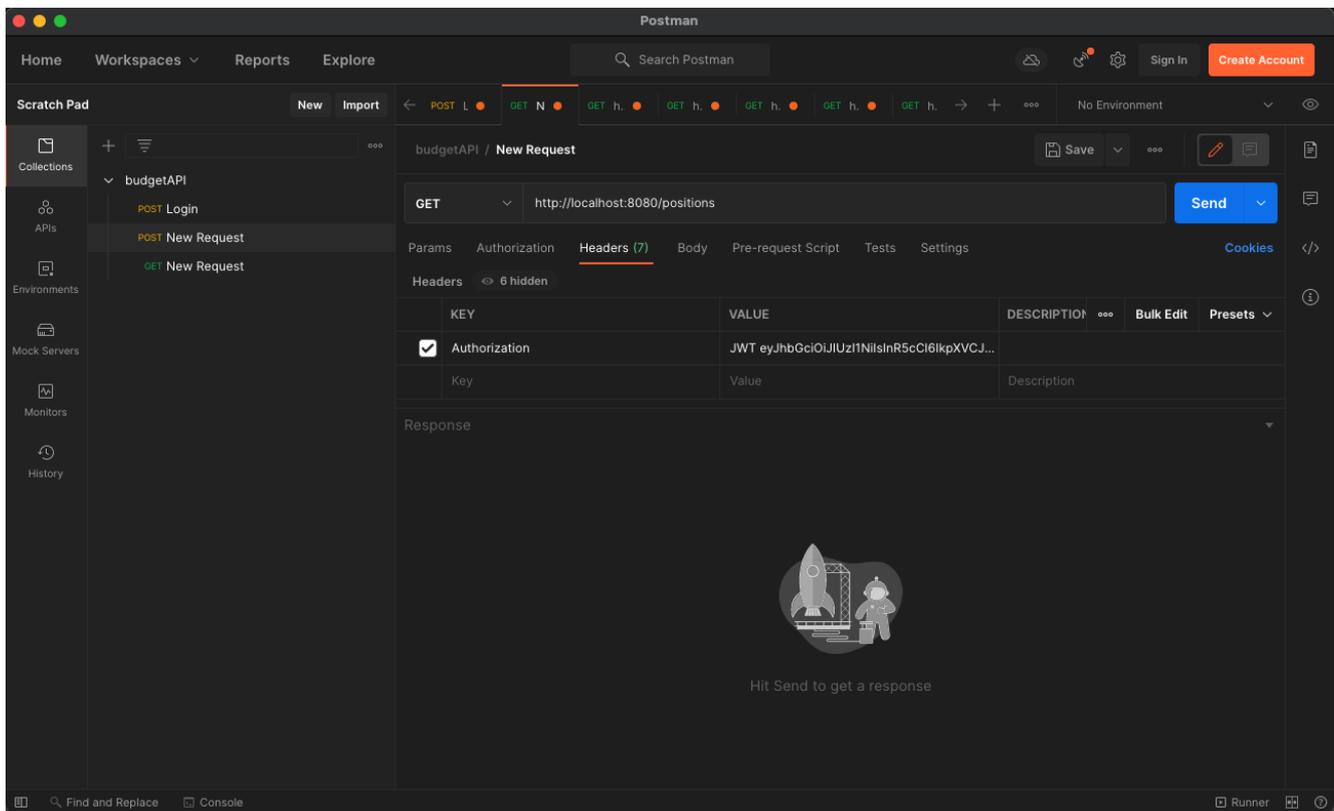


Figura 9.1: Header de autorización de las peticiones

Obtener todos los cargos

Con esta petición obtenemos todos los cargos del sistema.

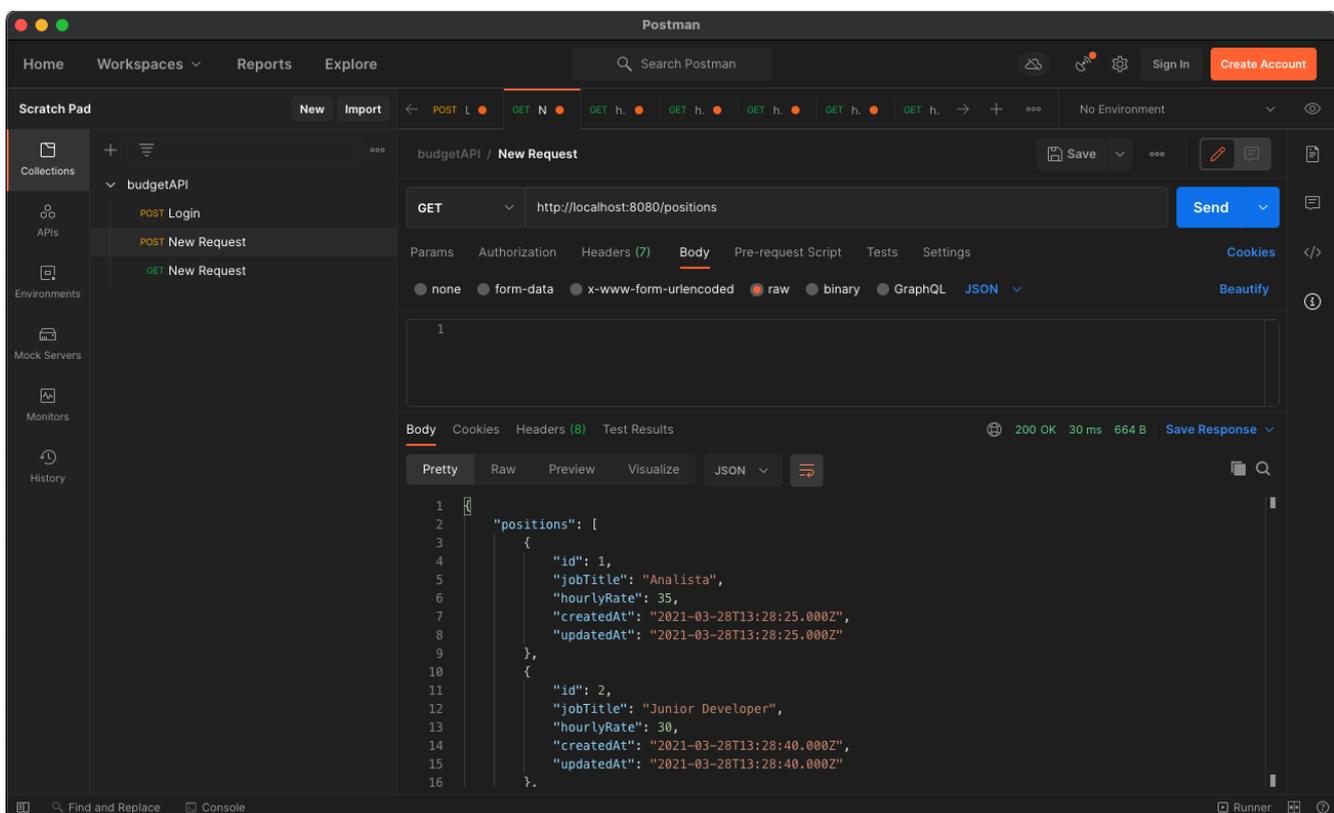


Figura 9.2: GET todos los cargos

Obtener un cargo

Con esta petición obtenemos el cargo con identificador 1 del sistema.

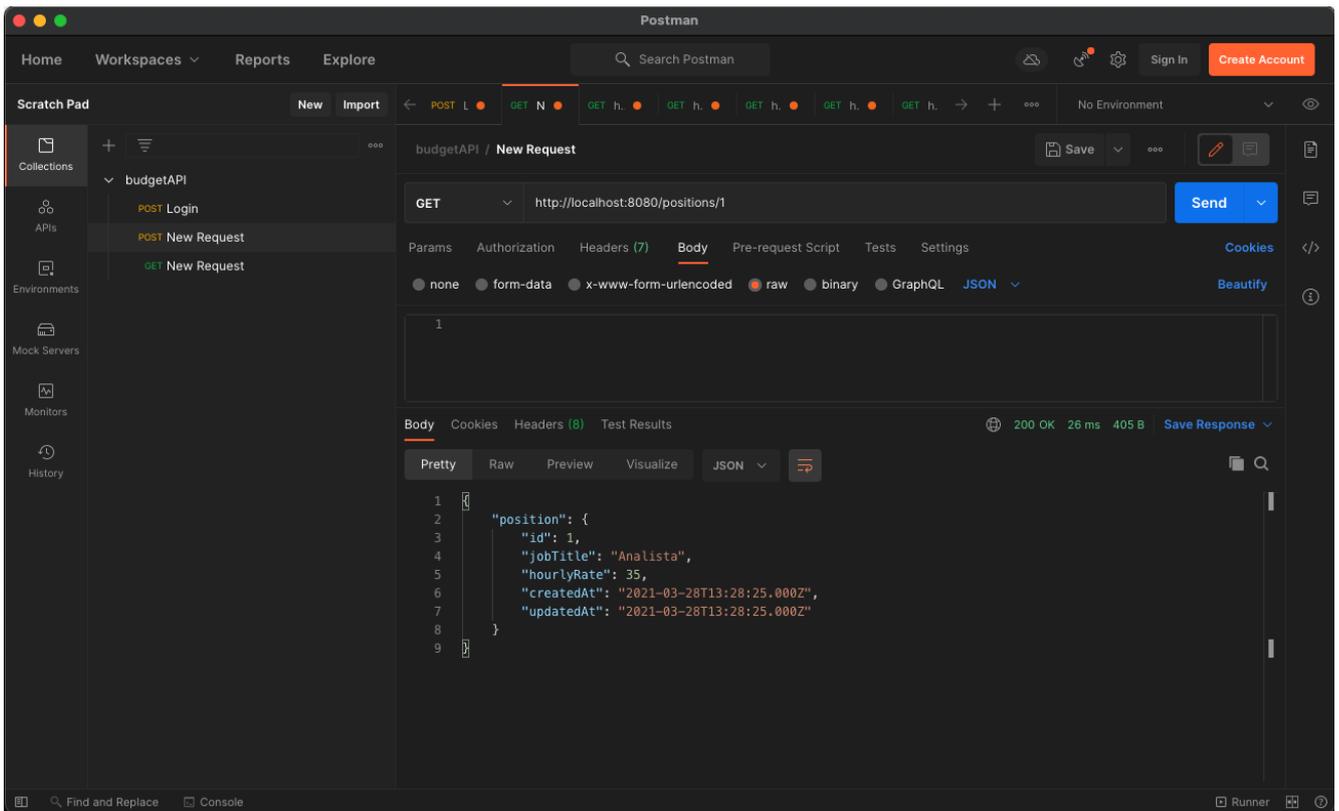


Figura 9.3: GET cargo con identificador 1

Crear un cargo

Con esta petición crearemos un cargo nuevo en el sistema.

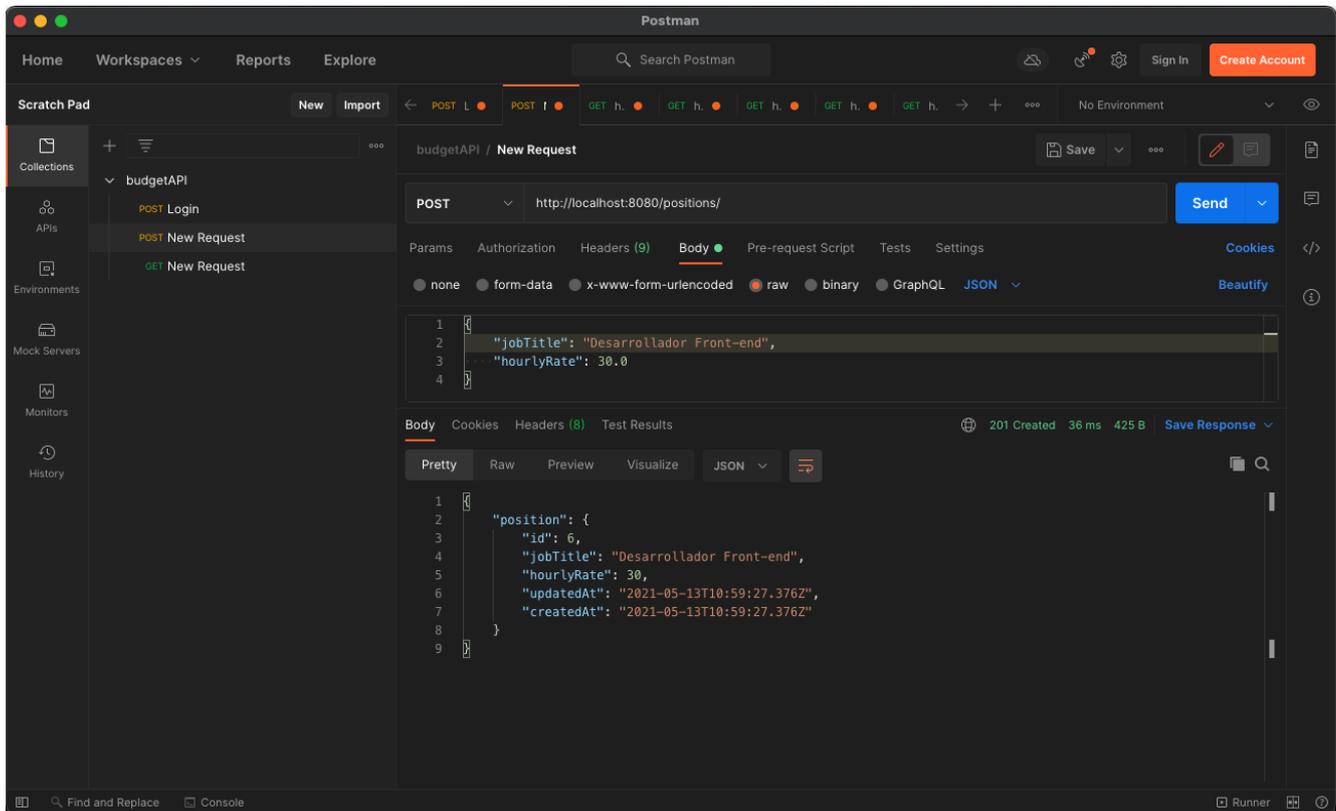


Figura 9.4: POST añadir un cargo

Editar un cargo

Con esta petición editaremos el cargo previamente creado en el sistema.

