

GRADO EN INGENIERÍA EN SISTEMAS DE
INFORMACIÓN



Trabajo Fin de Grado

Análisis y selección de herramientas para soporte de proyectos
ágiles

ESCUELA POLITECNICA

Autor: Jorge López Pérez

Tutor: M. Carmen Pagés Arévalo

UNIVERSIDAD DE ALCALÁ
Escuela Politécnica Superior

**GRADO EN INGENIERÍA EN SISTEMAS DE
INFORMACIÓN**

Trabajo Fin de Grado

Análisis y selección de herramientas para soporte de proyectos ágiles

Autor: Jorge López Pérez

Tutor: M. Carmen Pagés Arévalo

TRIBUNAL:

Presidente:

Vocal 1º:

Vocal 2º:

FECHA:

“La medida de la inteligencia es la capacidad de cambiar”

Albert Einstein.

INDICE

1. RESUMEN	9
2. ABSTRACT	10
3. RESUMEN EXTENDIDO	11
4. GLOSARIO DE ACRÓNIMOS Y ABREVIATURAS	12
5. INTRODUCCIÓN.....	13
5.1 Presentación	13
5.2 Planteamiento y estructura de la memoria	13
5.3 Objetivos	14
6. ESTADO DEL ARTE	15
6.1 Concepto de metodología de desarrollo software	15
6.2 Las metodologías tradicionales o pesadas.....	15
6.2.1 Concepto.....	15
6.2.2 Características generales	15
6.2.3 Taxonomía.....	16
6.3 Las metodologías ágiles	19
6.3.1 Concepto.....	19
6.3.2 Características generales	21
6.3.3 Principales modelos de metodologías ágiles	21
6.3.3.1 Scrum	22
6.3.3.2 XP.....	23
6.3.3.3 Kanban	24
6.4 Factores de éxito y de fracaso en la aplicación de metodologías ágiles.....	25
6.5 Necesidades de soporte y beneficios del uso de herramientas en un proyecto ágil.....	27
6.5.1 Necesidades de soporte en un proyecto ágil.....	27
6.5.2 Beneficios del uso de herramientas de soporte en proyectos ágiles	28
7. ESTUDIO Y COMPARACIÓN DE HERRAMIENTAS DE SOPORTE DE PROYECTOS ÁGILES	29
7.1 Herramientas seleccionadas	29
7.2 Estudio individual	29
7.2.1 Trello	29
7.2.1.1 Características principales.....	29
7.2.1.2 Organización de los equipos de trabajo.....	31
7.2.1.3 Interfaz del menú principal.....	31
7.2.1.4 Revisión de funcionalidades versión gratuita.....	32
7.2.2 Jira Software.....	34
7.2.2.1 Características principales.....	34
7.2.2.2 Organización de los equipos de trabajo.....	35
7.2.2.3 Interfaz del menú principal.....	36
7.2.2.4 Revisión de funcionalidades versión gratuita.....	37
7.2.3 Asana.....	40
7.2.3.1 Características principales.....	40
7.2.3.2 Organización de los equipos de trabajo.....	41
7.2.3.3 Interfaz del menú principal.....	41
7.2.3.4 Revisión de funcionalidades versión gratuita.....	42
7.2.4 Wrike.....	45
7.2.4.1 Características principales.....	45
7.2.4.2 Organización de los equipos de trabajo.....	46
7.2.4.3 Interfaz del menú principal.....	46
7.2.4.4 Revisión de funcionalidades versión gratuita.....	47
7.2.5 iceScrum.....	48
7.2.5.1 Características principales.....	48
7.2.5.2 Organización de los equipos de trabajo.....	49

7.2.5.3	Interfaz del menú principal.....	50
7.2.5.4	Revisión de funcionalidades versión gratuita.....	50
7.2.6	ClickUp.....	54
7.2.6.1	Características principales.....	54
7.2.6.2	Organización de los equipos de trabajo.....	55
7.2.6.3	Interfaz del menú principal.....	56
7.2.6.4	Revisión de funcionalidades versión gratuita.....	57
7.2.7	Taiga.....	59
7.2.7.1	Características principales.....	59
7.2.7.2	Organización de los equipos de trabajo.....	60
7.2.7.3	Interfaz del menú principal.....	61
7.2.7.4	Revisión de funcionalidades versión gratuita.....	61
7.2.8	Freedcamp.....	63
7.2.8.1	Características principales.....	63
7.2.8.2	Organización de los equipos de trabajo.....	64
7.2.8.3	Interfaz del menú principal.....	65
7.2.8.4	Revisión de funcionalidades versión gratuita.....	66
7.2.9	Monday.com.....	68
7.2.9.1	Características principales.....	68
7.2.9.2	Organización de los equipos de trabajo.....	68
7.2.9.3	Interfaz del menú principal.....	69
7.2.9.4	Revisión de funcionalidades versión gratuita.....	70
7.2.10	Zoho Sprints.....	72
7.2.10.1	Características principales.....	72
7.2.10.2	Organización de los equipos de trabajo.....	73
7.2.10.3	Interfaz del menú principal.....	73
7.2.10.4	Revisión de funcionalidades versión gratuita.....	74
7.3	Criterios de comparación.....	75
7.3.1	Bloque 1 – Criterios generales.....	76
7.3.2	Bloque 2 – Almacenamiento y Soporte.....	76
7.3.3	Bloque 3 – Seguridad de la información.....	76
7.3.4	Bloque 4 – Funcionalidades.....	76
7.4	Tablas comparativas.....	77
7.4.1	Planes Cloud.....	78
7.4.1.1	Nivel 1.....	78
7.4.1.2	Nivel 2.....	79
7.4.1.3	Nivel 3.....	80
7.4.1.4	Nivel 4.....	81
7.4.1.5	Nivel 5.....	82
7.4.2	Planes On-Premise.....	83
8.	DESARROLLO DE APLICACIÓN WEB.....	84
8.1	Resumen Técnico.....	84
8.2	Creación del proyecto y estructura de directorios.....	84
8.3	Explicación detallada de los componentes del proyecto.....	85
8.4	Implementación de las funcionalidades.....	86
9.	CONCLUSIONES Y TRABAJO FUTURO.....	89
9.1	Conclusiones.....	89
9.2	Trabajo futuro.....	89
10.	ANEXOS.....	90
10.1	Manual de instalación de la aplicación web.....	90
10.2	Manual de usuario de la aplicación web.....	91
11.	REFERENCIAS.....	95

INDICE DE TABLAS Y FIGURAS

FIGURAS

<i>Figura 1 - Modelos en Cascada</i>	17
<i>Figura 2 - Modelos Evolutivos</i>	17
<i>Figura 3 - Modelo RUP</i>	18
<i>Figura 4 - Valores fundamentales del manifiesto ágil</i>	20
<i>Figura 5 - Encuesta realizada por el PMI (2017) sobre el uso de metodologías ágiles en las organizaciones [6]</i>	21
<i>Figura 6 - Pasos de un Sprint</i>	23
<i>Figura 7 - Marco de trabajo Sprint Scrum [7]</i>	23
<i>Figura 8 - Tablero Kanban</i>	25
<i>Figura 9 - Planes y precios de Trello</i>	30
<i>Figura 10 - Características incluidas planes Trello</i>	30
<i>Figura 11 - Plataformas Trello</i>	30
<i>Figura 12 - Organización equipos en Trello</i>	31
<i>Figura 13 - Barra de navegación superior de Trello</i>	32
<i>Figura 14 - Barra de navegación lateral de Trello</i>	32
<i>Figura 15 - Tablero en Trello</i>	32
<i>Figura 16 - Tarjeta en Trello</i>	33
<i>Figura 17 - Filtrado de tarjetas en Trello</i>	33
<i>Figura 18 - Creación de regla en Trello</i>	34
<i>Figura 19 - Power-Up Calendario Trello</i>	34
<i>Figura 20 - Planes y precios Jira</i>	35
<i>Figura 21 - Organización equipos en Jira</i>	36
<i>Figura 22 - Barra navegación superior Jira</i>	36
<i>Figura 23 - Barra de navegación lateral Jira</i>	36
<i>Figura 24 - Vista de "Backlog" Jira</i>	37
<i>Figura 25 - Vista de "Hoja de Ruta" Jira</i>	37
<i>Figura 26 - Vista tablero Jira</i>	38
<i>Figura 27 - Búsqueda avanzada de incidencias Jira</i>	38
<i>Figura 28 - Gráficas disponibles informes Jira</i>	39
<i>Figura 29 - Reglas de automatización disponibles Jira</i>	39
<i>Figura 30 - Integraciones Jira</i>	40
<i>Figura 31 - Planes y precios Asana</i>	40
<i>Figura 32 - Organización equipos en Asana</i>	41
<i>Figura 33 - Barra de navegación superior Asana</i>	41
<i>Figura 34 - Barra de navegación lateral Asana</i>	42
<i>Figura 35 - Creación de tarea Asana</i>	42
<i>Figura 36 - Vista tablero Asana</i>	43
<i>Figura 37 - Vista calendario Asana</i>	43
<i>Figura 38 - Gráficos informes Asana</i>	43
<i>Figura 39 - Secciones flujos de trabajo Asana</i>	44
<i>Figura 40 - Reglas flujos de trabajo Asana</i>	44
<i>Figura 41 - Creación objetivos Asana</i>	44
<i>Figura 42 - Integraciones Asana</i>	45
<i>Figura 43 - Planes y precios Wrike</i>	45
<i>Figura 44 - Organización equipos en Wrike</i>	46
<i>Figura 45 - Barra de navegación superior Wrike</i>	46
<i>Figura 46 - Barra de navegación lateral Wrike</i>	47
<i>Figura 47 - Creación de tarea Wrike</i>	47
<i>Figura 48 - Vista tablero Wrike</i>	48

<i>Figura 49 - Integraciones plan Free Wrike</i>	48
<i>Figura 50 - Planes y precios versiones Cloud iceScrum</i>	49
<i>Figura 51 - Planes y precios versiones On-premise iceScrum</i>	49
<i>Figura 52 - Organización equipos en iceScrum</i>	50
<i>Figura 53 - Pantalla de inicio iceScrum</i>	50
<i>Figura 54 - Barra de navegación superior iceScrum</i>	50
<i>Figura 55 - Vista "Dashboard" iceScrum</i>	51
<i>Figura 56 - Generación gráficos iceScrum</i>	51
<i>Figura 57 - Creación historia iceScrum</i>	52
<i>Figura 58 - Creación tarea iceScrum</i>	52
<i>Figura 59 - Vista de "Backlogs" iceScrum</i>	52
<i>Figura 60 - Vista de sprints iceScrum</i>	53
<i>Figura 61 - Vista de tablero iceScrum</i>	53
<i>Figura 62 - Creación característica iceScrum</i>	54
<i>Figura 63 - Integraciones iceScrum</i>	54
<i>Figura 64 - Planes y precios de ClickUp</i>	55
<i>Figura 65 - Organización equipos en ClickUp</i>	56
<i>Figura 66 - Barra de navegación superior ClickUp</i>	56
<i>Figura 67 - Creación de sprint ClickUp</i>	57
<i>Figura 68 - Creación de tarea ClickUp</i>	57
<i>Figura 69 - Vista de tablero ClickUp</i>	58
<i>Figura 70 - Vista Diagrama de Gantt ClickUp</i>	58
<i>Figura 71 - Integraciones ClickUp</i>	59
<i>Figura 72 - Creación de automatización ClickUp</i>	59
<i>Figura 73 - Planes y precios Taiga</i>	60
<i>Figura 74 - Organización equipos en Taiga</i>	60
<i>Figura 75 - Barra de navegación superior Taiga</i>	61
<i>Figura 76 - Barra de navegación lateral Taiga</i>	61
<i>Figura 77 - Creación historia Taiga</i>	62
<i>Figura 78 - Backlog y sprints Taiga</i>	62
<i>Figura 79 - Vista de tablero Taiga</i>	63
<i>Figura 80 - Issues en Taiga</i>	63
<i>Figura 81 - Planes y precios Freedcamp</i>	64
<i>Figura 82 - Organización equipos Freedcamp (1)</i>	65
<i>Figura 83 - Organización equipos Freedcamp (2)</i>	65
<i>Figura 84 - Barra de navegación lateral Freedcamp (vistas generales)</i>	65
<i>Figura 85 - Barra de navegación lateral Freedcamp (vistas del proyecto)</i>	66
<i>Figura 86 - Vista tablero Kanban Freedcamp</i>	67
<i>Figura 87 - Cronómetros de actividades Freedcamp</i>	67
<i>Figura 88 - Planes y precios Monday.com</i>	68
<i>Figura 89 - Organización equipos en Monday.com</i>	69
<i>Figura 90 - Barra principal Monday.com</i>	70
<i>Figura 91 - Barra work management Monday.com</i>	70
<i>Figura 92 - Vista tablero Kanban Monday.com</i>	71
<i>Figura 93 - Panel en Monday.com</i>	71
<i>Figura 94 - Plantilla plan de proyecto Monday.com</i>	72
<i>Figura 95 - Planes y precios Zoho Sprints</i>	72
<i>Figura 96 - Organización equipos Zoho Sprints</i>	73
<i>Figura 97 - Barra de navegación lateral Zoho Sprints</i>	74
<i>Figura 98 - Sprint creado en Zoho Sprints</i>	74
<i>Figura 99 - Backlog Zoho Sprints</i>	75
<i>Figura 100 - Vista "Dashboard" Zoho Sprints</i>	75
<i>Figura 101 - Estructura de repositorios</i>	85
<i>Figura 102 - Recogido de los datos introducidos en la pregunta 2</i>	86
<i>Figura 103 - Método encargado de calcular la puntuación de cada plan</i>	87