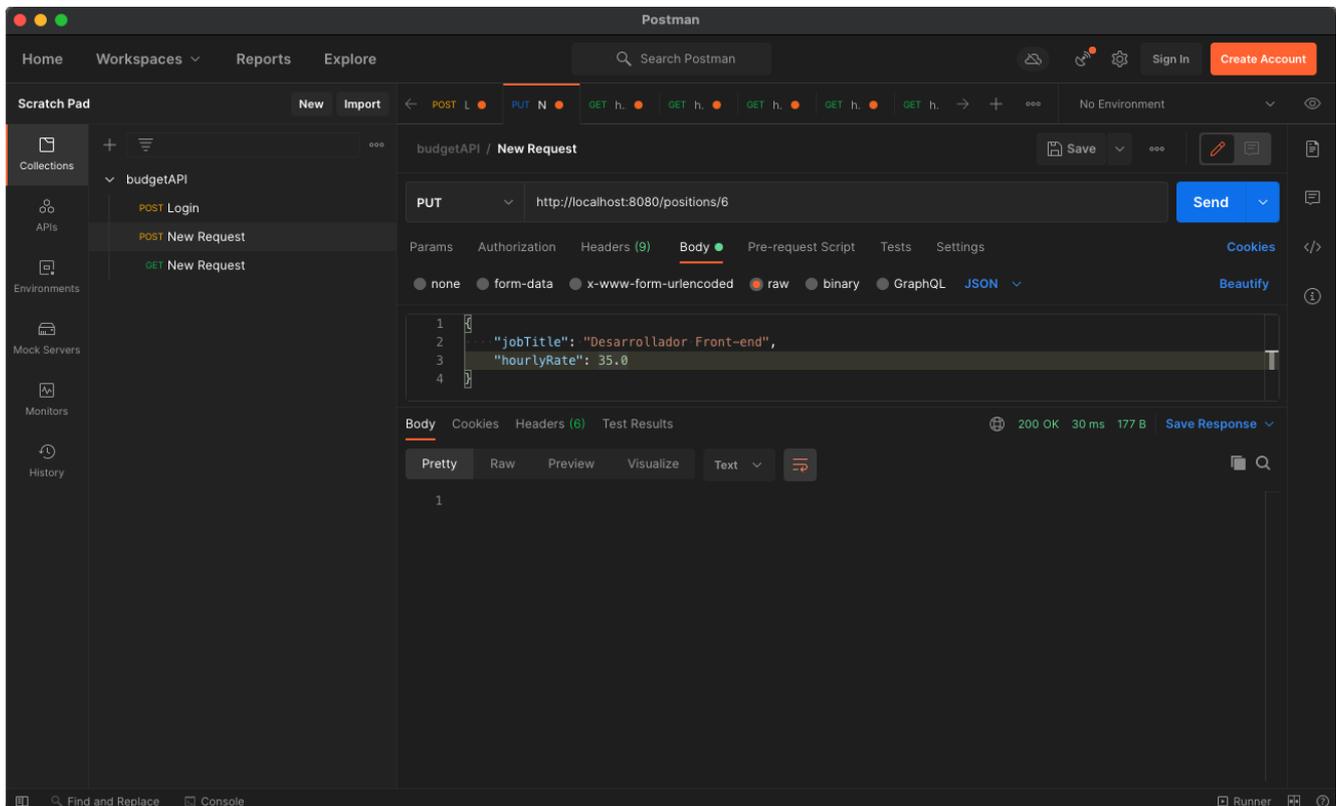


Figura 9.5: PUT editar un cargo

Eliminar un cargo

Con esta petición eliminaremos el cargo previamente creado en el sistema.

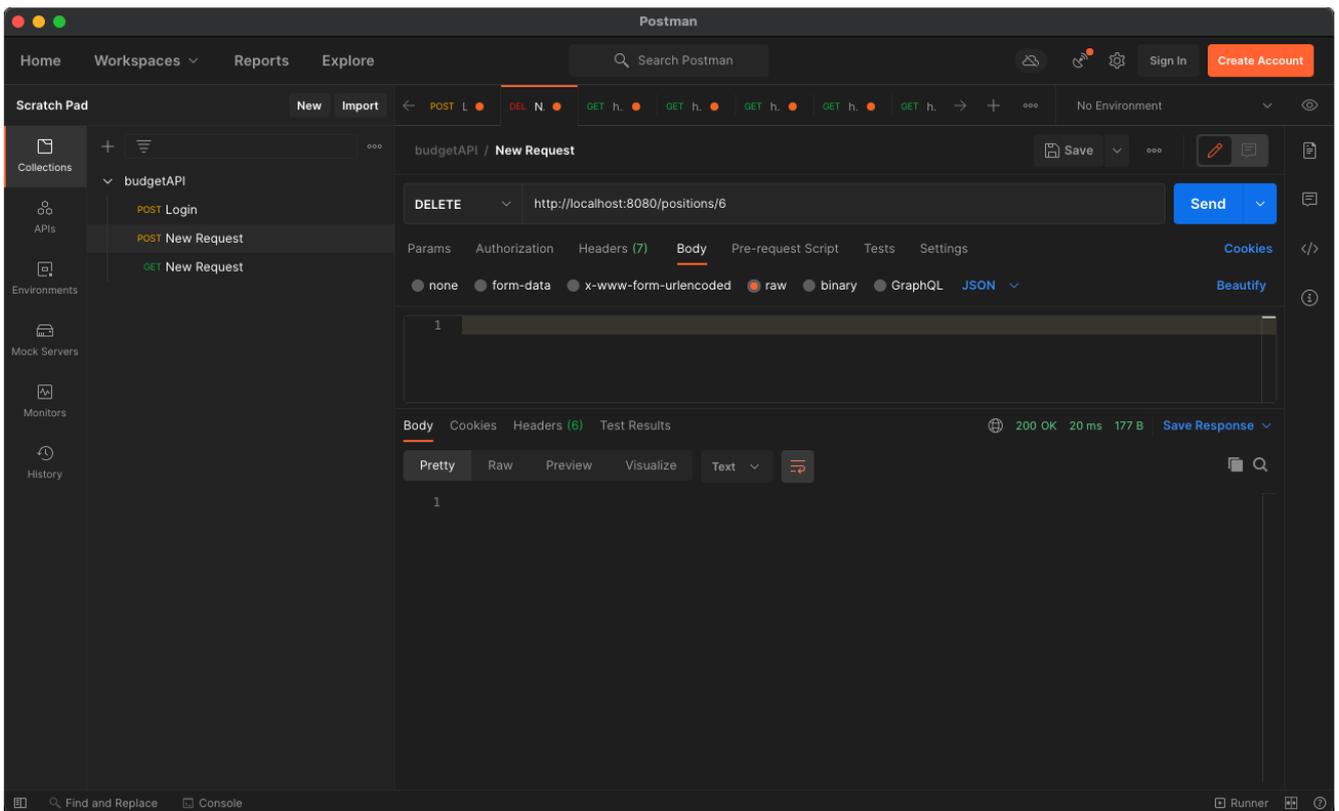


Figura 9.6: DELETE un cargo

9.2. Front-end

En este punto se detallará lo necesario para la comprensión del front-end del sistema implementado.

9.2.1. Directivas generales

En la raíz del proyecto (dentro de la carpeta 'src') encontramos varias carpetas, que a su vez tienen varias subcarpetas, que agrupan los ficheros dependiendo del tipo de funcionalidad que desempeñan en el sistema. Las carpetas en las que está dividido son 'components', 'pages', 'API', 'context' y 'utils'.

- **Carpeta components:** Dentro de esta carpeta se encuentran los componentes React de la aplicación. Están divididos según la vista a la que pertenecen, y a su vez subdivididos dependiendo de la funcionalidad que cumplan dentro de dicha vista.
- **Carpeta pages:** Esta carpeta también contiene componentes, pero únicamente contiene aquellos que se corresponden a cada una de las rutas de la aplicación. Es decir, los componentes que están más arriba en la jerarquía de la misma.
- **Carpeta API:** Contiene todas las llamadas a la API REST del sistema, siendo el único punto de comunicación del front-end con la misma. Dichas llamadas se encuentran divididas en ficheros que se corresponden con cada uno de los endpoints del sistema.
- **Carpeta context:** En esta carpeta se almacenan los dos contextos a los que acceden los componentes del sistema.

- **Carpeta utils:** Contiene utilidades variadas para la gestión y el parseado de los datos.

9.2.2. Carpeta components

La carpeta components contiene prácticamente todos los componentes del sistema. Esta a su vez está subdividida en carpetas que se corresponden a cada una de las vistas del sistema. A continuación se detallará lo que contiene cada una de ellas:

- **Editor:** Contiene todos los componentes relacionados con el editor de presupuestos del sistema. A su vez, está subdividida en los componentes relacionados con cada uno de las entidades que forman un presupuesto: BudgetMaterials, BudgetServices, BudgetTrips y BudgetTasks.
- **PDFView:** Contiene todos los componentes relacionados con el visor de PDF del sistema. Son todos los que se necesitan para la generación de los PDF, y a su vez está subdividida también en las distintas tablas con las que cuenta el archivo: ICTable, SummaryTable, TotalTasksTable, TasksTable, TripsTable, ServicesTable y MaterialsTable.
- **Positions:** Contiene todos los componentes relacionados con la gestión de los cargos del sistema: la tabla y el formulario de creación/edición.
- **Dashboard:** Contiene todos los componentes utilizados para la visualización de la página principal del sistema. Como dicha vista depende del rol del usuario, a su vez se subdivide en carpetas que corresponden al dashboard de cada tipo de usuario:
 - **AdminDashboard:** Página principal del administrador del sistema. Contiene todos los componentes referentes a la gestión de los usuarios del sistema.
 - **BossDashboard:** Página principal de los usuarios con rol de jefe. Contiene todos los componentes relacionados a la visualización y gestión de presupuestos aprobados y pendientes.
 - **CommercialDashboard:** Página principal de los usuarios con rol de comercial. Contiene todos los componentes relacionados a la visualización y gestión de todos los presupuestos del sistema, así como la creación de nuevos.

Además, hay ciertos componentes mas genéricos que no pertenecen a ninguna vista en particular, y son los siguientes:

- **Login:** Componente que permite a los usuarios iniciar sesión en el sistema. Se muestra siempre que no exista un token válido en el localStorage del navegador.
- **Sidebar:** Barra lateral que aparece en todas las vistas del sistema y permite a los usuarios navegar por el mismo.
- **Notification:** Snackbar lateral que aparece para notificar al usuario cuando se ha realizado una acción.
- **Popup:** Cuadro de diálogo que se utiliza para mostrar distintos formularios en todo el sistema.
- **ConfirmationDialog:** Cuadro de diálogo que se utiliza para solicitar confirmación al usuario a la hora de realizar distintos tipos de acciones en todo el sistema.

9.2.3. Carpeta pages

Esta carpeta contiene aquellos componentes que se encuentran en la parte superior de la jerarquía de la aplicación. Es decir, todos aquellos que se corresponden a una vista del sistema. Son los siguientes:

- **Dashboard:** Página principal del sistema. Se muestra al iniciar sesión, y es diferente para cada tipo de usuario. Los **administradores** verán el panel de gestión de usuarios, los **jefes** verán el panel de gestión de presupuestos, donde podrán visualizar presupuestos aprobados o pendientes, así como aprobar o rechazar aquellos que estén pendientes. Finalmente, los **comerciales** verán el panel de control con todos los presupuestos del sistema donde podrán visualizar, editar, eliminar y crear presupuestos.
- **Editor:** Página que se corresponde al editor de presupuestos, al que se accede al editar un presupuesto existente o crear uno nuevo. En él se puede editar el presupuesto, ver un resumen de las cifras del mismo y visualizar las distintas entidades que lo forman (materiales, servicios, viajes y tareas) así como crearlas, editarlas o eliminarlas.
- **Parameters:** En esta vista los usuarios con rol de jefe pueden visualizar y editar los parámetros del sistema que se utilizarán para calcular los valores pertinentes en los presupuestos.
- **Positions:** En esta vista los usuarios con rol de jefe podrán visualizar, editar, eliminar y crear cargos profesionales en el sistema.
- **PDFView:** Vista que permite a los usuarios visualizar un archivo PDF de un presupuesto.