<i>Figura 104 - Extracto del método encargado de generar las respuestas en texto</i>	87
<i>Figura 105 - Tabla que muestra los resultados del mejor plan</i>	88
<i>Figura 106 - Ejecución de la clase principal del programa</i>	90
<i>Figura 107 - Aplicación iniciada y lista para usar</i>	90
<i>Figura 108 - "localhost:8080" en un navegador para usar la aplicación</i>	90
<i>Figura 109 - Comprobación de la versión Java instalada</i>	91
<i>Figura 110 - Ejecución del archivo .jar proporcionado</i>	91
<i>Figura 111 - Página principal de bienvenida</i>	92
<i>Figura 112 - Pregunta inicial del cuestionario</i>	92
<i>Figura 113 - Resto de preguntas del cuestionario</i>	93
<i>Figura 114 - Aviso de error si las preguntas no se rellenan correctamente</i>	93
<i>Figura 115 - Mejor plan para el cuestionario realizado de ejemplo</i>	94
<i>Figura 116 - Resultados del resto de planes</i>	94

TABLAS

<i>Tabla 1 - Clasificación de modelos concretos en clases abstractas [1]</i>	16
<i>Tabla 2 - Factores de fracaso metodologías ágiles [9]</i>	26
<i>Tabla 3 - Factores de éxito metodologías ágiles [9] [10]</i>	27
<i>Tabla 4 - Comparación planes Cloud nivel 1</i>	78
<i>Tabla 5 - Comparación planes Cloud nivel 2</i>	79
<i>Tabla 6 - Comparación planes Cloud nivel 3</i>	80
<i>Tabla 7 - Comparación planes Cloud nivel 4</i>	81
<i>Tabla 8 - Comparación planes Cloud nivel 5</i>	82
<i>Tabla 9 - Comparación planes On-Premise</i>	83

1. RESUMEN

En el presente trabajo se realiza un estudio teórico tanto de las metodologías pesadas como de las metodologías ágiles en la gestión de proyectos software, en el que se revisan sus características, tipos y bondades, al mismo tiempo que se estudia la importancia de la utilización de herramientas de soporte en proyectos ágiles.

Además, se seleccionan un grupo de diez herramientas de soporte para su estudio y posterior comparación, en base a unos criterios comunes establecidos. Por último, se desarrolla una aplicación web para ayudar a las organizaciones o usuarios a decantarse por una herramienta u otra (de las estudiadas) según sus necesidades particulares.

Palabras clave: Gestión de proyectos, Desarrollo de software, Metodologías ágiles, Herramientas de soporte, Análisis, Selección.

2. ABSTRACT

In this project, a theoretical study is conducted on both traditional methodologies and agile methodologies in software project management, reviewing their characteristics, types, and advantages, while also examining the importance of using support tools in agile projects.

Furthermore, a group of ten support tools is selected for study and subsequent comparison, based on established common criteria. Finally, a web application is developed to assist organizations or users in choosing one tool or another (from the ones studied) according to their specific needs.

Keywords: Project management, Software development, Agile methodologies, Support tools, Analysis, Selection.

3. RESUMEN EXTENDIDO

En el presente trabajo, se lleva a cabo un estudio teórico sobre las metodologías pesadas y las metodologías ágiles en la gestión de proyectos software. Se analizan detalladamente sus características, tipos y beneficios, con el objetivo de comprender mejor su aplicación en diferentes contextos.

En primer lugar, se examinan las metodologías pesadas, también conocidas como tradicionales, que son aquellas que se caracterizan por su rigidez frente al cambio y por focalizarse en aspectos como la documentación, la planificación, los procesos y la comunicación escrita. Se describen sus principales características y su taxonomía (en la que vemos sus principales modelos como los evolutivos o en cascada).

Posteriormente, se profundiza en las metodologías ágiles, que han ganado popularidad en los últimos años debido a su enfoque flexible y adaptativo. Se exponen los valores y principios del manifiesto ágil y se describen algunos de sus principales modelos: Scrum, Kanban y XP. Además, se abordan los factores que se han identificado comúnmente como de éxito o fracaso en la aplicación de este tipo de metodologías.

En el contexto de las metodologías ágiles, se destaca la importancia y bondades de utilizar herramientas de soporte adecuadas para optimizar la gestión de proyectos. Con este fin, se realiza una selección de diez herramientas de soporte que se consideran relevantes y efectivas en el entorno ágil (teniendo en cuenta para la elección de dichas herramientas factores como la disponibilidad de versiones gratuitas).

Una vez seleccionadas las herramientas, se procede a un análisis individual de cada una de ellas en la que se analizan una serie de criterios básicos como su licencia de uso, sus planes y precios o las plataformas en las que está disponible. Además, se evalúa la forma en la que se organizan los grupos de trabajo dentro de esta, su interfaz principal y sus funcionalidades.

Posteriormente, se establecen una serie de criterios de comparación basados en cuatro bloques: aspectos generales, almacenamiento y soporte, seguridad de la información y funcionalidades. Con estos criterios se realiza un análisis comparativo detallado de las herramientas estudiadas previamente, con todos sus planes disponibles, donde se evalúan estos aspectos, con el objetivo de tener una visión general de lo que nos ofrecen todas las herramientas.

Como resultado final de este trabajo, se desarrolla una aplicación web personalizada que, mediante un cuestionario, actúa como una guía para ayudar a las organizaciones o usuarios a seleccionar la herramienta de soporte más adecuada para sus proyectos ágiles. Esta aplicación proporciona al usuario, en base a sus requerimientos, qué herramientas tienen un porcentaje de coincidencia mayor respecto a lo que solicitan, mostrando qué aspectos concretos incorpora dicha herramienta y cuáles no.

4. GLOSARIO DE ACRÓNIMOS Y ABREVIATURAS

RUP: Proceso Racional Unificado (Rational Unified Process)

XP: Programación Extrema (Extreme Programming)

FDD: Desarrollo Orientado a Funcionalidades (Feature Driven Development)

ASD: Desarrollo Adaptativo de Software (Adaptive Software Development)

WIP: Trabajo en progreso (Work in Progress)

USD: Dólar estadounidense (United States Dollar)

IDE: Entorno de Desarrollo Integrado (Integrated Development Environment)

MVC: Modelo Vista Controlador

JS: JavaScript

JDK: Kit de Desarrollo de Java (Java Kit Development)

CMD: Comando (Command)

5. INTRODUCCIÓN

En este apartado, se realizará una breve presentación del trabajo y se describirá el planteamiento a seguir, la estructura de la memoria y los objetivos a lograr.

5.1 Presentación

En el mundo empresarial y tecnológico dinámico y cambiante en el que vivimos, las organizaciones buscan constantemente formas de mejorar su eficiencia, calidad y colaboración en sus proyectos. Es en este punto donde entran en juego las metodologías ágiles y las herramientas de soporte.

En el pasado, las metodologías pesadas o tradicionales se basaban en procesos rígidos y secuenciales. Sin embargo, con la llegada de la necesidad de adaptarse rápidamente a los cambios, surgió otra necesidad: una metodología más flexible y ágil. Es por ello, que, en los últimos años, las metodologías ágiles (como Scrum, Kanban o XP) se han convertido en las opciones más populares para gestionar proyectos software de manera eficiente y efectiva.

Sin embargo, para el éxito de un proyecto, no sólo basta con adoptar una metodología ágil adecuada, sino también contar con las herramientas de soporte adecuadas que permitan una implementación eficiente de estas metodologías. La elección de la herramienta o herramientas de soporte adecuadas es esencial para garantizar el éxito de los proyectos ágiles. Y en los últimos años, con el crecimiento de la popularidad de estas metodologías y el aumento del uso de este tipo de herramientas, el número de herramientas de soporte disponibles en el mercado ha crecido de manera considerable.

En el presente trabajo, se aborda este problema y se trata de hacer este proceso de elección más sencillo, comparando y estudiando algunas de las herramientas de soporte más conocidas, a la vez que se desarrolla una aplicación web para ayudar en la elección de la herramienta que mejor se ajuste a las necesidades y características del usuario.

5.2 Planteamiento y estructura de la memoria

En este trabajo se va a realizar, en primer lugar, un estudio del estado del arte (**punto 6**) en el que se analizará desde el punto de vista teórico las metodologías tradicionales o pesadas (**punto 6.2**) y las metodologías ágiles o ligeras (**punto 6.3**). Veremos el concepto de ambas, las características que las definen y sus principales tipos. Además, en el caso de las metodologías ágiles veremos sus principales factores de éxito y fracaso (condiciones que se suelen dar en los equipos de trabajo y que acaban llevando al proyecto a una situación propicia al éxito o al fracaso) (**punto 6.4**), sus necesidades de soporte y los beneficios derivados del uso de herramientas de soporte en proyectos ágiles (**punto 6.5**).

Posteriormente, se realizará una selección de 10 herramientas de soporte de proyectos ágiles (**punto 7.1**) y se estudiará cada una en detalle (tanto sus características básicas, como sus funcionalidades, como la forma en la que se organizan los equipos de trabajo) (**punto 7.2**). Una vez realizado el estudio individual de cada herramienta, establecerán unos criterios de comparación (**punto 7.3**) y se compararán unas con otras teniendo en cuenta todos los planes disponibles de cada una (**punto 7.4**).

Y, por último, se elaborará una aplicación web que tendrá como objetivo, en base a un cuestionario, ayudar a las organizaciones o usuarios a elegir la herramienta de soporte que mejor se adapte a sus necesidades y características (**punto 8**).

5.3 Objetivos

El objetivo principal del trabajo consiste en **seleccionar y estudiar las herramientas de soporte de proyectos ágiles más destacadas del mercado, para posteriormente compararlas en base a unos criterios y elaborar una aplicación web que ayude a las empresas o usuarios en la tarea de decidir cuál de ellas es más adecuada según sus necesidades.**

Los objetivos específicos planteados son los siguientes:

1. Realizar un estudio teórico de las metodologías ágiles y las metodologías pesadas.
2. Documentar las principales necesidades de soporte en proyectos ágiles.
3. Describir las ventajas y beneficios del uso de herramientas de soporte en la gestión de proyectos ágiles.
4. Realizar un estudio de las plataformas de soporte de proyectos ágiles disponibles en el mercado.
5. Seleccionar las plataformas de soporte más destacadas y utilizadas, de las disponibles en el mercado.
6. Realizar un estudio detallado tanto del punto de vista teórico como práctico de cada una de las herramientas seleccionadas, exponiendo sus características y funcionalidades.
7. Establecer una lista de criterios de comparación para dichas herramientas.
8. Realizar una comparación de las plataformas estudiadas en base a los criterios de comparación establecidos (comparando los planes de cada una de ellas).
9. Programar una aplicación web que permita a los usuarios, mediante un cuestionario, la selección de la mejor herramienta según sus necesidades y características particulares.

6. ESTADO DEL ARTE

6.1 Concepto de metodología de desarrollo software

Para conseguir entender los conceptos de metodologías ágiles y tradicionales (o también llamadas pesadas) necesitamos definir previamente qué es una metodología de desarrollo de software.

A lo largo del proceso de desarrollo de un sistema software, se realizan una serie de actividades las cuales, en su conjunto, son llamadas “**proceso de desarrollo de software**”. Estas actividades son las siguientes: especificación de requisitos, diseño, codificación, validación (o pruebas) y mantenimiento. [1]

Este proceso de desarrollo, conformado por las actividades anteriormente descritas, también se le conoce como “**ciclo de vida del software**”, ya que dichas actividades conforman las etapas de la vida por las que pasa el software: nace con una especificación de requisitos determinados, se lleva a cabo su implantación (se diseña el sistema, se codifica y se prueba), se lleva a cabo la entrega al cliente y, por último, se lleva a cabo el mantenimiento de dicho software, hasta que finalmente se deja de utilizar o se desecha (muere). [1]

La **metodología de desarrollo de software** constituye una representación abstracta de todo este proceso de desarrollo, es decir, indica el orden, los procedimientos y las técnicas que deben ser seguidas durante todo el proceso de desarrollo del sistema. [1]

La elección de la metodología de desarrollo correcta y que más se adecue a las necesidades particulares del proyecto en cuestión, marcará en gran medida el éxito del proyecto, la calidad del producto final y los tiempos y costes asociados a este. Es por ello, que, en los puntos siguientes, veremos los dos grandes grupos de metodologías de desarrollo software (ágiles y pesadas), sus características y en qué circunstancias es más adecuado cada una de ellas.

6.2 Las metodologías tradicionales o pesadas

6.2.1 Concepto

Las metodologías tradicionales o pesadas son aquellas que promueven y se focalizan en la documentación, la planificación, los procesos y la comunicación escrita. La documentación exhaustiva de todas las partes del proceso de desarrollo y el cumplimiento del plan de proyecto elaborado al inicio de este, son las principales características de este tipo de modelos. Las metodologías tradicionales también se caracterizan por su rigidez frente al cambio, es decir, cambiar el plan de proyecto a lo largo del desarrollo supone un esfuerzo mayor que en las metodologías ligeras. [2]

6.2.2 Características generales

Las metodologías tradicionales comparten las siguientes características generales: [4]

- a) Se basan en las normas provenientes de estándares seguidos por el entorno de desarrollo.
- b) Hacen énfasis en la definición del proceso: roles, actividades y artefactos.
- c) Resistencia frente al cambio (más o menos incisiva dependiendo del modelo concreto).

- d) Impuestos de manera externa.
- e) Proceso muy controlado, con numerosas normas.
- f) Se basan en un contrato prefijado.
- g) El cliente interactúa con el equipo de desarrollo mediante reuniones (en algunos modelos, incluso, el cliente no interactúa hasta la entrega del producto).
- h) Grupos grandes, con más roles y específicos.
- i) Suelen ser más efectivos en proyectos grandes.
- j) La arquitectura seguida es esencial y se define en las partes iniciales del proyecto.

6.2.3 Taxonomía

Los principales modelos pesados o tradicionales podemos agruparlos, por sus características, en cuatro grandes grupos o modelos abstractos: en cascada, evolutivos, minimización de desarrollos e híbridos. [1]

Modelo Abstracto		Modelo Concreto
Tradicionales	En cascada	Pura Con fases solapadas Con subproyectos Con reducción de riesgos
	Evolutivos	Espiral Entrega por etapas Entrega iterativa Diseño por planificación Cascada en V
	Minimización de desarrollos	Componentes reutilizables Diseño por herramientas
	Híbridos	Proceso Unificado Racional (RUP) Otros

Tabla 1 - Clasificación de modelos concretos en clases abstractas [1]

- **Modelos en Cascada:** estos modelos se caracterizan por dividir las fases del proceso de desarrollo (o ciclo de vida) de manera lineal, de manera que una vez que se supera una fase no se vuelve a ejecutar de nuevo.

Dentro de este enfoque, encontramos diferentes variantes (modelos concretos), como el modelo en cascada con fases solapadas, el cual permite realizar una actividad de una fase al mismo tiempo que se desarrolla otra actividad de la fase anterior, o el modelo en cascada con subproyectos, que divide el proyecto en subproyectos de tamaño más reducido.

Casos de Aplicación: en los modelos en cascada, por la naturaleza de estos, los requerimientos del sistema deberán estar especificados desde el inicio del proyecto y debe existir pocas probabilidades de que cambien a lo largo de este, es decir, su aplicación puede ser recomendada en proyectos poco flexibles y bien especificados desde el inicio (escenario que, sobre todo, hoy en día, es bastante complicado de darse).

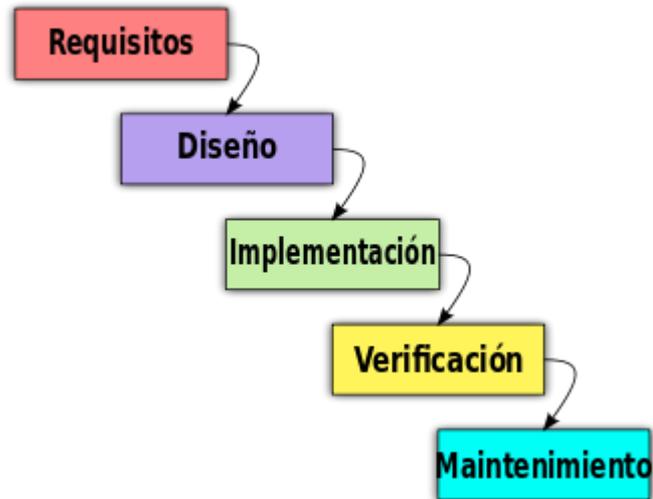


Figura 1 - Modelos en Cascada

- **Modelos evolutivos:** los modelos evolutivos se caracterizan por, a diferencia de los modelos en cascada, ejecutar las diferentes fases del proyecto tantas veces como sea necesario, sin un orden determinado, es decir, de la fase 4 se puede volver a la 1 sin tener que pasar por la 2 y la 3.

Casos de Aplicación: los modelos evolutivos, se recomiendan en situaciones en las que se comience con una serie de requisitos prioritarios y se completen el resto en ciclos posteriores. Además, se requiere de una participación activa del cliente, quien debe evaluar y dar su “feedback” en cada entrega parcial.



Figura 2 - Modelos Evolutivos

- **Modelos de minimización de desarrollo:** este tipo de modelos se fundamentan en la idea de aprovechar aquellos elementos o componentes que han sido desarrollados previamente. No hacen tanto énfasis en el cumplimiento estricto de los requisitos, sino en la minimización del tiempo y esfuerzo de desarrollo. Se ofrece al cliente un sistema que sea fácil de implementar y solo en caso de ser necesario, se propone un desarrollo nuevo.

Casos de Aplicación: en los modelos de desarrollo mínimo, es necesario que los requisitos del sistema a desarrollar sean similares a los de los componentes que se

disponen para ser reutilizados, de manera que el esfuerzo de modificar dichos componentes no sea demasiado elevado.

- **Modelos Híbridos:** los modelos híbridos reúnen elementos de modelos de procesos genéricos.

El modelo de esta categoría más importante es la **metodología RUP** (Proceso Racional Unificado “Rational Unified Process”) desarrollada por “Rational Software”, una compañía de IBM. Esta metodología describe una forma efectiva de aplicar procedimientos comerciales probados en el desarrollo de software. [3]

El modelo RUP, se sustenta principalmente en 3 características esenciales:

- **Está dirigido por casos de uso:** los casos de uso representan un fragmento de funcionalidad del sistema, y no solo sirven para especificar los requisitos del sistema, sino también, para guiar su diseño, implementación y pruebas.
- **Es un proceso centrado en la arquitectura:** la arquitectura de un sistema es la representación de sus partes más importantes o representativas, que permite tener una visión clara y global del sistema completo. En la metodología RUP se establece una arquitectura temprana lo más completa posible para que en fases posteriores se vea lo menos alterada posible. Dicha arquitectura se va consolidando mediante líneas base y modificando según las necesidades del proyecto.
- **Es un proceso iterativo e incremental:** el trabajo del proyecto se divide en partes más pequeñas, o también llamadas mini proyectos. Cada mini proyecto se puede asemejar a una iteración en la cual se obtiene un avance o crecimiento del producto. De esta forma se consigue un equilibrio entre casos de uso y arquitectura. El proceso iterativo puede llevarse a cabo siguiendo otras técnicas vistas, como, por ejemplo, mediante una cascada.

En general, todos los procesos que siguen la metodología RUP, pueden verse representados de la siguiente manera (Figura 3).

El eje horizontal representa el tiempo y los aspectos dinámicos del proceso, en términos de fases. En el caso de RUP, se suele dividir en 4 fases: inicio, elaboración, construcción y transición.

En el eje vertical se representan los componentes estáticos, es decir, las actividades o flujos de trabajo.

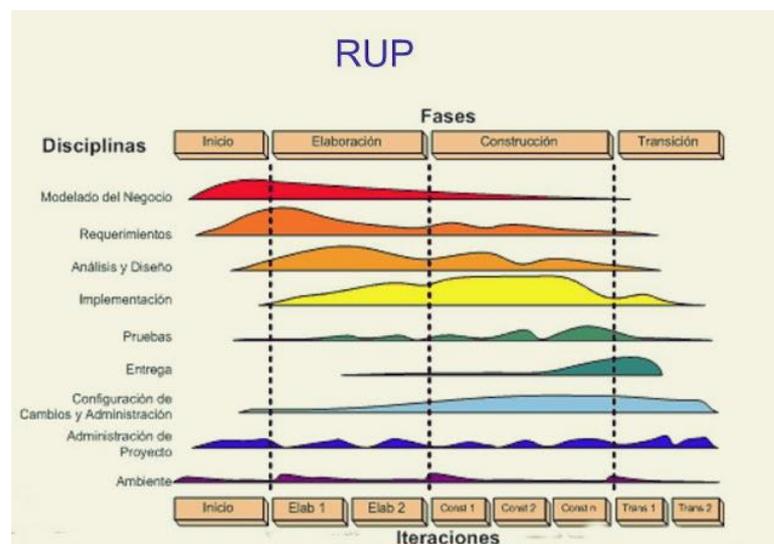


Figura 3 - Modelo RUP

Casos de Aplicación: en general, los modelos híbridos, debido a que sacan las mejores prácticas de cada uno de los modelos vistos anteriormente, puede llegar a ser el modelo más equilibrado de los vistos, en cuanto a costos, tiempo de desarrollo, riesgos y flexibilidad frente a cambios de requisitos. Sin embargo, debido a su alta complejidad respecto a otros modelos, requiere de una alta capacitación de todos los miembros del equipo de desarrollo.

6.3 Las metodologías ágiles

6.3.1 Concepto

A partir de los años 90, con la masificación de internet, cambia la manera en la que se demanda el software: mayor celeridad de desarrollo y requisitos más imprecisos y cambiantes.

Cuando se intentaron aplicar las metodologías tradicionales o pesadas, que hemos visto previamente, en proyectos con estas características, se obtuvieron resultados deficientes, como consecuencia del gran tiempo que demandan estas metodologías en términos de planificación y diseño, y de la rigidez asociada, que no proporciona flexibilidad frente al cambio. Los proyectos se convertían en procesos improductivos e ineficientes. [4]

De esta manera, las metodologías ágiles o ligeras surgieron como necesidad para solucionar estos problemas.

Con el nacimiento de esta nueva forma de afrontar el desarrollo de software, se redactó en el año 2001 el “*manifiesto ágil*” como base fundamental de todas las metodologías ligeras. En dicho manifiesto se recogen los 4 valores fundamentales y los 12 principios a seguir.

Los **cuatro valores fundamentales** del manifiesto son los siguientes: [5]

- a) **El individuo, las interacciones y la comunicación por encima de los procesos y herramientas:** el recurso humano es el principal factor de éxito.
- b) **Prioridad en el software por encima de la documentación:** la documentación se debe limitar a lo fundamental.
- c) **La colaboración con el cliente por encima de la negociación del contrato:** la participación del cliente debe ser constante durante todas las fases del proyecto.
- d) **Capacidad de respuesta frente al cambio por encima del seguimiento de un plan:** planificación adaptativa (los cambios en las especificaciones durante el desarrollo son muy frecuentes en el mercado actual).



Figura 4 - Valores fundamentales del manifiesto ágil

Citaremos, además, los **12 principios** descritos en el manifiesto: [5]

- I. Nuestra mayor prioridad es satisfacer al cliente mediante entregas tempranas y continuas de software con valor.
- II. Bienvenidos los cambios a los requerimientos, incluso los tardíos. Los procesos ágiles aprovechan los cambios para la ventaja competitiva del cliente.
- III. Liberar frecuentemente software funcionando, desde un par de semanas a un par de meses, con preferencia por los periodos más cortos.
- IV. Las personas de negocio y los trabajadores deben trabajar juntos diariamente a lo largo del proyecto.
- V. Construir proyectos en torno a individuos motivados. Darles el entorno y apoyo que necesiten, y confiar en ellos para que consigan hacer su trabajo.
- VI. El método más efectivo y eficiente de compartir información a, y dentro de un equipo de desarrollo, es la conversación cara a cara.
- VII. El software funcionando es la medida de progreso.
- VIII. Los procesos ágiles promueven el desarrollo sostenible. Los patrocinadores, desarrolladores y usuarios deberían ser capaces de mantener relaciones cordiales.
- IX. La atención continua a la excelencia técnica y al buen diseño incrementa la agilidad.
- X. La simplicidad (el arte de maximizar la cantidad de trabajo no hecho) es esencial.
- XI. Las mejores arquitecturas, requerimientos y diseños emergen de los equipos auto organizados.
- XII. En intervalos regulares, el equipo reflexiona sobre cómo volverse más efectivo, entonces afina y ajusta su comportamiento como corresponde.

Hoy en día, las metodologías ligeras o ágiles son tan importantes y utilizadas por las organizaciones, que incluso, según recoge el informe de 2017 que realiza anualmente el PMI llamado “*Pulse of the Profession*”, el 71% de las empresas encuestadas en dicho informe reportaron usar metodologías ágiles a veces, frecuentemente o siempre, por solo el 12% que indicaba no usar nunca este tipo de metodologías (Figura 5). [6]

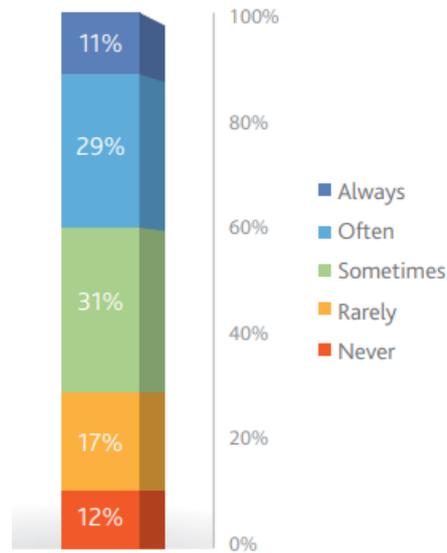


Figura 5 - Encuesta realizada por el PMI (2017) sobre el uso de metodologías ágiles en las organizaciones [6]

6.3.2 Características generales

En contraposición a las características generales de las metodologías tradicionales, enumeraremos también las principales características de las metodologías ágiles, para visualizar la claras diferencias entre ambas: [4]

- a) Se basan en conceptos y métodos provenientes de prácticas de producción de código.
- b) Hacen énfasis en los aspectos humanos: el individuo y el trabajo en equipo.
- c) Mejor preparados frente al cambio durante el proyecto.
- d) Impuestas de manera interna por el equipo.
- e) Proceso menos controlado, sin normas específicas.
- f) Contrato flexible o inexistente.
- g) El cliente es parte del desarrollo.
- h) Grupos pequeños, con pocos roles, más genéricos y flexibles.
- i) Orientados, generalmente, a proyectos de tamaño pequeño.
- j) La arquitectura software se puede ir definiendo y mejorando a lo largo de las fases del proyecto, no es necesario definirla de manera temprana y precisa.

6.3.3 Principales modelos de metodologías ágiles

Dentro de la familia de metodologías ágiles podemos citar numerosos modelos como pueden ser: Scrum, Kanban, Extreme Programming (XP), Lean, Crystal, Desarrollo orientado a funcionalidades (FDD), Desarrollo adaptativo de software (ASD), etc.