9.2.4. Carpeta API

Contiene todos los ficheros desde donde se realizan peticiones a la API REST utilizando Axios [23]. Existe un fichero para cada endpoint existente en la API, con el fin de dividir las peticiones de una forma práctica y fácil de entender.

9.2.5. Carpeta context

Contiene los dos contextos a los que tienen acceso los componentes del sistema. El concepto de contexto en React fue explicado en el capítulo 6. A fin de simplificar el sistema, se ha decidido tener únicamente dos contextos. El resto de estado se almacena de forma particular en cada componente y se pasará entre componentes utilizando las propiedades. Los dos contextos con los que cuenta el sistema son los siguientes:

- **context:** Contexto general del sistema. Contiene los datos del usuario que tiene la sesión iniciada y gestiona el almacenamiento de dichos datos en el LocalStorage (objeto que nos permite almacenar datos de manera local en un navegador) [22]. Tienen acceso a él todos los componentes del sistema.
- **editorContext:** Contexto asociado al editor de presupuestos del sistema. Contiene todos los datos del presupuesto con el que se está trabajando, así como de las entidades asociadas a él. También contiene distintas funcionalidades que permiten calcular los datos que se muestran en el resumen del presupuesto. Sólo tienen acceso a él los componentes de la vista Editor.

9.2.6. Carpeta utils

Contiene un fichero llamado **budgetUtils** que contiene funciones que ayudan al parseo de datos del presupuesto, como por ejemplo las fechas del mismo.

9.2.7. Otros ficheros de interés

Únicamente cabe destacar el fichero **app.js** que se encuentra en la raíz del sistema (/src) y contiene el punto de entrada de la aplicación, donde se declaran las rutas y se renderizan todas las vistas del sistema.

Capítulo 10

Manual de instalación

En este apartado se detallarán los pasos y dependencias necesarias para hacer funcionar el conjunto del proyecto en cualquier máquina. Se detallarán dos formas de hacerlo: una utilizando Docker [3], que ha sido la tecnología utilizada para centralizar todas las dependencias y así facilitar la portabilidad y el despliegue del sistema, y otra forma para desplegarlo localmente, todo ello dependiendo de las necesidades del desarrollador.

La instalación y despliegue del sistema puede realizarse en cualquier sistema operativo Windows, Linux o MacOS.

Se dividirá la explicación en tres apartados: base de datos, back-end y front-end. Además, a continuación se incluirá una sección que detalle los pasos a seguir para la instalación de Docker.

10.1. Instalación de Docker

10.1.1. Requisitos de software

Antes de comenzar, se deberá contar con el siguiente software:

- Sistema operativo Linux, Mac OS o Windows.

Las herramientas que se deben instalar son dos: Docker y Docker-compose.

10.1.2. Procedimiento

Para instalar ambas herramientas, debemos llevar a cabo los siguientes pasos (pueden variar ligeramente entre sistemas operativos. Los pasos aquí citados han sido probados en un sistema operativo Unix):

1. Entrar en la página de descarga de Docker [3] y seguir los pasos para la instalación del software en el dispositivo. Normalmente, ambas herramientas se instalan al unísono.
2. Comprobar si se ha realizado la instalación abriendo un terminal y escribiendo `docker --version` y `docker-compose --version`. Si estos comandos devuelven información acerca de la versión del sistema, es que la instalación ha resultado exitosa.

10.1.3. Breve introducción a Docker

Para que las instrucciones acerca de la realización del despliegue que tendrán lugar en las secciones posteriores de este capítulo sean más fáciles de entender, cabe explicar varios conceptos importantes de Docker, aunque sea de una forma superficial:

- **Imagen:** Captura del estado de un contenedor, que se asemeja al snapshot de una máquina virtual, pero mucho más ligero.
- **Contenedor:** Instancias en ejecución de una imagen.
- **Dockerfile:** Archivo de configuración que se utiliza para crear imágenes. En él indicamos que queremos que tenga la imagen, y los comandos necesarios para instalar las herramientas de la misma.
- **Docker-compose:** Establece la información asociada al contenedor para su ejecución.

Si se requiere más información acerca de estos conceptos, se puede consultar en la documentación oficial de Docker [19].

10.2. Base de datos

Se ha utilizado para el sistema una base de datos MySQL. Por tanto, a continuación se explicará como crear y poblar con los datos necesarios para el funcionamiento inicial del sistema (un usuario con el rol de administrador) una base de datos de tales características.

10.2.1. Instalación de MySQL

Para comenzar con la instalación de MySQL, se deberá contar con el siguiente software:

- Sistema operativo Linux, MacOS o Windows.

Para realizar la instalación, hay que abrir un terminal y ejecutar los siguientes comandos (Pueden variar dependiendo del sistema operativo. Los comandos mencionados a continuación han sido probados en un sistema Linux):

```

1   sudo apt update
2   sudo apt install mysql-server
3   mysql_secure_installation

```

Para arrancar el servidor, bastará con ejecutar *mysql.server start* y para pararlo, *mysql.server stop*.

10.2.2. Base de datos local

A continuación se explicará como crear y poblar una base de datos con lo necesario para empezar a utilizar el sistema.

1. Conectarse al servidor utilizando el comando:

```

1   mysql -u root -p

```

2. Crear un usuario en MySQL con un nombre de usuario y una contraseña. Dichos credenciales son los que se utilizarán posteriormente para acceder a la BBDD desde el back-end. Para ello, ejecutar el siguiente comando, sustituyendo *username* por el usuario y *password* por la contraseña:

```

1   CREATE USER 'username'@'localhost' IDENTIFIED BY 'password';

```

3. Crear una base de datos en MySQL. El nombre que se le otorgue a la base de datos en este paso es el que se utilizará posteriormente para la conexión con el back-end. Para crearla, ejecutar el siguiente comando, sustituyendo *databaseName* por el nombre de la base de datos:

```
1 CREATE DATABASE dbName;
```

4. Otorgar permisos al usuario en la base de datos que se acaba de crear:

```
1 GRANT ALL PRIVILEGES ON dbName . * TO
2 'username'@'localhost';
3 FLUSH PRIVILEGES;
```

5. Poblar la base de datos con el script de creación:

```
1 mysql -u root -p dbName < starterSQL.sql
```

10.2.3. Despliegue con Docker

Para crear un contenedor con nuestra base de datos, no necesitaremos compilar una imagen propia, ya que podemos utilizar una imagen de MySQL. Todo lo necesario para la creación del contenedor se encuentra en el Docker-compose del proyecto. El contenedor asociado a la BBDD tendrá el nombre **'db'**. Los credenciales (username, password, name, root password...) se encuentran en el apartado environments del contenedor en el Docker-compose. Dichos credenciales son los que se utilizarán posteriormente para conectarla con el back-end.

Para levantar el contenedor, desde el mismo directorio donde se encuentra el fichero docker-compose utilizaremos el siguiente comando:

```
1 docker-compose up -d
```

El flag **'-d'** lo que hace es levantar el contenedor en modo **'detached'**, es decir, en segundo plano.

Para poblar la base de datos con el script inicial (situado en la carpeta db-utils del backend), habrá que ejecutar el siguiente comando (sustituyendo dbName por el nombre de la base de datos):

```
1 docker exec -i db sh -c 'exec mysql -uroot
2 -p"$MYSQL_ROOT_PASSWORD" dbName < ./starterSQL.sql'
```

10.2.4. Conexión con Back-end

A continuación se explicará brevemente como adaptar la conexión de la base de datos con el back-end, dependiendo de si se está utilizando el despliegue en local o utilizando Docker. El código a modificar es el siguiente, localizado en el fichero **'db.js'**, que se encuentra en la raíz del proyecto:

```
1 const sequelize = new Sequelize('dbName', 'username', 'password', {
2   host: 'db',
3   dialect: 'mysql'
4 });
```

El fragmento de código aquí mostrado es la conexión que Sequelize, el ORM, realiza con la BBDD. Para hacerlo funcionar debemos sustituir los siguientes parámetros:

- dbName: Nombre de la base de datos. Sustituirlo por el nombre que hayamos elegido a la hora de su creación.
- username y password: Usuario y contraseña del usuario que hemos creado previamente para la gestión de la base de datos.

- host: Localización de la base de datos. El host debería ser **localhost** si se hace el despliegue en local y **db** si se hace el despliegue utilizando Docker (y no se ha cambiado el tag image del docker-compose proporcionado).

10.3. Back-end

10.3.1. Software necesario

- Node v15.12 o superior

10.3.2. Despliegue local

1. Clonar el repositorio situado en *budgets/budget-backend*.
2. Situarse en el directorio raíz y ejecutar:

```
1 npm install
2 node app.js
```

10.3.3. Despliegue con Docker

1. Clonar el repositorio situado en *budgets/budget-backend*.
2. Situarse en la raíz del proyecto y crear la imagen para el contenedor Docker con el siguiente comando:

```
1 docker build -t budget_api .
```

3. Situarse en la misma ruta que el Dockerfile del proyecto y levantar el contenedor con el comando:

```
1 docker-compose up -d
```

10.4. Front-end

10.4.1. Software necesario

- Node v15.12 o superior

10.4.2. Despliegue local

1. Clonar el repositorio situado en *budgets/budget-frontend*.
2. Situarse en el directorio raíz y ejecutar:

```
1 npm install
2 npm start
```

10.4.3. Despliegue con Docker

1. Clonar el repositorio situado en *budgets/budget-frontend*.
2. Situarse en la raíz del proyecto y crear la imagen para el contenedor Docker con el siguiente comando:

```
1 docker build -t budget_frontend .
```

3. Situarse en la misma ruta que el Dockerfile del proyecto y levantar el contenedor con el comando:

```
1 docker-compose up -d
```

NOTA: Los tres contenedores se encuentran definidos en el mismo Docker-compose. Por tanto, al ejecutar el comando up, se levantarán los tres a la vez.

Capítulo 11

Manual de usuario

En este capítulo se detallará paso por paso y utilizando apoyo visual la forma de utilizar la interfaz del sistema desarrollado durante el transcurso del proyecto. La explicación de dichas acciones se dividirá basándose en el rol que tenga en el sistema el usuario que pueda realizarlas.

11.1. Acceso a la aplicación

Este paso es común para todos los usuarios del sistema. La aplicación será accesible mediante un navegador web a través de un dominio dado. Para poder acceder a las funcionalidades del sistema, el usuario deberá haber sido dado de alta en el sistema previamente por el administrador del mismo, y deberá iniciar sesión.

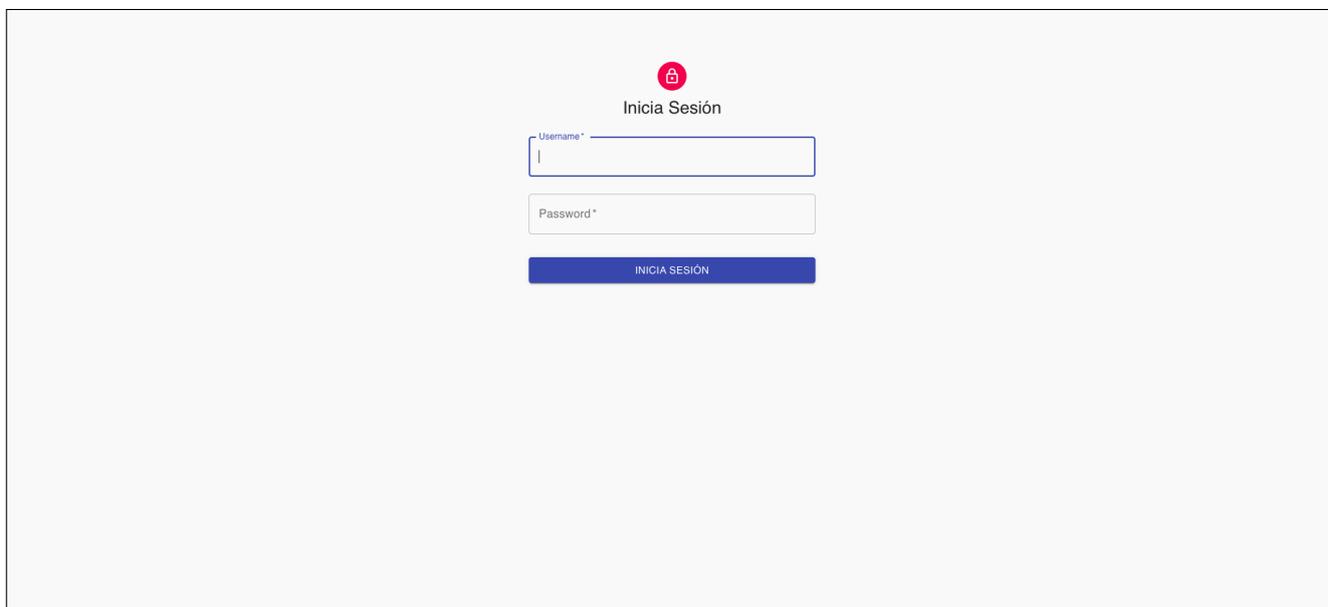
The image shows a login interface titled "Inicia Sesión". At the top center is a red circular icon with a white padlock. Below the title are two input fields: "Username*" and "Password*", both with asterisks indicating they are required. The "Username*" field contains the character "1". Below the input fields is a solid blue button with the text "INICIA SESIÓN" in white capital letters.