En nuestro caso, comentaremos los modelos que han sido más relevantes y utilizados en los últimos años: Scrum, XP y Kanban.

6.3.3.1 Scrum

Podemos definir Scrum como un marco de trabajo cuyo objetivo principal es la colaboración eficaz de equipos en proyectos. Para ello, se definen una serie de reglas, artefactos y roles que generan la estructura necesaria para el correcto funcionamiento de este. [7]

Scrum tiene un **enfoque incremental** (el producto final se va construyendo de manera progresiva) y se fundamenta en la **teoría de control empírico de los procesos**, la cual tiene como bases la transparencia (que garantiza visibilidad en cualquier aspecto que pueda afectar al resultado), la inspección (permite detectar variaciones no deseadas en el proceso) y la adaptación (permite reducir el impacto de dichas variaciones, mediante acciones correctivas).

Los **equipos de trabajo en Scrum** tienen las siguientes características: son **auto-gestionados** (no siguen directrices de personas externas al equipo de trabajo, sino que eligen ellos mismos la mejor forma de realizar el trabajo), son **multifuncionales** (tienen todos los conocimientos necesarios para realizar el trabajo) y **trabajan mediante iteraciones** (las entregas se hacen en iteraciones).

Dentro de Scrum tenemos además tres tipos de roles: el **Scrum Master** o dueño del proceso, cuya función es garantizar que se adoptan los valores y normas que se deben seguir (no gestiona el desarrollo), el **Dueño del producto**, que es una sola persona que representa a los interesados (clientes) (es el encargado, además, de gestionar el *Product Backlog* del que hablaremos más adelante) y el **Equipo de desarrollo**, cuya función es convertir el *Product Backlog* en iteraciones funcionales del producto. Todos los miembros del equipo de desarrollo tienen el mismo cargo: desarrollador, es decir, no hay jerarquías.

El proceso de desarrollo en Scrum se organiza en torno a la figura de los **Sprints** (podemos observar la estructura de un sprint en la Figura 6). Cada Sprint representa un periodo de tiempo (que no debe ser superior a 1 mes) en el cual se crea una versión utilizable del producto o también llamado “incremento”. Cada Sprint es considerado un proyecto independiente.

Un Sprint está conformado por los siguientes pasos o fases: [7]

1. **Planeación del sprint:** consistente en una reunión en la que se define qué se va a entregar y cómo se va a lograr.
2. **Daily Scrum:** reunión diaria del equipo de desarrollo (de unos quince minutos de duración) en el que cada miembro explica al resto lo que ha avanzado respecto a la reunión anterior, los obstáculos que ha encontrado en el camino (si los hay) y lo que hará para la siguiente reunión.
3. **Revisión del sprint:** reunión al final del sprint (de unas cuatro horas en sprints de 1 mes) en el que el dueño del producto revisa el trabajo realizado, revisa lo que no se ha hecho y discute acerca del *Product Backlog*. Por su parte, el equipo de desarrollo expone los problemas encontrados, cómo fueron resueltos, y, por último, muestra el producto resultado de la iteración en funcionamiento.
4. **Retrospectiva del sprint:** es la última fase de cada sprint. consiste en una reunión (de unas tres horas), en la que, como dice el propio nombre, se hace retrospectiva de todo el sprint, es decir, el equipo de desarrollo discute acerca de todo el proceso: qué estuvo bien, qué estuvo mal, y se elabora un plan de mejoras para el siguiente sprint, a fin de eliminar los problemas encontrados.

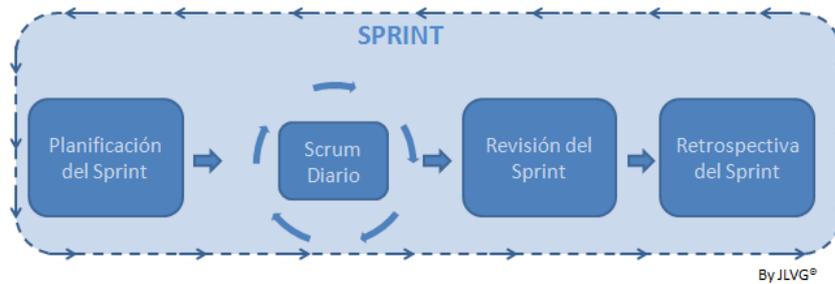


Figura 6 - Pasos de un Sprint

El marco de trabajo completo de cada Sprint Scrum es el siguiente (Figura 7):

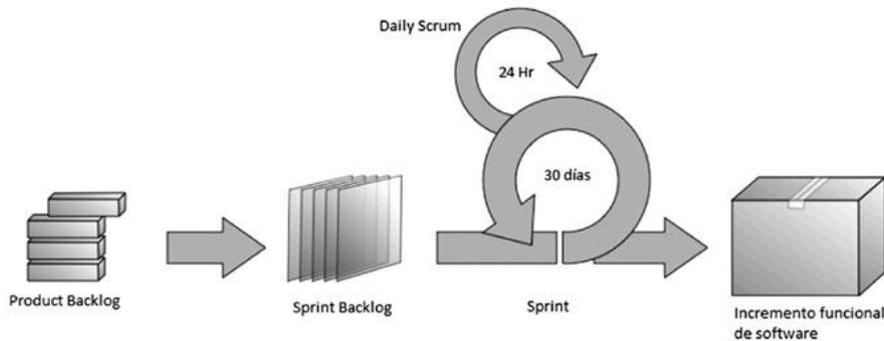


Figura 7 - Marco de trabajo Sprint Scrum [7]

Como podemos observar en la Figura 7, se parte de un *Product Backlog*, que son los requerimientos o requisitos que el dueño del producto define y va actualizando, es decir, es una lista que nunca está terminada, evoluciona de manera constante a lo largo del proyecto. Seguidamente, se elabora un *Sprint Backlog*, que es una sublista de ítems del *Product Backlog*, es decir, son los requerimientos en los que se trabajará en ese Sprint (se eligen por orden de prioridad, según el *Product Backlog*). A continuación, se inicia el Sprint con las fases que ya hemos visto, y una vez terminado el Sprint, llegamos al *incremento funcional del software*, es decir, la suma de todos los ítems terminados (funcionales) del *Sprint Backlog*. El producto a entregar en cada Sprint es la suma de todos los ítems que han sido terminados hasta la fecha.

6.3.3.2 XP

Extreme Programming, por sus siglas “XP”, es un modelo ágil de desarrollo software ideal para equipos de tamaño mediano o pequeño, y en proyectos con requisitos cambiantes o imprecisos.

XP se compone de una serie de valores y de prácticas que definen las bases de la metodología. Las prácticas son las actividades que se realizan en el día a día dentro del equipo de trabajo, y los valores, por su parte, marcan una forma de trabajar en base a unos valores personales y corporativos. Es decir, los valores aportan un propósito a las prácticas.

Los cinco valores de Extreme Programming son los siguientes: [8]

1. **Comunicación:** XP expone que muchos de los problemas en un proceso de desarrollo surgen por la falta de comunicación. Se debe trabajar en todos los procesos de manera conjunta y resolver los problemas entre todos.
2. **Simplicidad:** se deben intentar hacer las cosas de la manera más simple posible (hacer lo que se necesita y se pide).
3. **Feedback:** se necesita feedback constante de trabajos pasados para identificar las áreas en las que se debe mejorar y hacer el diseño más simple.
4. **Coraje:** se debe decir la verdad sobre el progreso y las estimaciones.

5. **Respeto:** los miembros del proyecto se deben respetar entre ellos y aceptar el feedback proveniente del resto.

Por otro lado, listaremos algunas de las principales prácticas del Extreme Programming: [8]

1. **Relacionados con la satisfacción del cliente:** diseñar un producto avanzado y con una alta tecnología no sirve si no se tiene en cuenta lo que quiere el cliente.

- a. **Entregas pequeñas:** como ya hemos mencionado en numerosas ocasiones, en los entornos ágiles, los requerimientos son cambiantes e imprecisos, es por ello, que realizar pequeñas entregas en plazos cortos de tiempo es beneficioso para ambas partes: reduce la probabilidad de cometer errores por parte de los desarrolladores y permite conocer el trabajo hecho y la capacidad del equipo de trabajo por parte del cliente.

2. Relacionados con la calidad del software:

- a. **Diseño simple:** en un entorno cambiante, la documentación es también escasa. Es por ello, que en XP se promueve el utilizar diseños simples y código fácil de entender con la menor documentación posible.
- b. **Refactorización:** los negocios están en constante proceso de cambio y mejora. Es por ello, que el código debe ser flexible para ser modificado, pero sin necesidad de cambiar la estructura básica del código o la funcionalidad del software para ello.
- c. **Pair programming:** XP también promueve la técnica conocida como “programación en pareja”, que consiste en que dos personas trabajen al mismo tiempo en un mismo código. Una de las personas se encarga de codificar y la otra de supervisar. De esta forma, ambas personas trabajan de forma colaborativa en la resolución de problemas, identificación de errores o desarrollo de ideas más creativas. Aunque esta técnica aumenta el coste de producción, se consigue: aumentar la calidad del software, reducir el tiempo de desarrollo y aumentar la satisfacción del equipo de desarrollo.

3. Relacionados con la planificación del proyecto:

- a. **Planning game:** el objetivo de esta práctica es definir los objetivos y planificar la siguiente entrega. Se deben definir las características más importantes y que deben ser añadidas a la entrega, el esfuerzo requerido, el número de iteraciones necesarias para completarla y cuándo se deberá tener lista dicha entrega.

6.3.3.3 Kanban

Kanban es un modelo de desarrollo de software ágil, que se centra en encontrar un equilibrio entre el trabajo que se necesita realizar y la disponibilidad con la que cuenta cada miembro del equipo de desarrollo. Para ello, hace hincapié en la gestión de la cantidad de trabajo en curso (Work in progress: WIP).

Se implementa por medio de los llamados **tableros Kanban**. Dichos tableros permiten visualizar de manera gráfica los flujos de trabajo y la carga de trabajo, y se organizan mediante columnas que representan etapas del trabajo.

La representación más simple de un tablero Kanban suele ser el siguiente: tres columnas (To Do, In Progress y Done), donde la primera columna representa las tareas por hacer, la segunda las tareas que están siendo realizadas y la tercera las tareas que ya han sido completadas. Cada una de las tareas es representada por una tarjeta, que tiene asignada una determinada importancia o prioridad y un responsable o responsables.

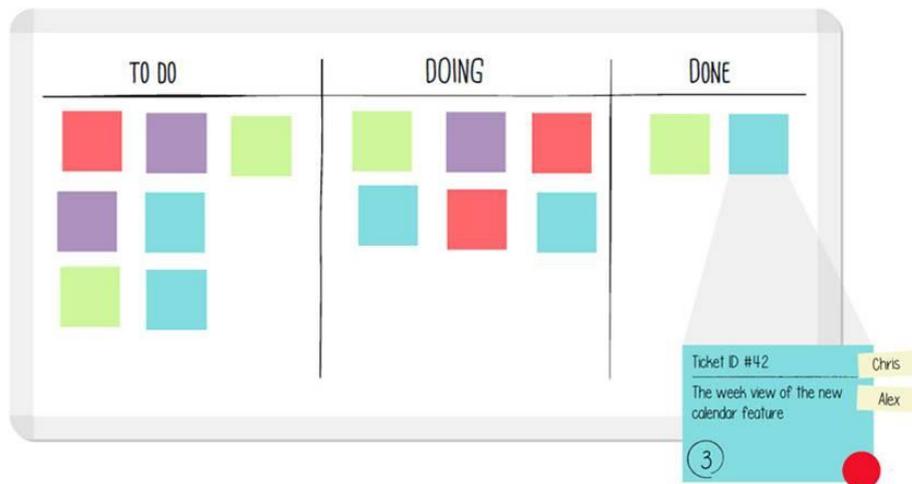


Figura 8 - Tablero Kanban

La cantidad de tareas presentes en la columna de “Work in progress” o “Doing”, marcan la capacidad o límites de trabajo y permiten enfocarse en una pequeña cantidad de trabajo a la vez. Estos límites pueden ajustarse a lo largo del proyecto según el esfuerzo esperado.

Una diferencia particular de Kanban es que, a diferencia de Scrum o XP, no tiene por qué ser un proceso iterativo, es decir, Kanban permite que el proceso de desarrollo se realice en un solo ciclo.

6.4 Factores de éxito y de fracaso en la aplicación de metodologías ágiles

En muchos casos, se aplican metodologías ágiles en entornos y equipos de trabajo en los que este tipo de metodologías acaban fracasando. En este punto describiremos algunos factores que han sido recabados como de éxito y de fracaso, es decir, condiciones que se suelen dar en los equipos de trabajo y que acaban llevando al proyecto a una situación propicia al éxito o, todo lo contrario.