Figura 11.1: Inicio de sesión

11.2. Administrador del Sistema

El administrador del sistema visualizará un panel que le permitirá realizar las siguientes acciones:

- **Visualizar listado de usuarios:** Visualizar una tabla con todos los usuarios del sistema, su nombre, correo electrónico, nombre de usuario y rol. (Figura 11.2)

- **Eliminar un usuario:** Para dar de baja a un usuario en el sistema, el administrador deberá pulsar el botón de la papelera de la tabla de usuarios que se corresponda al usuario que desea eliminar. Tras pulsarlo, aparecerá un cuadro de diálogo solicitando confirmación para realizar la acción. (Figura 11.3)
- **Crear un usuario:** Para crear un usuario, el administrador deberá rellenar el formulario de la parte superior de la pantalla y pulsar el botón de 'REGISTRAR'. (Figura 11.2)

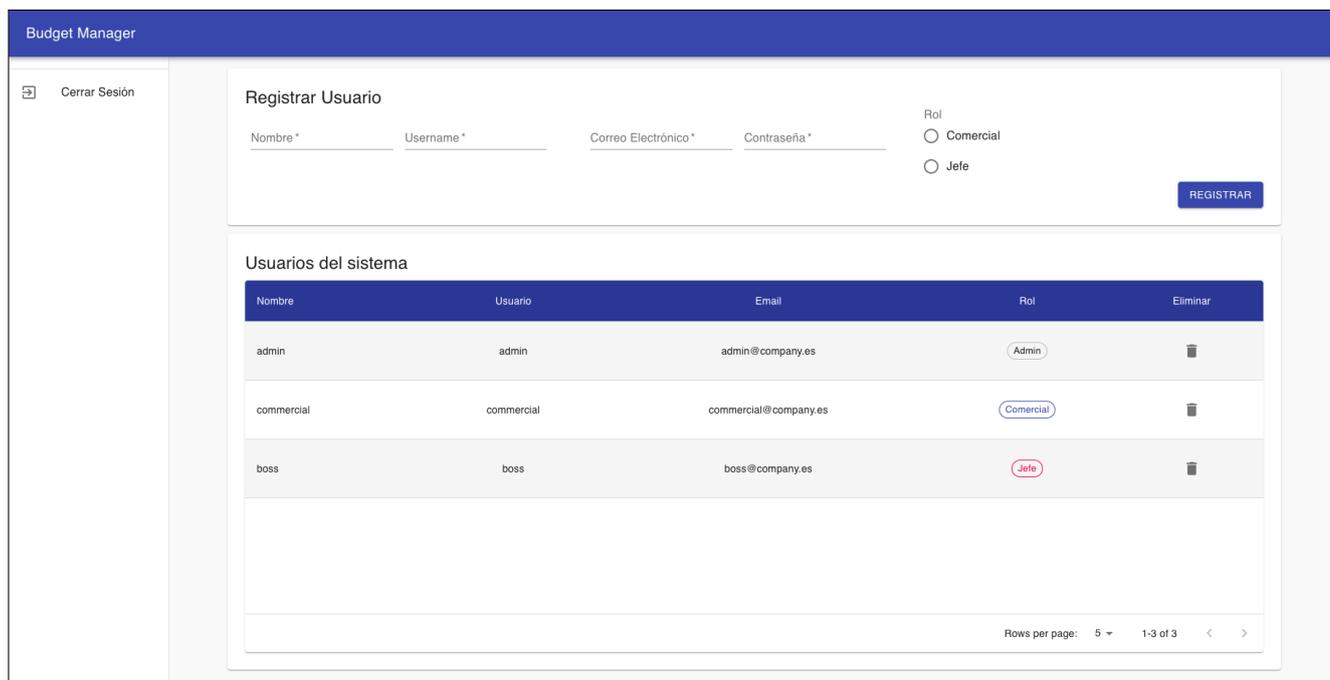


Figura 11.2: Panel de Administración de Usuarios

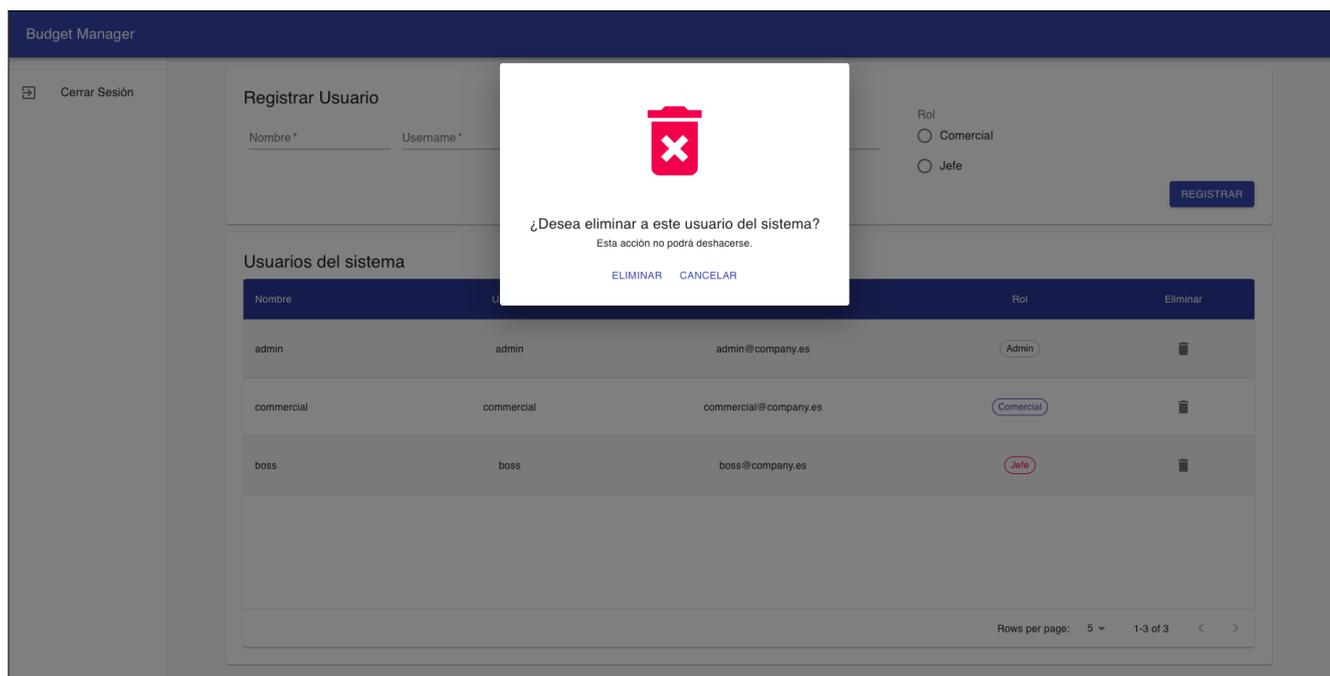


Figura 11.3: Dialogo de confirmación - Eliminar usuario

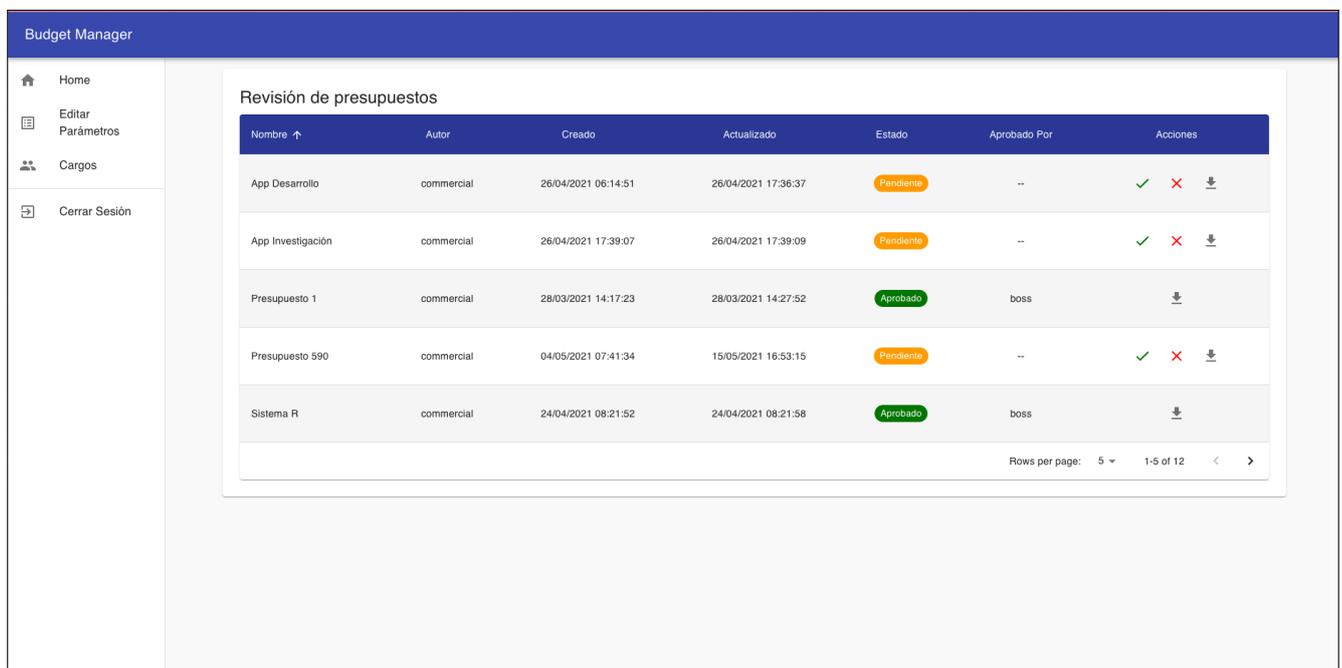
11.3. Jefes

Los usuarios del sistema con rol de jefe podrán realizar las siguientes acciones: aprobar y rechazar presupuestos, visualizar presupuestos en formato PDF y gestionar los cargos y los parámetros del sistema. A continuación se explicará con mayor detalle en que parte del sistema se llevan a cabo dichas acciones.

11.3.1. Revisión de Presupuestos

Al iniciar sesión con una cuenta de usuario con rol de jefe, se podrá visualizar un panel de control que mostrará todos los presupuestos del sistema con estado aprobado y pendiente de aprobación (figura 11.4). Sobre ellos se podrán llevar a cabo las siguientes acciones:

- **Aprobar un presupuesto:** Sólo se podrá llevar a cabo sobre los presupuestos pendientes de aprobación. Para aprobar un presupuesto el usuario deberá pulsar el botón con un check verde situado en la fila de la tabla correspondiente al presupuesto que desee aprobar. Tras pulsar, el estado del presupuesto cambiará a aprobado.
- **Rechazar un presupuesto:** Sólo se podrá llevar a cabo sobre los presupuestos pendientes de aprobación. Para rechazar un presupuesto el usuario deberá pulsar el botón con una cruz roja situado en la fila de la tabla correspondiente al presupuesto que desee rechazar. Tras pulsar, el estado del presupuesto cambiará a borrador. Por tanto, desaparecerá de la tabla y podrá volver a ser editado por el comercial.
- **Visualizar un archivo PDF de un presupuesto:** Para visualizar un presupuesto, bastará con pulsar en el botón con el icono de descarga situado en la fila de la tabla correspondiente al presupuesto que desee visualizar. Tras pulsar, se abrirá un visualizador de PDF como el de la figura 11.5 que permitirá al usuario ver el presupuesto, descargarlo, imprimirlo...



The screenshot shows a web application interface titled 'Budget Manager'. On the left is a navigation sidebar with options: Home, Editar Parámetros, Cargos, and Cerrar Sesión. The main content area is titled 'Revisión de presupuestos' and contains a table with the following data:

Nombre ↑	Autor	Creado	Actualizado	Estado	Aprobado Por	Acciones
App Desarrollo	commercial	26/04/2021 06:14:51	26/04/2021 17:36:37	Pendiente	--	✓ ✗ 📄
App Investigación	commercial	26/04/2021 17:39:07	26/04/2021 17:39:09	Pendiente	--	✓ ✗ 📄
Presupuesto 1	commercial	28/03/2021 14:17:23	28/03/2021 14:27:52	Aprobado	boss	📄
Presupuesto 590	commercial	04/05/2021 07:41:34	15/05/2021 16:53:15	Pendiente	--	✓ ✗ 📄
Sistema R	commercial	24/04/2021 08:21:52	24/04/2021 08:21:58	Aprobado	boss	📄

At the bottom right of the table, there is a pagination control: 'Rows per page: 5' and '1-5 of 12'.

Figura 11.4: Panel de Revisión de Presupuestos

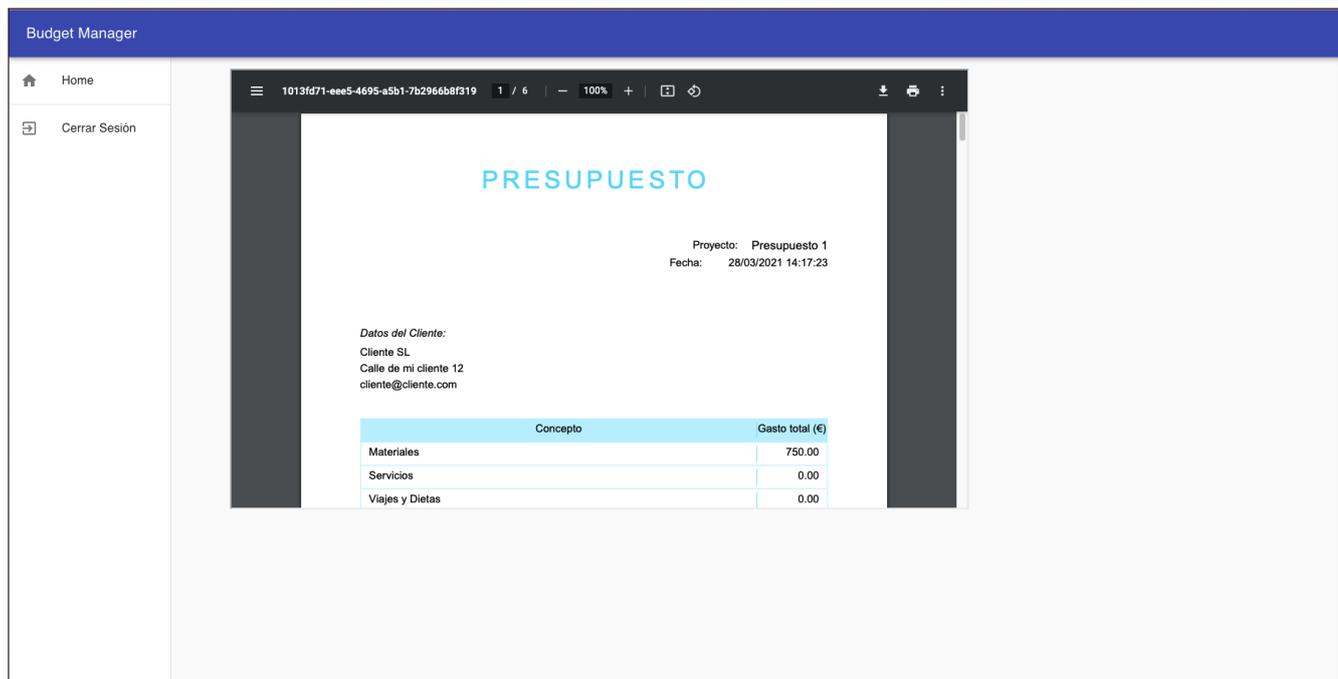


Figura 11.5: Visualizador de archivos PDF

11.3.2. Gestión de Cargos

Para acceder a la vista de gestión de cargos del sistema, será necesario pulsar el botón 'CARGOS' situado en la barra lateral. Desde esa vista, el usuario podrá realizar las siguientes acciones:

- **Visualizar cargos del sistema:** Al acceder a la vista de cargos, el usuario podrá visualizar una tabla con todos los cargos del sistema, incluyendo el nombre del cargo y su salario por hora (figura 11.6).
- **Crear un cargo:** Para añadir un cargo al sistema, el usuario deberá pulsar el botón de 'AÑADIR' de la parte superior de la pantalla, rellenar el formulario del diálogo (figura 11.7) con el nombre del cargo y su salario y nuevamente pulsar 'AÑADIR'. Se creará el cargo y aparecerá en la tabla.
- **Editar un cargo:** Para editar un cargo, el usuario deberá pulsar el botón del lápiz situado en la fila de la tabla correspondiente al cargo que desee editar. Nuevamente, aparecerá un cuadro de diálogo con un formulario con los datos del cargo (figura 11.8). Cuando el usuario haya acabado de editar los datos, deberá pulsar 'GUARDAR'. El cargo se mostrará actualizado en la tabla.
- **Eliminar un cargo:** Para eliminar un cargo, el usuario deberá pulsar el botón del cubo de la basura situado en la fila de la tabla correspondiente al cargo que desee eliminar. Aparecerá un diálogo de confirmación similar al visto previamente para eliminar un usuario (figura 11.3). Al pulsar 'ELIMINAR', el cargo ya no aparecerá en la tabla.

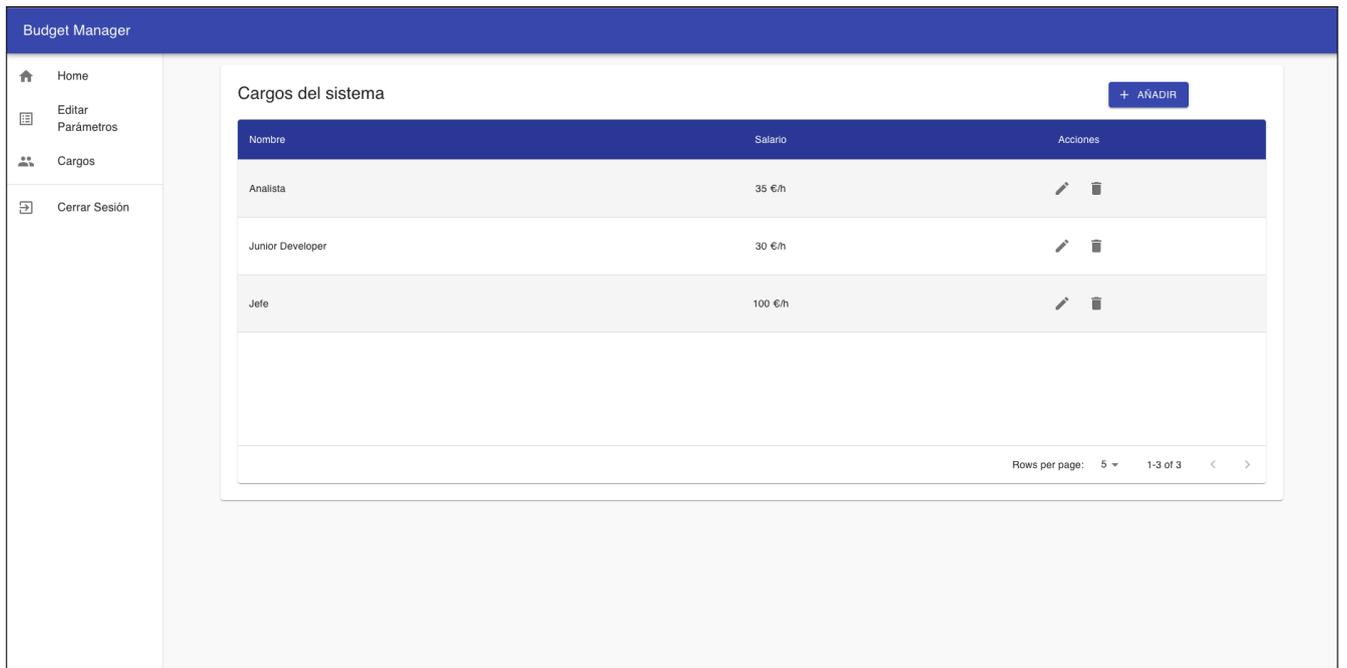


Figura 11.6: Panel de Gestión de Cargos

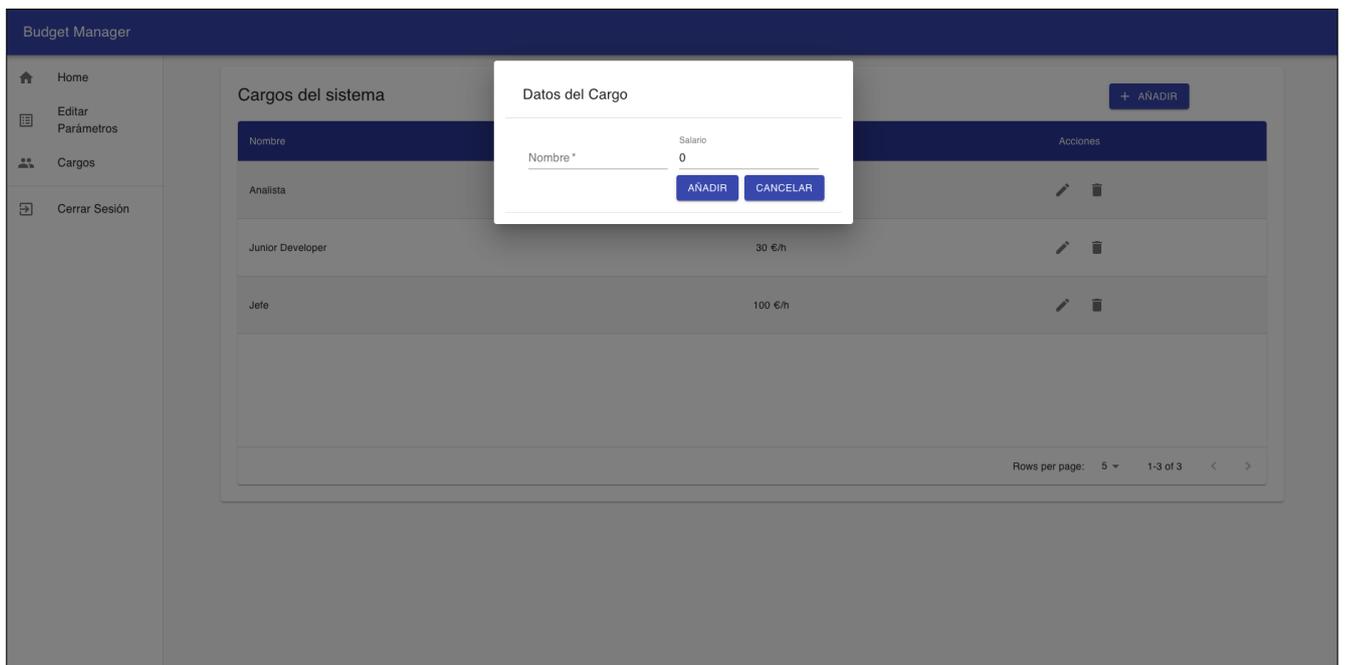


Figura 11.7: Crear Cargo

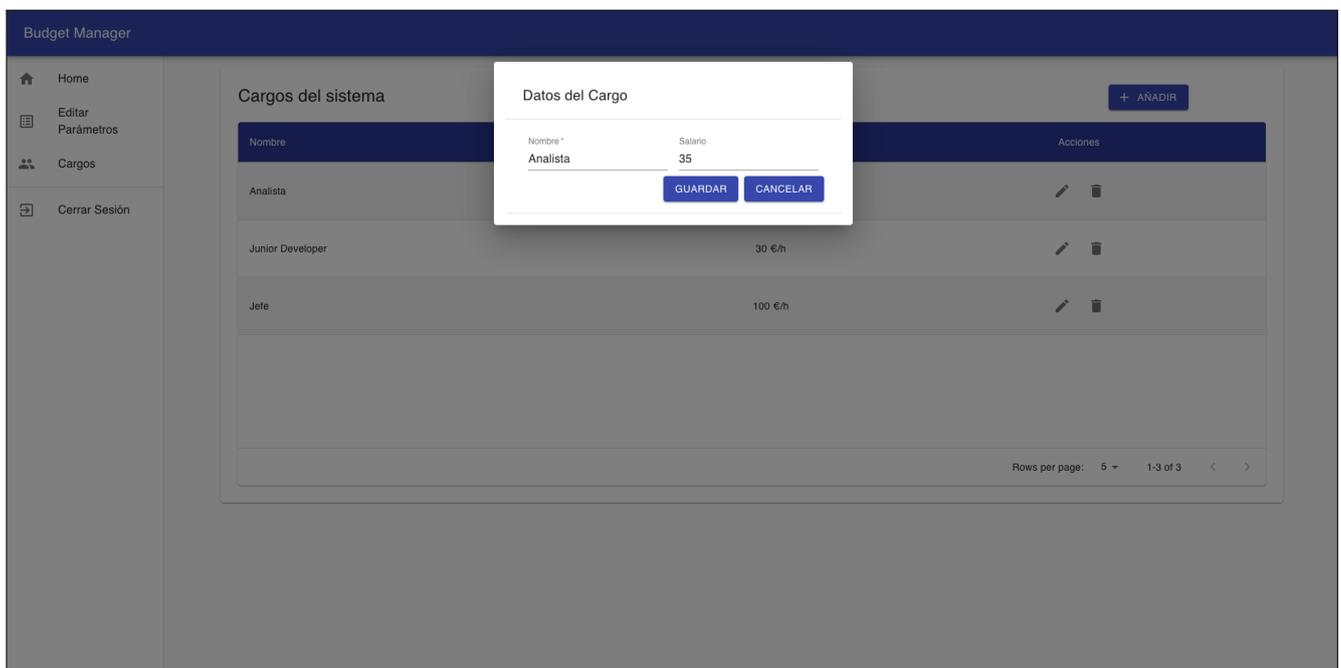


Figura 11.8: Editar Cargo

11.3.3. Parámetros del Sistema

Para acceder a la vista de gestión de parámetros del sistema, será necesario pulsar el botón 'EDITAR PARÁMETROS' situado en la barra lateral. Desde esa pantalla (figura 11.9), el usuario podrá visualizar los parámetros que tiene actualmente el sistema y editarlos cambiando los valores de los campos de texto y pulsando en 'ACTUALIZAR'.