Primero, veremos una serie de factores comunes que han llevado al fracaso en la aplicación de metodologías ágiles. Estos factores se recopilan en base a “lecciones aprendidas” de proyectos pasados. Dichos factores se organizan, según su naturaleza, en dimensiones: [9]

Factores de fracaso	
Dimensión	Factor
Organizacional	<ol style="list-style-type: none"> 1. Falta de compromiso de la dirección 2. Cultura tradicional de la organización 3. Tamaño de la organización muy grande
Personas	<ol style="list-style-type: none"> 4. Falta de competencia en gestión de proyectos 5. Falta de trabajo en equipo 6. Reticencias de ciertos grupos o individuos del equipo 7. Mala comunicación y relación con el cliente

Proceso	<ul style="list-style-type: none"> 8. Alcance del proyecto mal definido 9. Requisitos del proyecto mal definidos 10. Planificación del proyecto mal definido 11. Falta de mecanismos de seguimiento de proyectos ágiles 12. Falta de presencia del cliente en el proceso
Técnica	<ul style="list-style-type: none"> 13. Herramientas y tecnologías no apropiadas 14. Falta de un conjunto completo de prácticas ágiles correctas

Tabla 2 - Factores de fracaso metodologías ágiles [9]

Una vez vistos aquellos factores comunes que llevan al fracaso de los proyectos ágiles, veremos el lado opuesto, es decir, aquellos factores o características que acaban llevando a buen puerto al proyecto: [9] [10]

Factores de éxito	
Dimensión	Factor
Organización	<ul style="list-style-type: none"> 1. Apoyo de la dirección 2. Cultura de la organización cooperativa en lugar de jerárquica 3. Metodologías ágiles globalmente aceptadas dentro de la organización
Personas	<ul style="list-style-type: none"> 4. Alta experiencia y competencia en los miembros del equipo 5. Motivación en los miembros del equipo 6. Gerentes con un estilo de gestión adaptativo 7. Equipo de trabajo autoorganizado 8. Buena relación con el cliente 9. Comunicación y feedback efectivo
Proceso	<ul style="list-style-type: none"> 10. Fuerte comunicación con reuniones diarias 11. Alta presencia del cliente en el proceso 12. El cliente tiene plena autoridad en el proceso. 13. Reportes continuados del estado del proyecto.
Proyecto	<ul style="list-style-type: none"> 14. Proyecto con calendario realista 15. Proyecto sin múltiples equipos independientes 16. Proyecto con una buena estimación de costes (presupuesto realista) 17. Proyecto con un buen análisis de riesgos
Técnica	<ul style="list-style-type: none"> 18. Cantidad de documentación justa 19. Entregas regulares 20. Entregar los requerimientos más importantes primero 21. Correcta integración de los tests

	22. Buen entrenamiento técnico del equipo
	23. Familiarización con las tecnologías usadas

Tabla 3 - Factores de éxito metodologías ágiles [9] [10]

6.5 Necesidades de soporte y beneficios del uso de herramientas en un proyecto ágil

6.5.1 Necesidades de soporte en un proyecto ágil

Ya hemos visto qué son las metodologías ágiles y cuáles son los factores de éxito y fracaso más comunes en este tipo de proyectos. En este punto, veremos cuáles son las necesidades de soporte que nos podemos encontrar habitualmente en un proyecto ágil y que podemos solucionar gracias a herramientas de soporte.

Dichas necesidades, si logramos solventarlas de manera exitosa, están estrechamente relacionadas con el cumplimiento de los factores de éxito y con evitar los factores de fracaso vistos anteriormente.

Las necesidades de soporte en proyectos ágiles más relevantes son las siguientes:

1. **Organización y planificación:** las herramientas de soporte nos ayudarán en gran parte a la organización y planificación del trabajo, ya sea mediante gestión de tareas, flujos de trabajo, calendarios, etc.
2. **Visión general del proyecto:** las herramientas de soporte permiten, tanto al equipo de desarrollo como a la dirección, tener una visión clara y general del estado del proyecto, como: tareas asignadas, plazos de entrega, etc. Esto ayudará en la toma de decisiones y a ajustar la estrategia del proyecto si es necesario.
3. **Comunicación entre los miembros del equipo:** la comunicación es una pieza fundamental en cualquier proyecto ágil. Las herramientas de soporte nos pueden ayudar a tener una interacción constante y fluida entre todos los miembros del equipo de trabajo, mediante, por ejemplo, chats grupales o videollamadas.
4. **Colaboración entre los miembros del equipo:** la colaboración es otro punto fundamental en todo proyecto ágil. Las herramientas de soporte nos proporcionarán la forma de colaborar de forma efectiva y en tiempo real con, por ejemplo, sistemas de compartición de archivos, sistemas de tickets, etc.
5. **Contacto con el cliente:** las herramientas de soporte simplifican de manera significativa este proceso. Se suelen incorporar sistemas de tickets para quejas, mensajes o consultas entre el equipo de desarrollo y el cliente. De esta forma, se puede responder al cliente de una manera más rápida y efectiva, aumentando su satisfacción.
6. **Control de versiones y seguridad:** es muy importante también en un proyecto ágil llevar un control de versiones efectivo, para que todos los miembros del equipo de desarrollo sepan en todo momento en qué versión del código están o deben trabajar. Además, pueden proporcionar mecanismos de backups o copias de seguridad para reaccionar de manera más rápida y eficiente frente a errores humanos.
7. **Retroalimentación:** el aprender del trabajo realizado y mejorar en la siguiente entrega, también es una parte fundamental para perfeccionar el estilo de trabajo. Es por ello, que este tipo de herramientas suelen contar con recogida de datos e información valiosa para su posterior visualización, como, por ejemplo, indicadores de rendimiento o recomendaciones de mejora. Así, de esta forma podemos ver, por ejemplo, qué

miembro del equipo ha sido más productivo y por qué o en qué fase del proyecto hemos tardado más de lo debido.

Como hemos visto, las necesidades de soporte en proyectos ágiles son numerosas y, es por eso, que la utilización de este tipo de herramientas, para solventarlas, se antoja primordial para el éxito del proyecto.

6.5.2 Beneficios del uso de herramientas de soporte en proyectos ágiles

En base a las necesidades planteadas, algunos de los beneficios que nos aportan estas herramientas son: una mejor coordinación y planificación del proyecto, una mejor redistribución de la carga de trabajo, un mejor seguimiento del estado del proyecto para una mejor toma de decisiones, una mayor comunicación y colaboración entre los miembros del equipo de trabajo, una mayor participación y satisfacción del cliente en todo el proceso, y, en general, una mayor productividad.

Además, también se ven afectados tres factores clave en cualquier proyecto software: el **tiempo**, el **coste** y la **calidad** del producto (los proyectos ágiles que incorporan herramientas de soporte experimentan una identificación y corrección de errores o defectos más temprana y rápida, lo que deriva en una mayor calidad del producto, un aumento en la velocidad de entrega del producto y una reducción de costes).

7. ESTUDIO Y COMPARACIÓN DE HERRAMIENTAS DE SOPORTE DE PROYECTOS ÁGILES

Una vez visto el estado del arte de las metodologías ágiles (en comparación con las metodologías pesadas o tradicionales, los principales modelos utilizados, sus factores de éxito y fracaso más comunes y sus necesidades de soporte), en este apartado 7 seleccionaremos algunas de las principales herramientas de soporte de proyectos ágiles que podemos encontrar en el mercado, haremos un estudio individual para cada una de ellas, estableceremos unos criterios de comparación y, por último, las compararemos en base a esos criterios establecidos.

7.1 Herramientas seleccionadas

Se han seleccionado para su estudio y comparación un total de diez herramientas de soporte de proyectos ágiles. Son las siguientes:

1. Trello
2. Jira Software
3. Asana
4. Wrike
5. iceScrum
6. ClickUp
7. Taiga
8. Freedcamp
9. Monday.com
10. Zoho Sprints

Para la selección de las herramientas mencionadas se ha tenido en cuenta la importancia y relevancia de dichas plataformas y que dispongan de una versión gratuita, para la comprobación de su utilización de manera práctica y para que el estudio de cada una de ellas sea justo.

7.2 Estudio individual

En este apartado se estudiará de manera individual cada una de las diez herramientas de soporte de proyectos ágiles seleccionadas. **En cada uno de estos estudios, para profundizar y revisar de manera práctica las funcionalidades de cada herramienta, se utilizará la versión gratuita de la herramienta correspondiente**, con el fin de que haya equidad en la versión utilizada (no sería justo realizar los estudios con la versión gratuita de Trello y con la Premium de Jira, por ejemplo). Sin embargo, para el apartado 7.4 sí que tendremos en cuenta todas las versiones disponibles para compararlas (siempre con su equivalente de la herramienta rival).

7.2.1 Trello

7.2.1.1 Características principales

- **Página web oficial:** <https://trello.com/es>
- **Compañía:** Atlassian Corporation Plc.
- **Fecha de lanzamiento:** 13 de septiembre de 2011.
- **Licencia:** Software propietario (no libre).
- **Planes y precios:** Trello dispone de 4 planes a elegir: plan **Gratuito** (0 USD), plan **Standard** (5 USD por usuario al mes con facturación anual o 6 USD con facturación

ESTUDIO Y COMPARACIÓN DE HERRAMIENTAS DE SOPORTE DE PROYECTOS ÁGILES

mensual), plan **Premium** (10 USD por usuario al mes con facturación anual o 12.50 USD con facturación mensual) y el plan **Enterprise** (17.50 USD por usuario al mes con facturación anual, sólo disponible a partir de 50 usuarios). Todas las versiones de Trello funcionan en la nube, es decir, no cuentan con ningún plan de implementación local.

FREE	STANDARD	PREMIUM	ENTERPRISE
<p>\$0 USD</p> <p>Gratis para todo el equipo</p> <p>Para personas o equipos que quieren organizar cualquier proyecto que se les ocurra.</p>	<p>\$5 USD</p> <p>por usuario al mes con facturación anual (\$5 con facturación mensual)</p> <p>Para los equipos pequeños que necesitan gestionar el trabajo y escalar la colaboración.</p>	<p>\$10 USD</p> <p>por usuario al mes con facturación anual (\$12.50 con facturación mensual)</p> <p>Para equipos que necesiten supervisar y visualizar varios proyectos de varias maneras, como en tableros, cronogramas, calendarios, etc.</p>	<p>\$17.50 USD</p> <p>por usuario al mes con facturación anual (\$210.00 precio anual por usuario)</p> <p>Para organizaciones que necesitan conectar el trabajo de distintos equipos con mayor seguridad y más controles.</p> <p>Coste estimado para 50 usuarios</p>

Figura 9 - Planes y precios de Trello

Las características incluidas en cada plan son las siguientes:

<p>QUÉ INCLUYE EL PLAN FREE:</p> <ul style="list-style-type: none"> ✓ Tarjetas ilimitadas ✓ Hasta 10 tableros por Espacio de trabajo ✓ Power-Ups ilimitados por tablero ✓ Almacenamiento ilimitado (10 MB/archivo) ✓ 250 ejecuciones de comandos por Espacio de trabajo al mes ✓ Fondos y pegatinas personalizados ✓ Registros de actividad ilimitados ✓ Fechas de vencimiento y personas asignadas ✓ Aplicaciones móviles para iOS y Android ✓ Autenticación en dos fases 	<p>TODAS LAS PRESTACIONES DEL PLAN FREE Y, ADEMÁS:</p> <ul style="list-style-type: none"> ✓ Tableros ilimitados ✓ Checklists avanzadas ✓ Campos personalizados ✓ Almacenamiento ilimitado (250 MB/archivo) ✓ 1000 ejecuciones de comandos por Espacio de trabajo al mes ✓ Invitados de un solo tablero ✓ Búsquedas guardadas 	<p>TODAS LAS PRESTACIONES DEL PLAN STANDARD, ADEMÁS DE:</p> <ul style="list-style-type: none"> ✓ Vistas: Calendario, Cronograma, Tabla, Panel y Mapa ✓ Vistas del Espacio de trabajo: Tabla y Calendario ✓ Ejecuciones ilimitadas de comandos en el Espacio de trabajo ✓ Funciones de administración y seguridad ✓ Plantillas de nivel de Espacio de trabajo ✓ Colecciones ✓ Observadores ✓ Soporte Priority ✓ Exportación sencilla de datos 	<p>TODAS LAS PRESTACIONES DEL PLAN PREMIUM Y, ADEMÁS:</p> <ul style="list-style-type: none"> ✓ Espacios de trabajo ilimitados ✓ Permisos a nivel de organización ✓ Tableros visibles a nivel de organización ✓ Gestión de tableros públicos ✓ Invitados multitablero ✓ Permisos de adjuntos ✓ Administración de Power-Ups ✓ Cuentas SSO gratuitas y provisionamiento de usuarios con Atlassian Access
---	--	--	--

Figura 10 - Características incluidas planes Trello

- **Plataformas:** Trello se puede usar desde un navegador web, desde un móvil con su App (IOS o Android) o desde un ordenador (MAC o Windows).

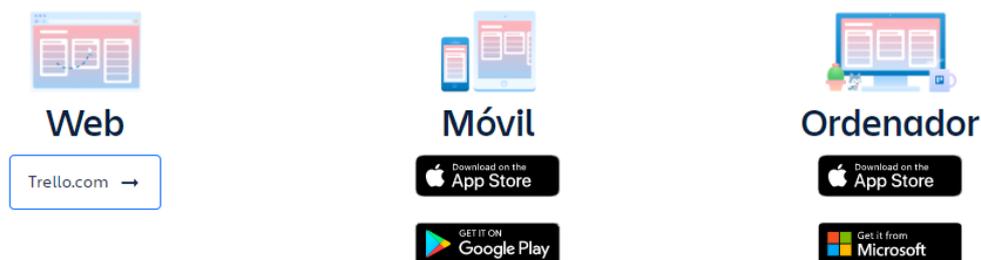


Figura 11 - Plataformas Trello

- **Grado de documentación / información:** Alta (en la página web oficial de Trello se ofrece de manera gratuita una guía de introducción a la herramienta por capítulos y enlaces a seminarios web, también gratuitos, a través de su canal de YouTube. Además, por si tuvieses algún problema concreto, dispones de documentación oficial bastante completa, un apartado de preguntas frecuentes y un foro para comentar dudas con otros usuarios de Trello).

7.2.1.2 Organización de los equipos de trabajo

En Trello los usuarios se organizan en torno a los **espacios de trabajo**. Como usuario, podemos unirnos a varios espacios de trabajo (el número de espacios de trabajo en los que podemos estar depende del plan que tengamos, el único que tiene espacios de trabajo ilimitados es el plan Enterprise). En una organización, lo más común, es que se necesiten varios espacios de trabajo.