Al actualizar los parámetros del sistema, todos los presupuestos en estado 'borrador' del sistema se recalcularán utilizando los nuevos valores proporcionados por el usuario, mientras que aquellos que estén 'pendientes de aprobación' o 'aprobados' utilizarán los parámetros que tuviesen en el momento de su aprobación.

Costes Indirectos de los Proyectos	
CI viajes	0,05
CI materiales y servicios	0,1
CI M/O	0,6

Dietas	
Dieta completa	35
Media dieta	17,5

Costes Asociados a Viajes	
Costes hotel	60
Precio del Km	0,22

Figura 11.9: Editar Parámetros

11.4. Comerciales

Los usuarios del sistema con rol de comerciales podrán realizar las siguientes acciones: visualizar todos los presupuestos del sistema, crear un presupuesto, editarlo, eliminarlo, duplicarlo y visualizar un archivo PDF del mismo.

11.4.1. Administración de Presupuestos

Al iniciar sesión con una cuenta de usuario con rol de jefe, se podrá visualizar un panel de control (figura 11.10) que mostrará todos los presupuestos del sistema: su nombre, su creador, su estado, su fecha de creación, su última fecha de actualización y, en el caso de los presupuestos aprobados, el usuario que lo aprobó. En dicho panel se podrán realizar las siguientes acciones:

- **Crear un presupuesto:** Para crear un nuevo presupuesto, el usuario deberá introducir un nombre para el mismo en el formulario de la parte superior de la pantalla y pulsar en 'CREAR'. Esto le llevará al editor de presupuestos con un presupuesto "vacío" sobre el que trabajar. Mas adelante se detallarán todas las acciones que se pueden llevar a cabo en dicha pantalla para construir los presupuestos.

- **Eliminar un presupuesto:** Esta acción sólo se puede llevar a cabo sobre los presupuestos que aún están en estado de borrador. Para eliminar un presupuesto, el usuario deberá pulsar el botón del cubo de la basura situado en la fila de la tabla correspondiente al cargo que desee eliminar. Aparecerá un diálogo de confirmación similar al visto previamente para eliminar un usuario (figura 11.3). Al pulsar 'ELIMINAR', el presupuesto ya no aparecerá en la tabla.
- **Visualizar un archivo PDF de un presupuesto:** Se realizará de la misma forma que se explicó previamente para la visualización de archivos en la pantalla de gestión de presupuestos. Esta acción sólo se puede llevar a cabo sobre los presupuestos en estado aprobado o pendiente de aprobación.
- **Duplicar un presupuesto:** Para duplicar un presupuesto, el usuario deberá pulsar el botón con el icono de copia situado en la fila de la tabla correspondiente al presupuesto que desee duplicar. Al pulsarlo, se creará un nuevo presupuesto con el mismo contenido que el seleccionado, pero en su nombre habrá añadido un '(copia)' al final, para que puedan ser distinguidos.
- **Editar un presupuesto:** Esta acción sólo se puede llevar a cabo sobre los presupuestos que aún están en estado de borrador. Para editar un presupuesto, el usuario deberá pulsar el botón del lápiz situado en la fila de la tabla correspondiente al presupuesto que desee editar. Al pulsarlo, el usuario será llevado al editor de presupuestos con el presupuesto seleccionado para que pueda trabajar sobre él. A continuación se explicará con detalle todas las acciones que se pueden llevar a cabo en dicha pantalla.

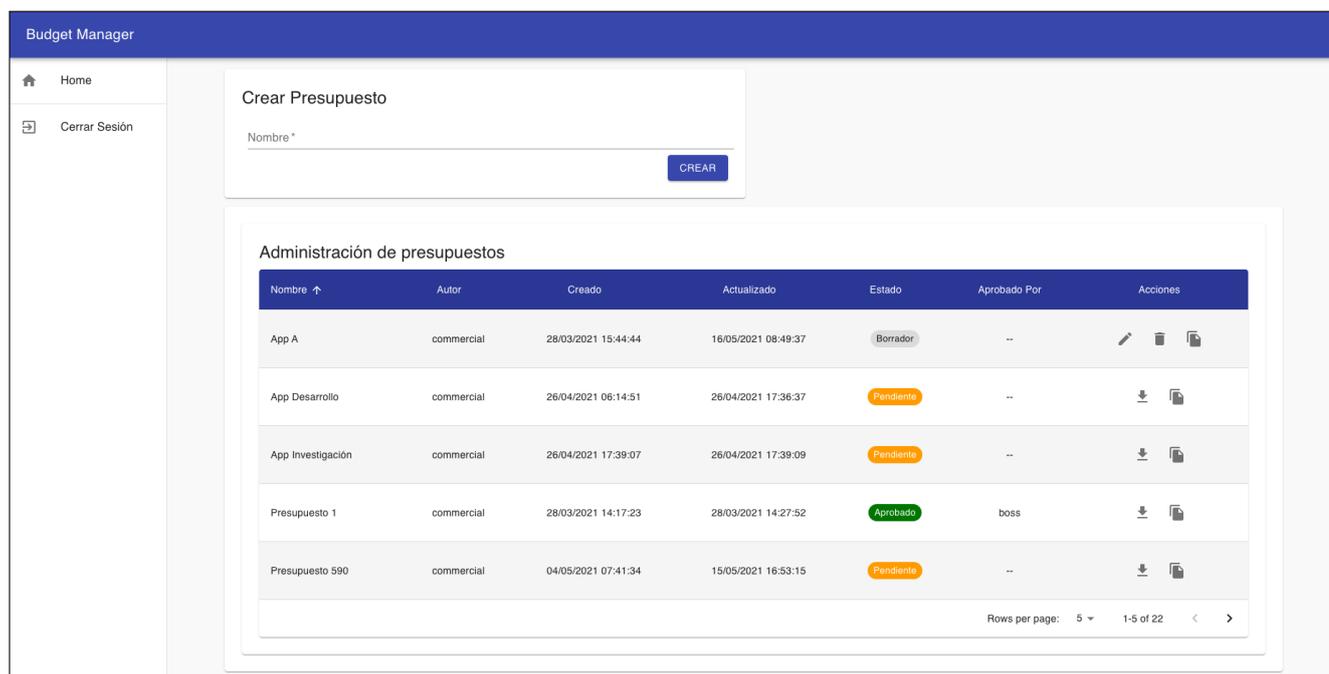


Figura 11.10: Panel de Administración de Presupuestos

11.4.2. Editor de Presupuestos

La vista de edición de presupuestos permite al usuario editar los datos de un presupuesto y añadirle materiales, servicios, tareas o viajes. Como se ha explicado previamente, se puede acceder a ella al crear un presupuesto o al editar uno.

En primer lugar, se muestra un formulario (figura 11.11) con los datos principales del presupuesto. Desde este formulario se pueden editar dichos datos y para modificarlos, el usuario tiene dos opciones:

- **SALIR SIN GUARDAR:** Al pulsar el botón, el usuario volverá a la pantalla de administración.
- **GUARDAR Y SALIR:** Al pulsar el botón, el usuario actualizará los datos del presupuesto y volverá a la pantalla de administración. El estado del presupuesto seguirá siendo el de 'borrador'.
- **GUARDAR Y FINALIZAR:** Al pulsar el botón, el usuario actualizará los datos del presupuesto y volverá a la pantalla de administración. El estado del presupuesto pasará a ser 'pendiente de aprobación', por lo que el usuario no podrá continuar editándolo.

The screenshot shows a web application interface titled 'Budget Manager'. The main content area displays a form for 'Proyecto Blockchain' with the following details:

- Nombre:** Proyecto Blockchain
- Oferta al cliente:** 26500
- Nombre del Cliente:** Empresa Z
- Dirección del Cliente:** Empresa Z
- CIF del Cliente:** 1234567F
- Información de Contacto:** ezenterprise@email.com

Three buttons are visible on the right side of the form: 'SALIR SIN GUARDAR', 'GUARDAR Y SALIR', and 'GUARDAR Y FINALIZAR'.

Below the form, there are two summary tables:

Concepto	Gasto total (€)
Materiales	30.00
Servicios	60.00
Viajes y Dietas	192.20
Total	282.20

Mano de Obra	Precio Hora (€)	Horas	Total (€)
Analista	35	8	280
Total Mano de Obra			280
Total Costes Directos			562.2

Figura 11.11: Formulario del Presupuesto

Debajo del formulario ya descrito, el usuario dispondrá de una serie de datos que le permitirán tener un vistazo general del presupuesto. Estos datos serán los siguientes:

- Gasto total en materiales, servicios, viajes, y dietas. Se mostrará además el total de estas tres cifras y la diferencia entre dicho total y la oferta que se le ha realizado al cliente, lo que se conoce como la **FACTURACIÓN NETA**.
- Gasto total en mano de obra, incluyendo un resumen sobre la cantidad de horas que se necesitará de cada cargo que esté involucrado en el presupuesto, así como el precio total de estas horas. Además, se mostrará la suma de esta cifra y el total en materiales, servicios y viajes, lo que se denomina **TOTAL COSTES DIRECTOS**.
- **MARGEN TEÓRICO** del proyecto. Es decir, la diferencia entre la oferta al cliente y el total de costes directos del presupuesto.
- Recuperación de costes indirectos del proyecto. Se utilizan los porcentajes definidos en los parámetros del sistema para realizar el cálculo de dichos costes para materiales y servicios, viajes y mano de obra. El total es el denominado **TOTAL COSTES INDIRECTOS**.
- Oferta al cliente - (dinero que se espera ganar)
- **COSTE TOTAL:** Suma de los costes directos e indirectos del proyecto.
- **BENEFICIO ANTES DE IMPUESTOS:** Diferencia entre el coste total y la oferta al cliente.

- Porcentaje de ganancias del proyecto.

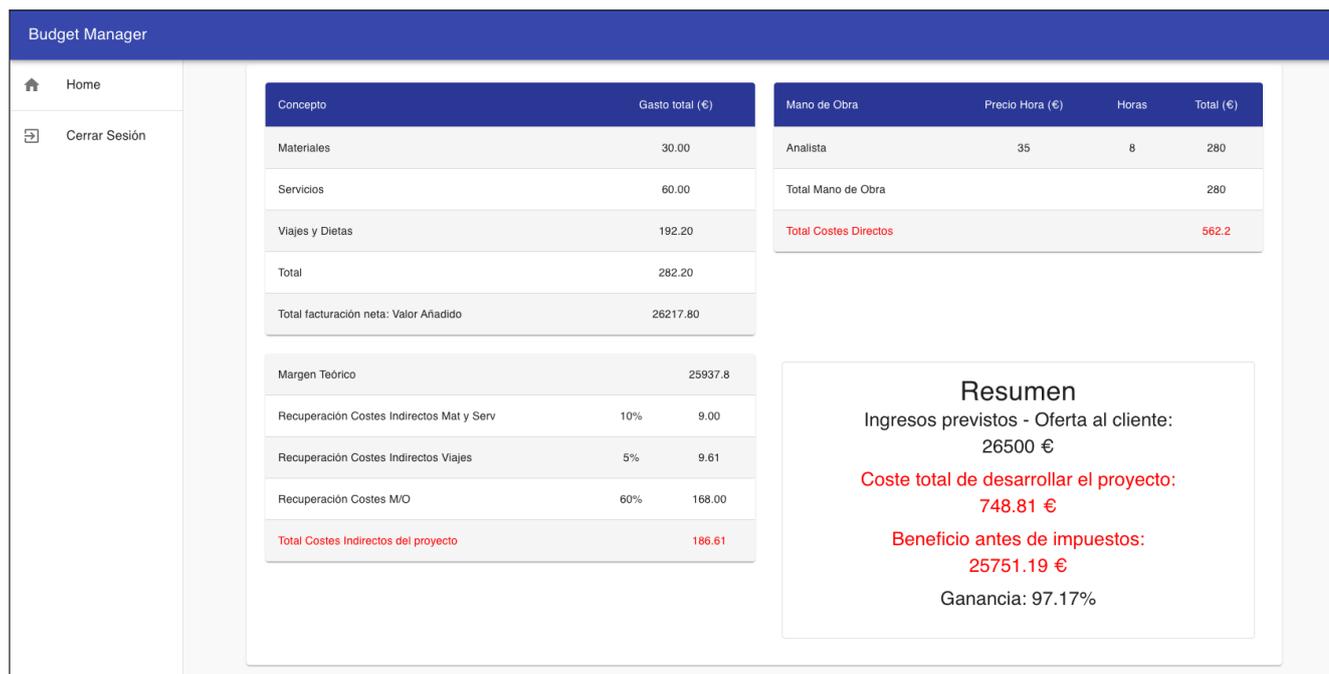


Figura 11.12: Resumen del Presupuesto

Finalmente, hay una serie de pestañas con cada una de las entidades que componen los presupuestos: Materiales, servicios, viajes y tareas. A continuación se explica con más detalle las acciones que se pueden llevar a cabo en dichas pestañas.

Materiales

Desde la pestaña de materiales, el usuario podrá visualizar todos los materiales que pertenecen al presupuesto (figura 11.15), añadir nuevos, editarlos y eliminarlos.

Además, como todos los materiales deben ir asociados a un artículo que haya sido creado previamente en el historial del sistema, el usuario también podrá ver el historial y crear nuevos artículos. Las acciones que se pueden llevar a cabo sobre el historial (figura 11.13) serán las siguientes:

- **Añadir un artículo:** Para añadir un artículo al historial, el usuario deberá pulsar el botón de 'AÑADIR' que se encuentra encima de la tabla del historial. Aparecerá un diálogo con un formulario que deberá rellenar con los datos del artículo (nombre y proveedor) tal y como se ha visto previamente para otros casos similares (como al añadir un cargo en la figura 11.7). Tras confirmar, se habrá creado el artículo en el historial del sistema.
- **Eliminar un artículo:** Procedimiento análogo al de eliminar un usuario del sistema. Se deberá seleccionar el icono del cubo de la basura y aceptar el diálogo de confirmación.
- **Visualizar historial de un artículo:** Para visualizar el historial de un artículo (es decir, en que otros presupuestos se ha utilizado, y la cantidad y precio que tuvo en ellos), el usuario debe pulsar el botón con el icono de un ojo de la fila de la tabla correspondiente al artículo que desee. Se le mostrará un cuadro de diálogo con la información (figura 11.16).

A continuación, las acciones que se podrán llevar a cabo sobre los materiales del presupuesto:

- **Añadir un material:** Para añadir un material, el usuario deberá pulsar el botón '+' situado al lado del artículo al que corresponda el material que quiere añadir. Una vez pulsado, el nombre del

artículo aparecerá en el campo correspondiente en el formulario, y el usuario deberá rellenar la cantidad y el precio del material y pulsar en 'AÑADIR'. El material se habrá creado y aparecerá en la tabla de materiales del presupuesto.

- **Editar un material:** Para editar un material, el usuario deberá pulsar el botón del lápiz situado en la fila de la tabla correspondiente al material que desee editar. En el formulario aparecerán los datos de dicho material para ser modificados. Una vez haya acabado, el usuario deberá pulsar 'GUARDAR' para actualizar los datos.
- **Eliminar un material:** El procedimiento es análogo al de eliminar un cargo del sistema. El usuario deberá pulsar el botón del cubo de la basura de la fila de la tabla correspondiente al material que desee eliminar y confirmar el cuadro de diálogo correspondiente.

La información con la que se contará de cada material del presupuesto será la siguiente:

- **Artículo:** Nombre del artículo del historial al que está asociado el material.
- **Cantidad**
- **Precio**
- **Proveedor**

Artículo	Proveedor	Acciones
Raspberry Pi	PCComponentes	+ 🔍 🗑️
QR TAG	PCComponentes	+ 🔍 🗑️

Figura 11.13: Historial de Artículos

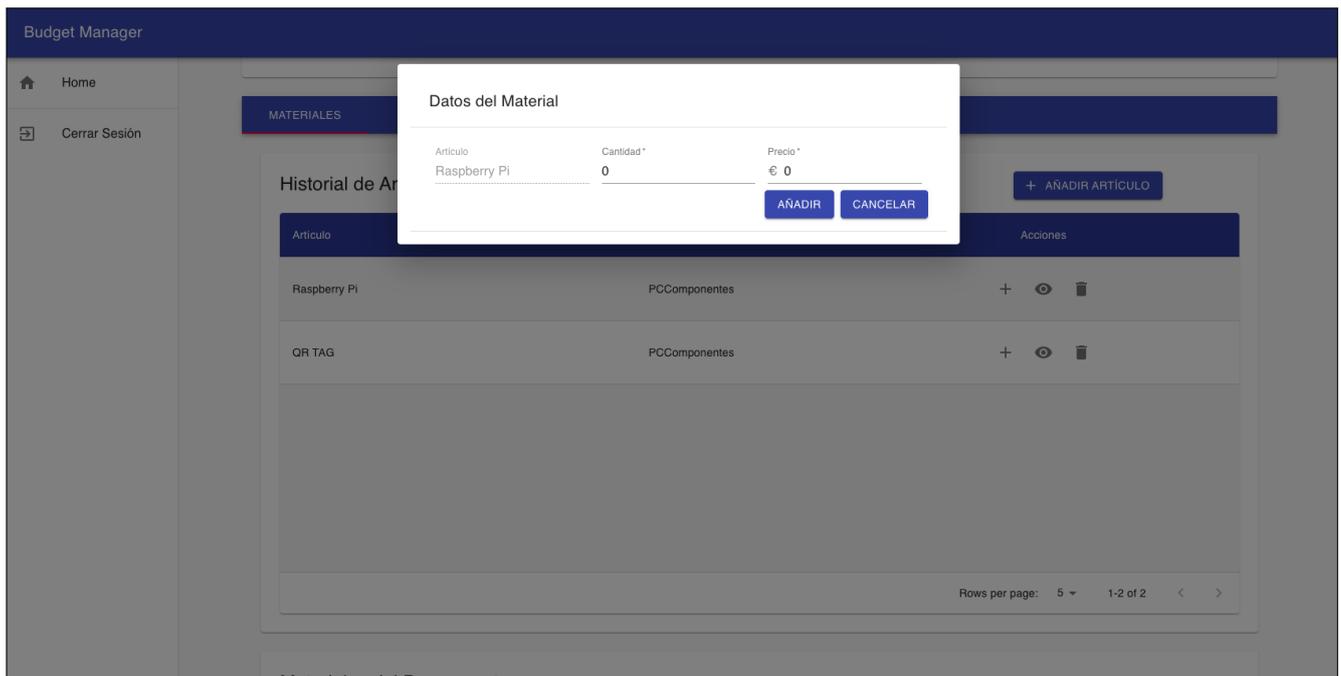


Figura 11.14: Formulario de Materiales

The image shows the 'Materiales del Presupuesto' table in the Budget Manager interface. The table has the following columns: Material, Cantidad, Precio, Proveedor, Total, and Acciones. The data is as follows:

Material	Cantidad	Precio	Proveedor	Total	Acciones
Raspberry Pi	2	35 €	PCComponentes	70 €	
QR TAG	1	23 €	PCComponentes	23 €	
Total				93 €	

At the top of the table area, there is a pagination control showing 'Rows per page: 5' and '1-2 of 2'. At the bottom right of the table area, there is another pagination control showing 'Rows per page: 5' and '1-2 of 2'.

Figura 11.15: Tabla de Materiales del Presupuesto

Budget Manager

Home

Cerrar Sesión

MATERIALES SERVICIOS

Historial de Artículos

Artículo

Raspberry Pi

QR TAG

Historial del Artículo

Presupuesto	Cantidad	Precio
App A	55	60 €
Sistema X	2	35 €
Sistema X(copia)	2	35 €
Sistema B	2	35 €
Proyecto K	3	30 €

CERRAR

ANADIR CANCELAR

Rows per page: 5 1-2 of 2

Materiales del Presupuesto

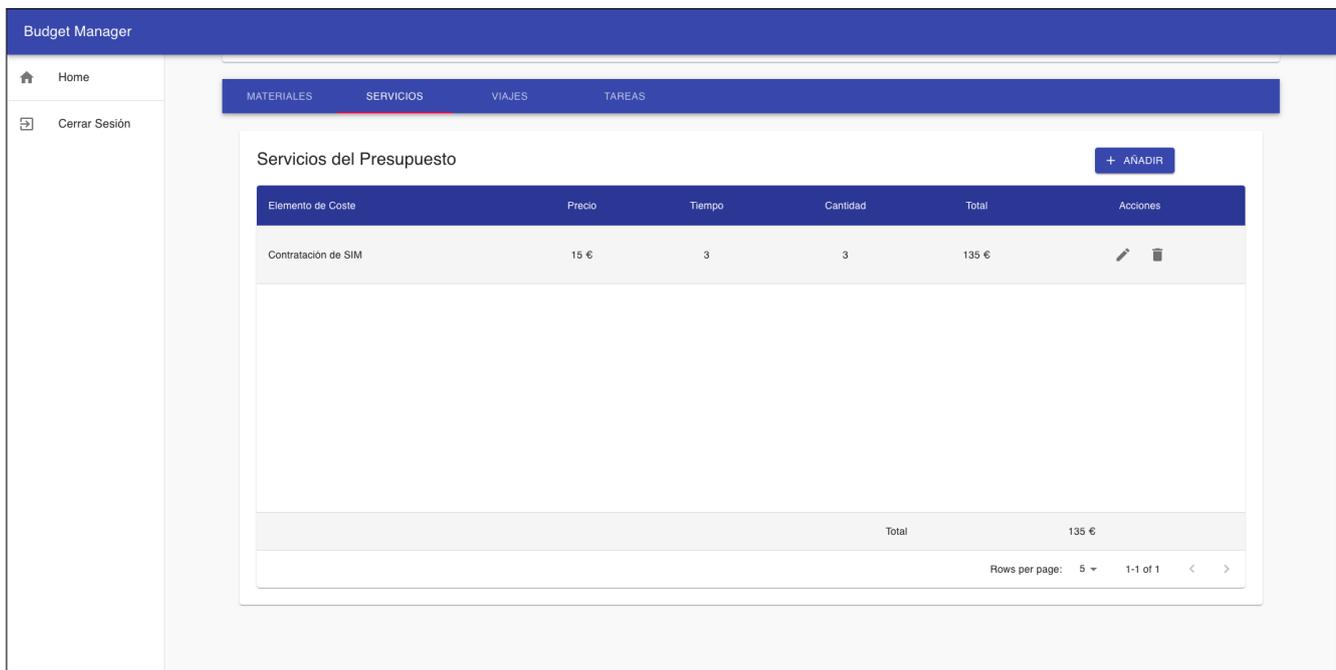
Material	Cantidad	Precio	Proveedor	Total	Acciones
----------	----------	--------	-----------	-------	----------

Figura 11.16: Historial de un Artículo

Servicios

Desde la pestaña de servicios, se podrán visualizar todos los servicios asociados al presupuesto (figura 11.17), añadir nuevos, editarlos y eliminarlos. No se detallará el procedimiento para estas acciones ya que es análogo a otras previamente vistas, como por ejemplo las acciones de gestión de cargos del sistema. La información con la que se contará de cada servicio del presupuesto será la siguiente:

- **Elemento de Coste**
- **Precio:** Coste del servicio en una unidad de tiempo.
- **Tiempo:** Periodo durante el que será necesario disponer de dicho servicio.
- **Cantidad**



The screenshot shows a web application interface for 'Budget Manager'. The main content area is titled 'Servicios del Presupuesto' and contains a table with the following data:

Elemento de Coste	Precio	Tiempo	Cantidad	Total	Acciones
Contratación de SIM	15 €	3	3	135 €	 
				Total	135 €

At the bottom right of the table, there is a pagination control: 'Rows per page: 5' and '1-1 of 1'.

Figura 11.17: Tabla de Servicios del Presupuesto

Viajes

Desde la pestaña de viajes, se podrán visualizar todos los viajes asociados al presupuesto (figura 11.18), añadir nuevos, editarlos y eliminarlos. No se detallará el procedimiento para estas acciones ya que es análogo a otras previamente vistas, como por ejemplo las acciones de gestión de cargos del sistema. La información con la que se contará de cada viaje del presupuesto será la siguiente:

- **Concepto**
- **Km:** Distancia del destino del viaje.
- **Total:** Coste del viaje teniendo en cuenta los km de distancia y el coste del km según los parámetros del sistema.
- **Tren:** Coste del tren (en el caso de que fuese necesario).
- **Avión:** Coste del avión (en el caso de que fuese necesario).
- **Otros:** Otros gastos necesarios para el viaje.