Por otro lado, en cada espacio de trabajo podemos crear varios tableros (en el plan gratuito podemos crear hasta 10 tableros por espacio de trabajo, en el resto son ilimitados) y, además, tenemos puntos de acceso de información compartida por todo el espacio como la vista de tabla o calendario.

En la figura 12 ilustramos la forma en la que se organizan los miembros de una compañía en Trello:

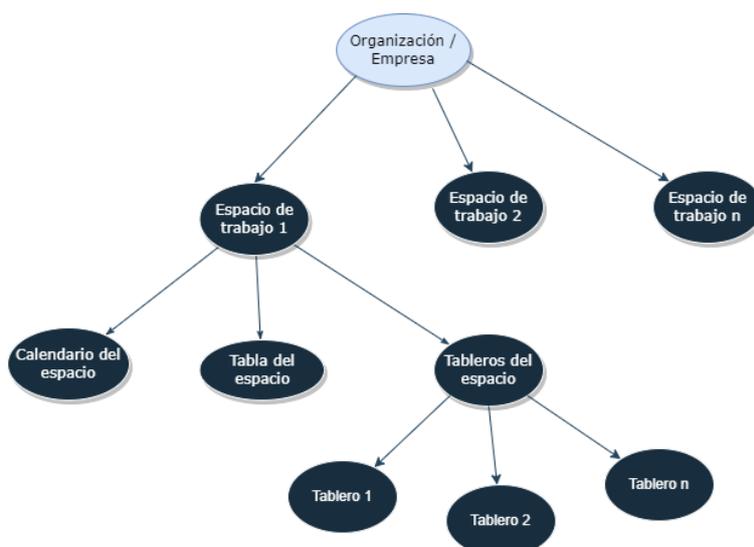


Figura 12 - Organización equipos en Trello

7.2.1.3 Interfaz del menú principal

La interfaz del menú principal de Trello consta de una barra de navegación superior y de una barra de navegación lateral.

En la barra superior nos encontramos con las opciones principales: cambiar de espacio de trabajo, ver los tableros recientes consultados, consultar los tableros marcados como favoritos a modo de acceso directo, explorar plantillas de personalización para nuestros tableros y un botón para crear nuevos espacios de trabajo, tableros o plantillas. Además, Trello nos permite cambiar de manera simple y rápida entre las distintas aplicaciones de Atlassian (con el botón más a la izquierda de la imagen), como, por ejemplo, Jira:

ESTUDIO Y COMPARACIÓN DE HERRAMIENTAS DE SOPORTE DE PROYECTOS ÁGILES



Figura 13 - Barra de navegación superior de Trello

Por otro lado, en la barra de navegación lateral, encontramos las opciones relativas al espacio de trabajo en el que nos encontremos: podemos añadir miembros o acceder a nuestros tableros (las vistas de tabla y calendario no están disponibles para el plan gratuito):

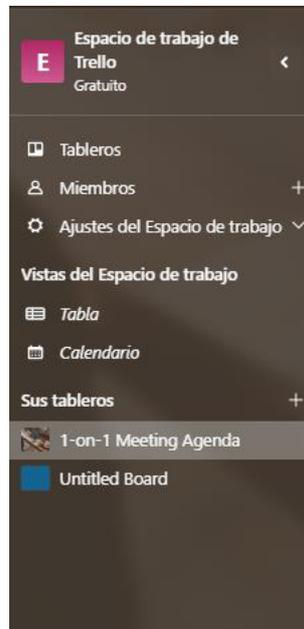


Figura 14 - Barra de navegación lateral de Trello

7.2.1.4 Revisión de funcionalidades versión gratuita

Las funcionalidades que nos ofrece Trello en su plan gratuito son las siguientes:

1. **Vista de tablero:** podremos crear tableros personalizados para la organización de las tareas. En nuestro caso, por ejemplo, hemos creado un tablero a partir de una de las plantillas que nos ofrece Trello. Además, en cada tablero podremos crear las listas que queramos para clasificar las tareas. Por defecto (sin incorporar ninguna plantilla), Trello nos incorpora 3 listas (Lista de tareas, En proceso y Hecho).

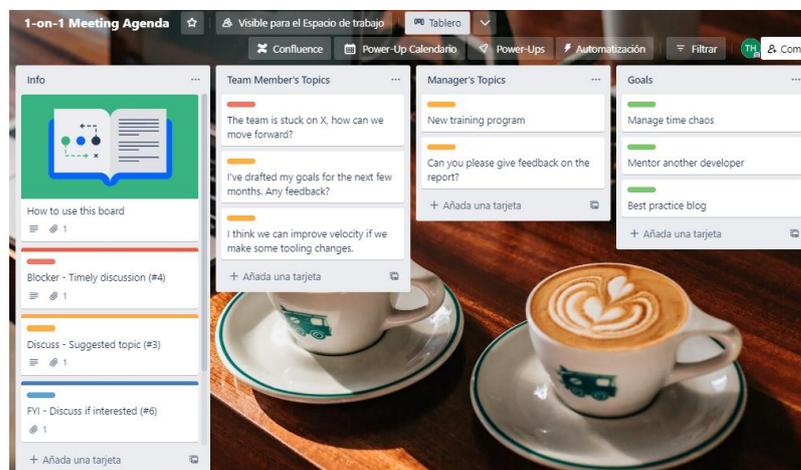


Figura 15 - Tablero en Trello

2. **Creación de tarjetas:** podemos crear tarjetas y moverlas de una lista a otra según su estado. Las opciones que nos brinda cada tarjeta son las siguientes:

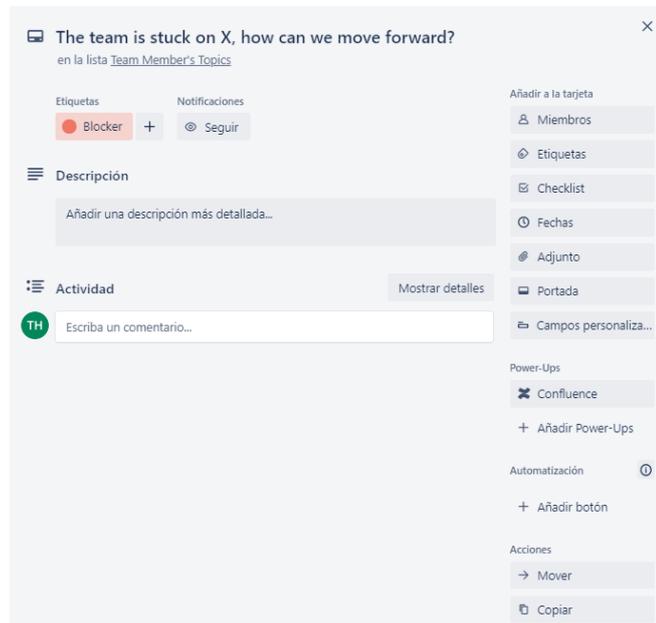


Figura 16 - Tarjeta en Trello

- Añadir descripción
 - Escribir comentarios
 - Asignación de miembros
 - Añadir etiquetas
 - Añadir checklist
 - Asignación de fechas
 - Adjuntar archivos
 - Añadir portada de imagen
3. **Filtrado de tarjetas:** podemos usar el filtrado de tarjetas, para encontrar una en particular. Los filtros pueden ser: por miembros asignados, por fecha de vencimiento o por etiquetas asociadas.

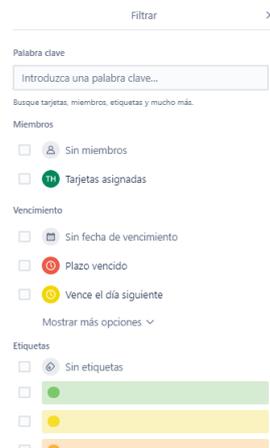


Figura 17 - Filtrado de tarjetas en Trello

4. **Automatizaciones:** Trello nos permite definir reglas de automatización de procesos sencillos, evitándonos el tiempo de realizarlos manualmente.

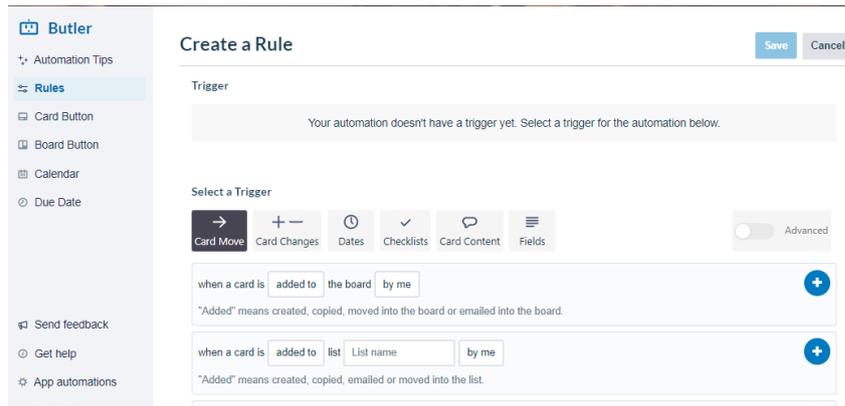


Figura 18 - Creación de regla en Trello

5. **Integraciones o Power-Ups:** podemos ampliar las funcionalidades de Trello gracias a la posibilidad de integrar aplicaciones externas en nuestro proyecto, como Google Drive, Gmail, Microsoft Teams o Slack entre muchas otras (disponemos de más de 200 aplicaciones disponibles para integrar en Trello).

Por ejemplo, nuestro tablero (gracias a la plantilla) viene precargado con el Power-Up del calendario, que permite visualizar las tarjetas en un calendario sin necesidad de disponer de la vista calendario de planes superiores:



Figura 19 - Power-Up Calendario Trello

7.2.2 Jira Software

7.2.2.1 Características principales

- **Página web oficial:** <https://www.atlassian.com/es/software/jira>
- **Compañía:** Atlassian Corporation Plc.
- **Fecha de lanzamiento:** 2002.
- **Licencia:** Software propietario (no libre).
- **Planes y precios:** Ofrecen dos tipos de soluciones: la primera como servicio SaaS y alojado en Cloud y la segunda llamada “Data Center” que es una solución autogestionada, es decir, permite a las organizaciones alojar la herramienta en su propia infraestructura.

ESTUDIO Y COMPARACIÓN DE HERRAMIENTAS DE SOPORTE DE PROYECTOS ÁGILES

Respecto a las soluciones Cloud tenemos cuatro planes: **Free** (gratuito, hasta máximo 10 usuarios), **Standard** (los precios dependen del número de usuarios, aunque el precio más alto es de 7.75 USD al mes por usuario (también se puede elegir facturación anual), máximo 35.000 usuarios), **Premium** (hasta 15.25 USD al mes por usuario (se puede elegir facturación anual), máximo 35.000 usuarios) y **Enterprise** (disponible a partir de 800 usuarios. El precio depende del número de usuarios, aunque empieza en 134.500 USD al año para 800 usuarios. El límite también es de 35.000 usuarios por sitio, pero esta vez tenemos sitios ilimitados).

Free	Standard	Premium	Enterprise
\$0 Siempre gratuito para 10 usuarios, Solo suscripción mensual Consíguela ahora	\$7.75 por usuario (estimado) \$77.50 al mes Iniciar versión de prueba	\$15.25 por usuario (estimado) \$152.50 al mes Iniciar versión de prueba	Facturación anual Pasa al ciclo de facturación anual para ver precios de Enterprise. Contacto con el departamento de ventas
<ul style="list-style-type: none"> ✓ Hasta 10 usuarios ✓ Tableros de proyecto ilimitados ✓ Backlog y hojas de ruta básicas ✓ Informes e información útil ✓ 2 GB de almacenamiento ✓ Soporte de la comunidad 	<p>Todas las funciones de Free más:</p> <ul style="list-style-type: none"> ✓ Hasta 35 000 usuarios ✓ Funciones y permisos de usuario ✓ Registros de auditoría ✓ Residencia de datos ✓ 250 GB de almacenamiento ✓ Soporte en horario comercial local 	<p>Todas las funciones de Standard más:</p> <ul style="list-style-type: none"> ✓ Hojas de ruta avanzadas ✓ Espacio aislado y canales de publicaciones ✓ Archivado de proyectos ✓ SLA con tiempo de actividad garantizado ✓ Almacenamiento ilimitado ✓ Soporte Premium ininterrumpido 	<p>Todas las funciones de Premium más:</p> <ul style="list-style-type: none"> ✓ Sitios ilimitados ✓ Controles de seguridad centralizados ✓ Suscripciones de usuario centralizadas ✓ Soporte Enterprise ininterrumpido

Figura 20 - Planes y precios Jira

Por otro lado, la solución Data Center se puede contratar a partir de 500 usuarios con un precio 42.000 USD (licencia anual).

- **Plataformas:** mientras que la opción Data Center la puedes implementar en tu propia infraestructura local, la opción Cloud sólo es accesible a través de un navegador web y de su App para móviles (IOS y Android).
- **Grado de documentación / información:** Alta (al igual que Trello, dispone de una guía de introducción sobre Jira, documentación oficial, un foro oficial de usuarios de Jira y un apartado de preguntas frecuentes. Además, Atlassian dispone de webinars grabados y en vivo sobre sus herramientas, incluida Jira).

7.2.2.2 Organización de los equipos de trabajo

En Jira los usuarios se organizan en “sitios” o también llamados “instancias”. Según la propia herramienta, para la mayoría de las empresas, de tamaño mediano o pequeño, un sitio es suficiente, sin embargo, algunas empresas optan por varios sitios para mejorar la seguridad de sus datos (al separar los datos por departamento) o por ofrecer mayor autonomía organizativa. En ese caso (de requerir más de un sitio) necesitaríamos optar por el plan más caro (Enterprise), ya que el resto solo permiten un sitio. En cada sitio, podemos crear varios proyectos diferentes.

En la figura 21 ilustramos la forma en la que se organizan los miembros de una empresa en Jira:

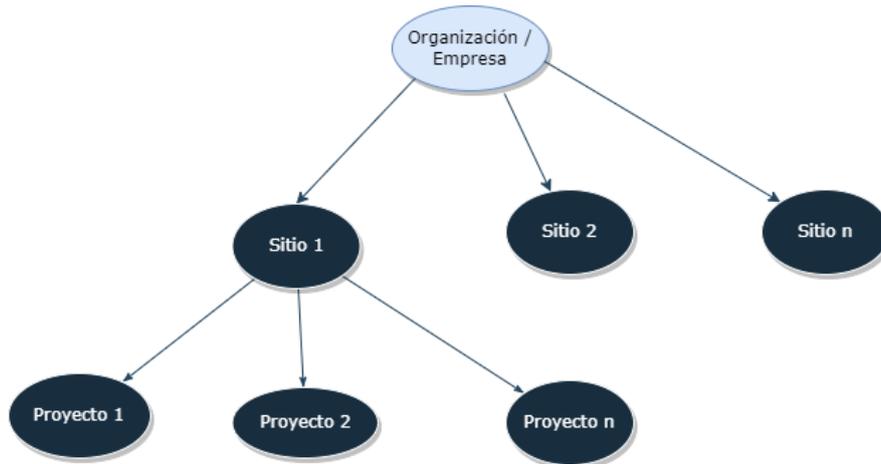


Figura 21 - Organización equipos en Jira

7.2.2.3 Interfaz del menú principal

La interfaz del menú de Jira es muy similar al de Trello (no olvidemos que ambas herramientas pertenecen a la misma compañía). Nos encontramos con una barra de navegación superior en la que podemos acceder de manera directa a incidencias asignadas que se nos hayan asignado, mis tableros, un botón para crear incidencias, y a la izquierda del menú, el botón que vimos anteriormente con Trello, para cambiar entre las aplicaciones de Atlassian:



Figura 22 - Barra navegación superior Jira

Por otro lado, tenemos la barra de navegación lateral, en la que encontramos todas las funcionalidades de Jira (por defecto, muchas de las funcionalidades del proyecto, están desactivadas, deberemos activarlas para incorporarlas a nuestro proyecto), divididas en tres bloques: planificación del proyecto (hoja de ruta, backlog, tablero, informes e incidencias), desarrollo (código y versiones) y operaciones (implementaciones y de guardia).

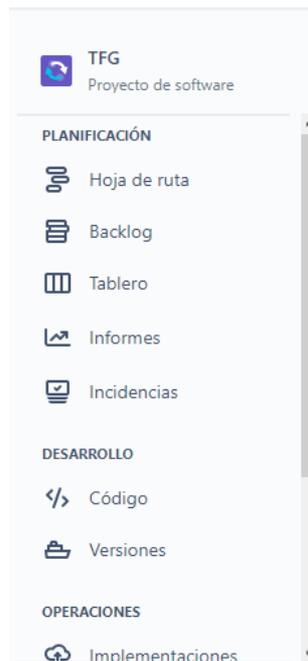


Figura 23 - Barra de navegación lateral Jira

7.2.2.4 Revisión de funcionalidades versión gratuita

En la versión gratuita de Jira disponemos de las siguientes funcionalidades:

1. **Creación de incidencias y de sprints:** en la ventana de “Backlog” de Jira, podremos crear las incidencias que deseemos en el Backlog (las incidencias en Jira pueden ser tareas, historias, errores, epics, etc. Dichos tipos deberemos añadirlos y configurarlos en nuestro proyecto según nuestras necesidades) (a cada incidencia le podemos asignar un responsable, adjuntar archivos, añadir comentarios o vincular otras incidencias, entre otros), crear Sprints y arrastrar las incidencias desde el Backlog al Sprint correspondiente. Para cada Sprint podemos establecer la fecha de inicio, de fin y el objetivo de dicho sprint.

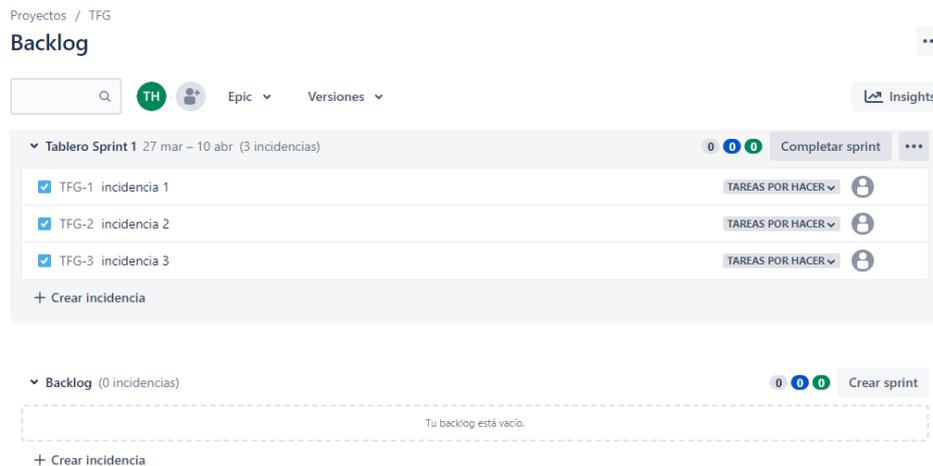


Figura 24 - Vista de "Backlog" Jira

2. **Creación de epics:** tanto desde la ventana de “Backlog” como de “Hoja de ruta” podremos crear epics y asignarles subincidencias (los epics son un tipo de incidencias que representan una gran cantidad de trabajo, el cual puede subdividirse en historias de menor tamaño). Desde la hoja de rota podemos visualizar dichos epics:

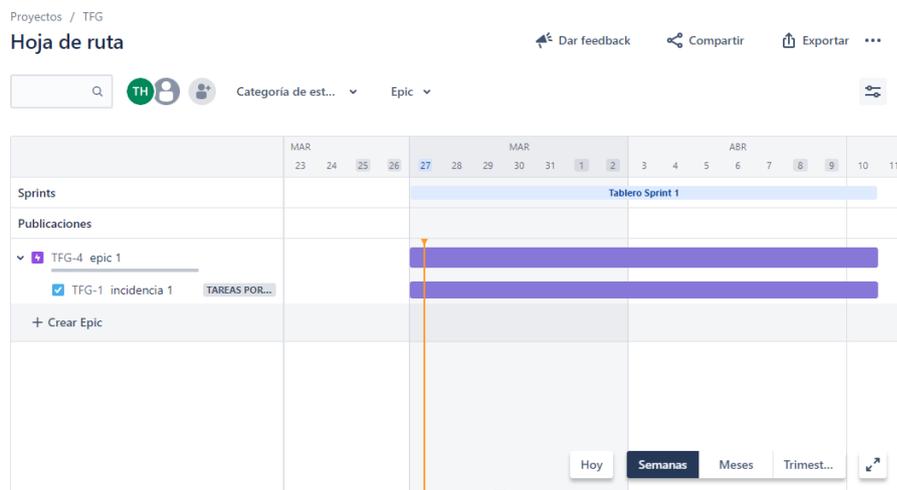


Figura 25 - Vista de "Hoja de Ruta" Jira

3. **Creación de tableros:** para cada sprint, desde la vista “Tablero”, podemos crear su respectivo tablero. Por defecto, cada tablero tiene tres columnas: por hacer, en curso y

ESTUDIO Y COMPARACIÓN DE HERRAMIENTAS DE SOPORTE DE PROYECTOS ÁGILES

listo, aunque podemos añadir las columnas que deseemos. Las incidencias del sprint podemos ir moviéndolas de una columna a otra en función de su estado.



Figura 26 - Vista tablero Jira

4. **Búsqueda avanzada de incidencias:** desde la ventana de “Incidencias” disponemos de varias opciones de filtrado avanzadas para encontrar incidencias concretas. Podemos filtrar por el tipo (epics, tareas o subtareas), persona asignada o estado.

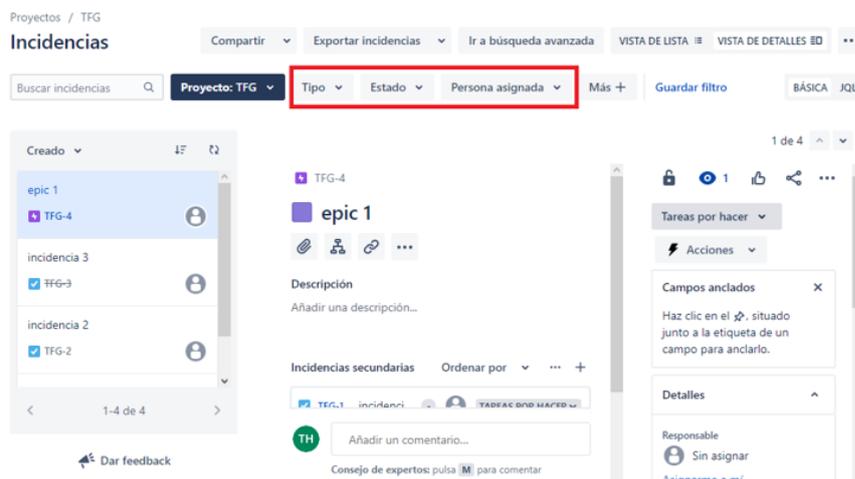


Figura 27 - Búsqueda avanzada de incidencias Jira

5. **Generación de informes:** desde la ventana “Informes” podemos generar diferentes informes con gráficos como: la gráfica de trabajo hecho, de trabajo pendiente, de velocidad, de flujo acumulado, de duración del ciclo o de frecuencia de implementación:



Figura 28 - Gráficas disponibles informes Jira

6. **Creación de versiones:** permite a los equipos organizar y realizar un seguimiento del progreso de su trabajo gracias a la creación de versiones.
7. **Gestión de implementaciones:** permite a los equipos planificar y realizar un seguimiento de las implementaciones de software realizadas.
8. **Establecer programaciones de guardia:** permite crear programaciones de guardia para mantener el control cuando algo salga mal.
9. **Automatizaciones:** al igual que en Trello, podemos definir reglas sencillas para automatizar procesos, como, por ejemplo, cerrar un epic automáticamente cuando todas las historias estén completadas.

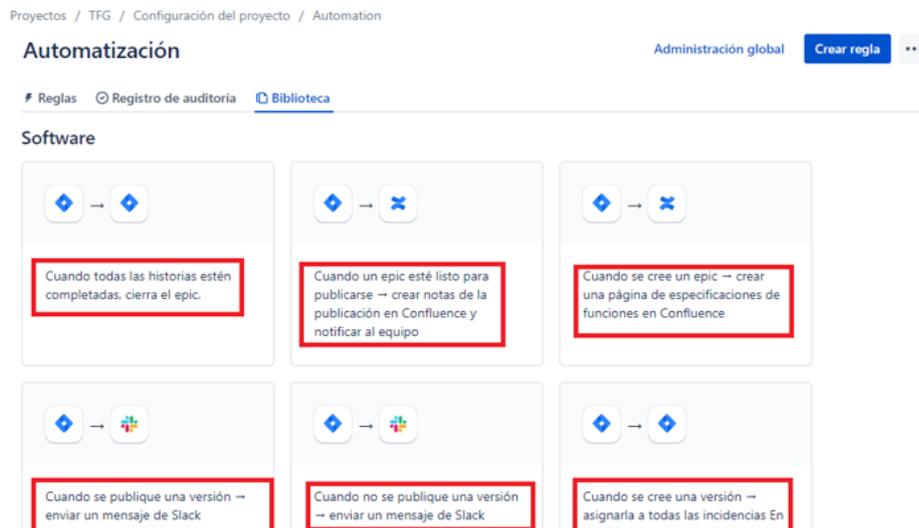


Figura 29 - Reglas de automatización disponibles Jira

10. **Integraciones:** podemos conectar Jira con numerosas aplicaciones externas (más de 1000) disponibles en Atlassian Marketplace.

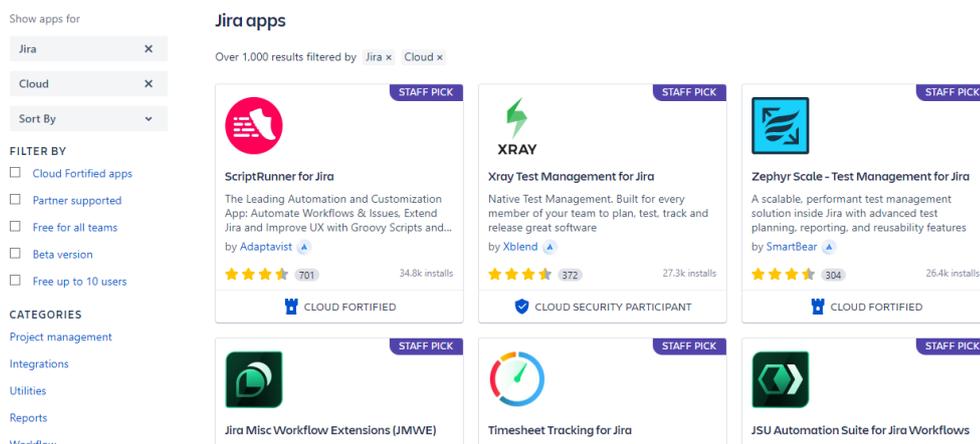


Figura 30 - Integraciones Jira

7.2.3 Asana

7.2.3.1 Características principales

- **Página web oficial:** <https://asana.com/es>
- **Compañía:** Asana Inc.
- **Fecha de lanzamiento:** 2011.
- **Licencia:** Software propietario (no libre).
- **Planes y precios:** hay disponibles un total de cuatro planes (todos en la nube): **Basic** (plan gratuito, hasta 15 usuarios), **Premium** (13,49 € por usuario al mes si la facturación es mensual o 10,99 € al mes por usuario si la facturación es anual), **Business** (30,49 € por usuario al mes con facturación mensual o 24,99 € al mes por usuario con facturación anual) y **Enterprise** (es necesario contactar con Asana para solicitar presupuesto).