- **Días completos**
- **Días parciales**

Además, para cada viaje se calculará el total del coste teniendo en cuenta el coste de los km, el coste de transportes adicionales, y el coste de las dietas y hotel dependiendo de la cantidad de días que dure el viaje y los parámetros del sistema correspondientes (precio de dieta completa y parcial y coste de hotel)

Concepto	Km	Total	Trenes	Aviones	Otros	Dias	1/2 Dias	Cantidad	Total	Acciones
Reunion Final	150	33 €	0	0	0	5	1	1	525.5 €	[Edit] [Delete]
									Total	525.5 €

Rows per page: 5 | 1-1 of 1

Parámetros

- Dieta Completa Nacional: 35 €
- 1/2 Dieta Nacional: 17.5 €
- Coste Km: 0.22 €
- Coste Hotel: 60 €

Figura 11.18: Tabla de Viajes del Presupuesto

Tareas

Desde la pestaña de tareas, se podrán visualizar todas las tareas asociadas al presupuesto (figura 11.19), añadir nuevas, editarlas y eliminarlas. No se detallará el procedimiento para estas acciones ya que es análogo a otras previamente vistas, como por ejemplo las acciones de gestión de cargos del sistema. La información con la que se contará de cada tarea del presupuesto será la siguiente:

- **Concepto**
- **Unidades:** Número de veces a realizar la tarea.
- **Horas/Unidad:** Horas que se tarda en realizar la tarea.
- **Incertidumbre**
- **Perfil:** Para cada tarea, se podrá seleccionar entre los cargos del sistema dados de alta por el jefe, como ya se explicó previamente.

Para calcular el total de horas de cada tarea, se tendrá en cuenta su duración en horas, las unidades y la incertidumbre, ya que una mayor incertidumbre aumentará la estimación de horas. Además, se mostrará un cuadro resumen con la cantidad de horas que debe dedicar al proyecto cada uno de los cargos involucrados en él.

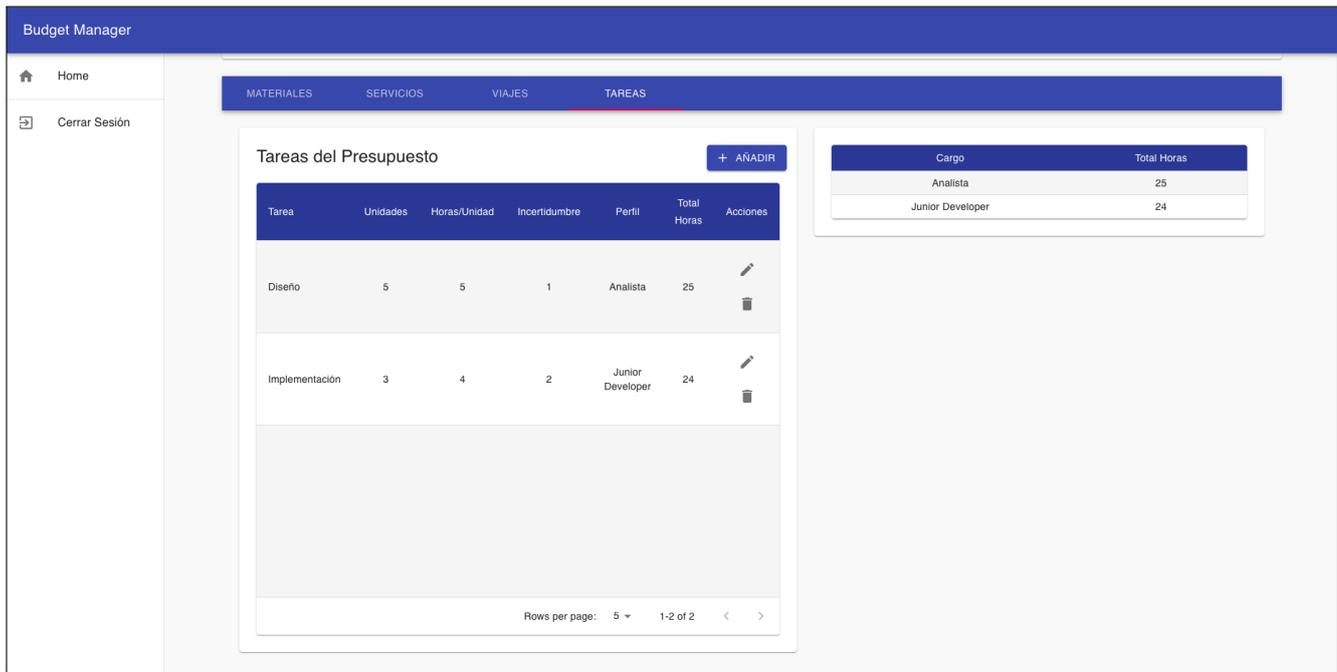


Figura 11.19: Tabla de Tareas del Presupuesto

11.5. Salir del sistema

Para cerrar sesión en el sistema, bastará con pulsar en el botón de la barra lateral de 'CERRAR SESIÓN', al que se puede acceder desde todas las vistas del mismo.

Capítulo 12

Conclusiones y trabajo futuro

12.1. Conclusiones

Durante el desarrollo de este TFG se ha llevado a cabo la implementación de distintas funcionalidades para realizar una aplicación administrativa sencilla y funcional, que permita gestionar presupuestos de proyectos software, así como los elementos asociados a ellos. Antes de la elaboración de este proyecto, dichas tareas se realizaban de forma manual utilizando software de hojas de cálculo, de forma que el proceso resultaba más difícil y menos estructurado, con un mayor margen de error, y además imposibilitaba la organización de grandes cantidades de datos.

A continuación se presentará un resumen de las tareas realizadas durante el desarrollo del proyecto:

- Se ha llevado a cabo la elicitación de requisitos, el análisis y el diseño del sistema.
- Se ha realizado un plan de proyecto que incluye una planificación temporal del desarrollo del mismo, riesgos y costes monetarios.
- Se han escrito una serie de manuales para facilitar el uso y mantenimiento del sistema, entre los que se incluyen manual de usuario, del programador y de instalación.
- Se ha llevado a cabo un proceso de investigación y aprendizaje del entorno de ejecución Javascript Node.js.
- Se ha desarrollado un API REST que permite gestionar y procesar la información necesaria para las distintas entidades del dominio del sistema. Además, se ha implementado un sistema de autenticación por medio de tokens para restringir el acceso a los endpoints a los usuarios autorizados.
- Se ha llevado a cabo un proceso de investigación y aprendizaje del framework Javascript React.js.
- Se ha desarrollado una interfaz web que aprovecha el API REST anteriormente mencionado, permitiendo llevar a cabo de forma sencilla y gráfica todas las tareas pertinentes a la gestión de presupuestos.
- Se ha implementado un sistema que permite dar de alta y de baja usuarios con distintos roles.
- Se ha implementado un sistema que permite dar de alta y de baja perfiles profesionales para usar en la asignación de tareas de los presupuestos.
- Se ha implementado un sistema que permite modificar parámetros globales del sistema que se utilizan para el cálculo de los distintos valores de los presupuestos.
- Se ha implementado un sistema que permite visualizar archivos en formato PDF de los presupuestos.

- Se ha implementado un sistema que permite la visualización, creación y eliminación de presupuestos.
- Se ha implementado un sistema que permite la edición de presupuestos, facilitando la realización de operaciones CRUD sobre las distintas partes de los mismos (materiales, servicios, viajes y tareas).
- Se ha implementado un sistema que permite visualizar un historial de los artículos utilizados y las cifras asociadas a ellos en distintos presupuestos del sistema.
- Se ha realizado una investigación y aprendizaje sobre la tecnología Docker.
- Se ha realizado el despliegue de toda la aplicación utilizando Docker.

12.2. Trabajo futuro

Para finalizar, se podrían considerar diversas líneas de trabajo mediante las que se puede ampliar y mejorar el funcionamiento del sistema desarrollado como fruto del trabajo realizado durante este TFG. A continuación se explican brevemente las vías de trabajo futuro que resultan más interesantes:

- Desarrollo de un sistema que permita a los usuarios crear planes de trabajo detallados basándose en los presupuestos, incluyendo diagramas de Gantt y otros datos útiles.
- Desarrollo de un panel de control que permita a los usuarios comparar el progreso actual de sus proyectos con el progreso que habían previsto en el plan de proyecto, avisándoles cuando ciertas cifras de control superan las cifras estimadas en el presupuesto inicial del que parte el proyecto.
- Desarrollar un sistema que permita a los usuarios visualizar estadísticas acerca del desarrollo de sus proyectos: cuántos de ellos se han salido de presupuesto, la media del beneficio obtenido en los distintos presupuestos de un período de tiempo determinado, etc.
- Realizar un desarrollo multiplataforma, que permita que el sistema pueda utilizarse en cualquier tipo de dispositivo.

Referencias

- [1] <https://hubstaff.com/>, Última visita: Febrero 2021.
- [2] Calc: Libreoffice en español - suite ofimática libre, basada en openoffice, compatible con microsoft. <https://es.libreoffice.org/descubre/calc/>, Última visita: Febrero 2021.
- [3] Empowering app development for developers. <https://www.docker.com/>, Última visita: Mayo 2021.
- [4] Express. <https://expressjs.com/es/>, Última visita: Marzo 2021.
- [5] Hojas de cálculo de google: crea y edita hojas de cálculo online de forma gratuita. <https://www.google.es/intl/es/sheets/about/>, Última visita: Febrero 2021.
- [6] Material ui: Un framework popular para ui en react. <https://material-ui.com/es/>, Última visita: Abril 2021.
- [7] Moqup. <https://moqups.com/es/>, Última visita: Marzo 2021.
- [8] Mysql. <https://www.mysql.com/>, Última visita: Marzo 2021.
- [9] Numbers. <https://www.apple.com/es/numbers/>, Última visita: Febrero 2021.
- [10] Passport.js. <http://www.passportjs.org/>, Última visita: Marzo 2021.
- [11] Project management software: Easy projects. <https://www.easyprojects.net/>, Última visita: Febrero 2021.
- [12] React pdf. <https://react-pdf.org/>, Última visita: Abril 2021.
- [13] React – una biblioteca de javascript para construir interfaces de usuario. <https://es.reactjs.org/>, Última visita: Abril 2021.
- [14] Sequelize. <https://sequelize.org/master/>, Última visita: Marzo 2021.
- [15] Software de hojas de cálculo microsoft excel: Microsoft 365. <https://www.microsoft.com/es-es/microsoft-365/excel>, Última visita: Febrero 2021.
- [16] Mvvm architectural pattern for a reactjs application, Apr 2018. <https://stackoverflow.com/questions/51506440/mvvm-architectural-pattern-for-a-reactjs-application?noredirect=1&lq=1>, Última visita: Mayo 2021.
- [17] Las dietas ¿cuándo debe recibirlas el trabajador?: Laboral 2021, Dec 2019. <https://loentiendo.com/dietas-desplazamiento-trabajadores/>, Última visita: Junio 2021.
- [18] What is a project budget? - example, overview, and what to include, Mar 2020. <https://corporatefinanceinstitute.com/resources/knowledge/finance/project-budget-overview/>, Última visita: Junio 2021.

- [19] May 2021. <https://docs.docker.com/>, Última visita: Mayo 2021.
- [20] High performance load balancer, web server, reverse proxy, May 2021. <https://www.nginx.com/>, Última visita: Mayo 2021.
- [21] React architecture best practices and tips from community experts: Simform, Apr 2021. <https://www.simform.com/react-architecture-best-practices/>, Última visita: Abril 2021.
- [22] Alexis Mora Angulo. ¿qué es y cómo utilizar localStorage y sessionStorage?, Dec 2018. <https://ed.team/blog/que-es-y-como-utilizar-localstorage-y-sessionstorage>, Última visita: Abril 2021.
- [23] Axios. axios/axios. <https://github.com/axios/axios>, Última visita: Abril 2021.
- [24] Node.js. Node.js. <https://nodejs.org/es/>, Última visita: Marzo 2021.
- [25] Postman. The collaboration platform for api development. <https://www.postman.com>, Última visita: Marzo 2021.
- [26] React. Componentes y propiedades. <https://es.reactjs.org/docs/components-and-props.html>, Última visita: Abril 2021.
- [27] React. Context. <https://es.reactjs.org/docs/context.html>, Última visita: Abril 2021.
- [28] React. Estado y ciclo de vida. <https://es.reactjs.org/docs/state-and-lifecycle.html>, Última visita: Abril 2021.
- [29] Leomaris Reyes. Aplicando el patrón de diseño mvvm, Dec 2018. <https://medium.com/aplicando-el-patron-de-dise~no-mvvm-d4156e51bbe5>, Última visita: Mayo 2021.
- [30] Mark Richards. Software architecture patterns. <https://www.oreilly.com/library/view/software-architecture-patterns/9781491971437/ch01.html>, Última visita: Abril 2021.
- [31] Tracy, Harwinder Singh, Bonnie, Tony, and Joel Espinosa Ayroso. Direct costs vs indirect costs in project management, Aug 2017. <https://www.deepfriedbrainproject.com/2017/08/direct-costs-vs-indirect-costs-project-management.html>, Última visita: Junio 2021.
- [32] Iryna Viter. Creating a project budget – a complete guide for 2021, Mar 2021. https://www.forecast.app/blog/how-to-create-a-project-budget?hs_amp=true#project-budget-checklist, Última visita: Junio 2021.