Figura 31 - Planes y precios Asana

- **Plataformas:** todos los planes son accesibles a través de su versión web o de navegador, a través de su aplicación de escritorio para MAC y Windows o a través de su App para móviles (compatible con IOS y Android).
- **Grado de documentación / información:** Alta (en la página oficial de Asana podemos encontrar una guía de uso básica de la herramienta para inicializarnos en ella, videotutoriales, webinars en vivo, cursos a medida según tus necesidades, ejemplos de proyectos en Asana o el foro oficial de la comunidad).

7.2.3.2 Organización de los equipos de trabajo

En Asana, los usuarios están organizados en **equipos**. Los equipos, según Asana, son subgrupos de personas de la organización que colaboran conjuntamente en proyectos. [16]

En una organización, lo normal es convertir un espacio de trabajo o equipo en una **organización** (esto se puede realizar siguiendo los pasos que indica Asana para ello). Las organizaciones conectan a todas las personas de una empresa que usan Asana, en un solo espacio de trabajo (en base al dominio de email de la compañía). [16]

Una vez que tengamos la organización creada, podemos crear diferentes equipos dentro de esta, y de esta manera, todos los empleados que estén dentro de la organización podrán visualizar y unirse a los equipos que formen parte de ella. Dentro de cada equipo, podemos crear diferentes proyectos. El número de proyectos que podemos crear en cada equipo es ilimitado en todos los planes.

En la figura 32 ilustramos la forma en la que se organizan los miembros de una empresa en Asana:

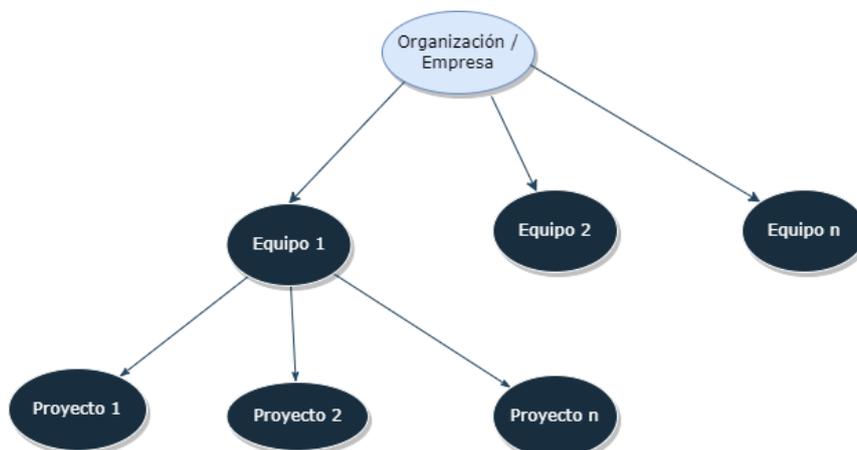


Figura 32 - Organización equipos en Asana

7.2.3.3 Interfaz del menú principal

En la barra de navegación superior de Asana encontramos una opción directa para crear tareas, un nuevo proyecto, mensajes a personas, proyectos o equipos, e invitaciones de nuevos miembros a nuestro equipo de trabajo:

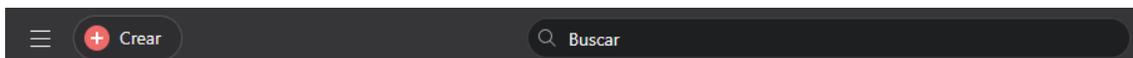


Figura 33 - Barra de navegación superior Asana

En la barra de navegación lateral encontramos las vistas que nos proporciona Asana: página de inicio, mis tareas, bandeja de entrada, informes, portafolios, objetivos, lista de proyectos y espacio de trabajo):

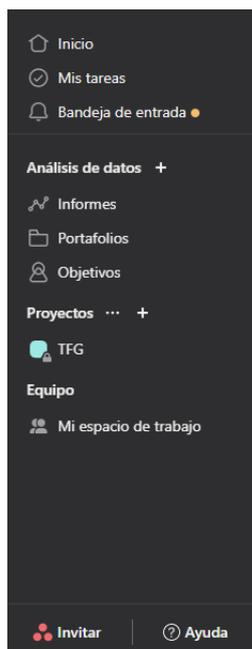


Figura 34 - Barra de navegación lateral Asana

7.2.3.4 Revisión de funcionalidades versión gratuita

En la versión gratuita de Asana podemos encontrar las siguientes funcionalidades:

1. **Creación de tareas:** podemos crear tareas para cada proyecto. En cada tarea podemos incluir: una descripción, un responsable, archivos o vídeos adjuntos, fecha de entrega o mencionar a otro miembro del equipo.

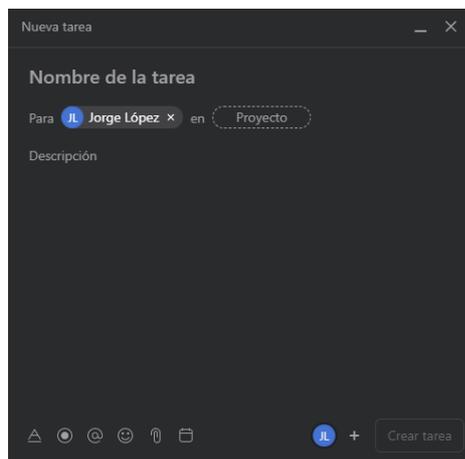


Figura 35 - Creación de tarea Asana

2. **Vistas de lista, calendario y tablero:** podemos visualizar nuestras tareas en tres vistas diferentes: en lista, en tablero o en un calendario:

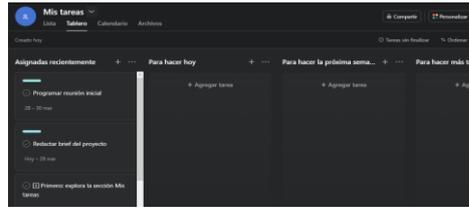


Figura 36 – Vista tablero Asana

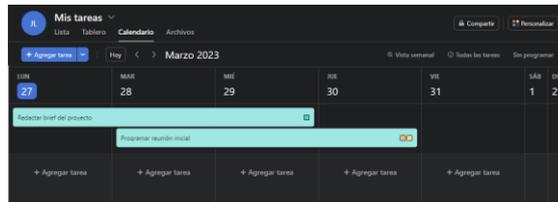


Figura 37 - Vista calendario Asana

3. **Envío de mensajes:** disponemos de una bandeja de entrada desde la que podemos enviar mensajes a personas concretas, a todo nuestro equipo de trabajo o a todas las personas de nuestro proyecto.
4. **Generación de informes:** podemos generar informes y agregar a estos los gráficos que deseemos (permite agregar gráficos personalizados o de entre los que proporciona Asana por defecto):



Figura 38 - Gráficos informes Asana

5. **Automatizaciones (versiones superiores):** aunque las automatizaciones no están disponibles en la versión gratuita de Asana, sí que considero que es importante que veamos su funcionamiento para compararlo con el resto de herramientas. La creación de los flujos de trabajo en Asana funciona de la siguiente manera: disponemos de varias secciones a las cuales les podemos añadir un disparador para que en función de este, las tareas se muevan a la sección de manera automática (un ejemplo de disparador puede ser asignar una prioridad alta a una tarea), posteriormente, en cada sección deberemos definir las reglas. Las reglas están conformadas por otro disparador (por defecto, es tarea agregada a dicha sección) y por una acción que será ejecutada cuando se cumpla este último disparador (un ejemplo de acción puede ser actualizar el estado de finalización de esa tarea). A estos flujos de trabajo podemos conectar aplicaciones externas como Gmail, Slack o Microsoft Teams.



Figura 39 - Secciones flujos de trabajo Asana



Figura 40 - Reglas flujos de trabajo Asana

6. **Definición de objetivos:** permite definir objetivos de empresa en base a lo conseguido en los proyectos:

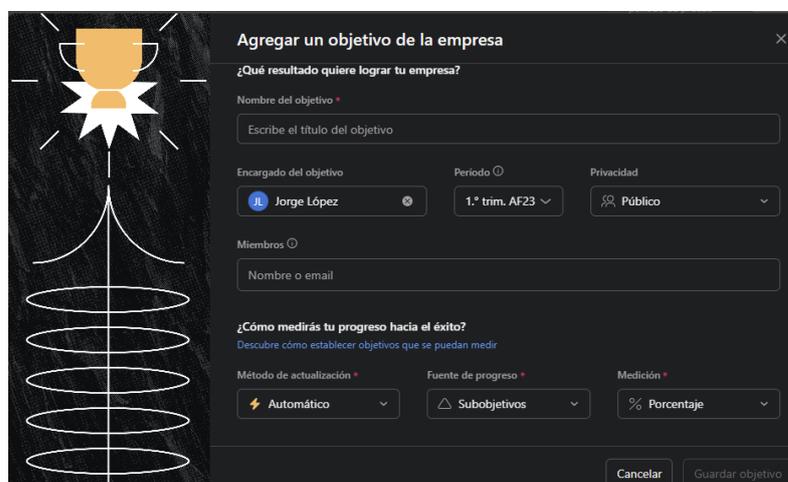


Figura 41 - Creación objetivos Asana

7. **Creación de portafolios:** podemos crear portafolios para reunir información de diferentes proyectos que tengamos activos.
8. **Integraciones:** podemos integrar aplicaciones externas en Asana.

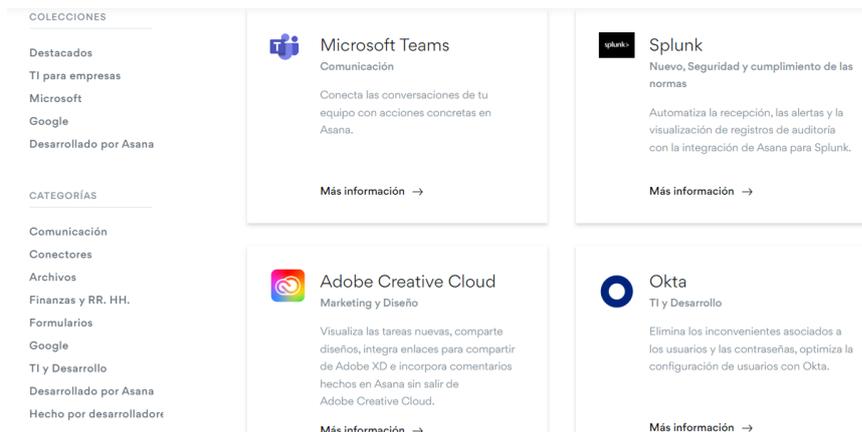


Figura 42 - Integraciones Asana

7.2.4 Wrike

7.2.4.1 Características principales

- **Página web oficial:** <https://www.wrike.com/es/>
- **Compañía:** Wrike Inc. (propiedad de Citrix Systems Inc.).
- **Fecha de lanzamiento:** 2006.
- **Licencia:** Software propietario (no libre).
- **Planes y precios:** Wrike dispone de cinco planes diferentes (no dispone de soluciones en local, todas son en la nube): **Free** (plan gratuito, sin límite de usuarios), **Team** (9.80 USD por usuario al mes, disponible de 2 a 25 usuarios), **Business** (24.80 USD por usuario al mes, disponible de 5 a 200 usuarios), **Enterprise** (es necesario consultar precio, disponible a partir de 5 usuarios) y **Pinnacle** (al igual que Enterprise es necesario consultar precios y está disponible a partir de 5 usuarios).

Free	Team	Business	Enterprise
Para los equipos que están empezando	Para los equipos en rápido crecimiento	Para todos los equipos de una organización	Para equipos grandes
Gestión de tareas centralizada	Permite gestionar proyectos de forma colaborativa y automatizar los procesos	Plataforma personalizada para cualquier equipo o caso de uso	Escalabilidad y seguridad de nivel empresarial
USD 0 <small>usuario/mes</small>	USD 9.80 <small>usuario/mes</small>	USD 24.80 <small>usuario/mes</small>	
Suscríbete ya	Iniciar prueba gratuita	Iniciar prueba gratuita	Contáctanos Iniciar prueba gratuita

Figura 43 - Planes y precios Wrike

- **Plataformas:** Wrike es accesible tanto desde un navegador web, como desde su aplicación de escritorio para MAC y Windows, como desde su App para móviles (Android e IOS).
- **Grado de documentación / información:** Alta (ofrecen webinars mensuales formativos, videotutoriales, portal de posts para la colaboración entre usuarios de la herramienta y documentación de uso oficial).

7.2.4.2 Organización de los equipos de trabajo

Los usuarios en Wrike en **espacios**. Los espacios ocupan el nivel más alto de la jerarquía (en cada organización podemos tener varios espacios). Dentro de cada espacio, podemos crear varios **proyectos** y **carpetas**. La diferencia entre las carpetas y los proyectos es que las primeras no tienen atributos de los que se pueda realizar un seguimiento (a diferencia de los proyectos). Las carpetas pueden utilizarse, por ejemplo, para agrupar tareas relacionadas.

En la figura 44 ilustramos la forma en la que se organizan los miembros de una compañía en Wrike:

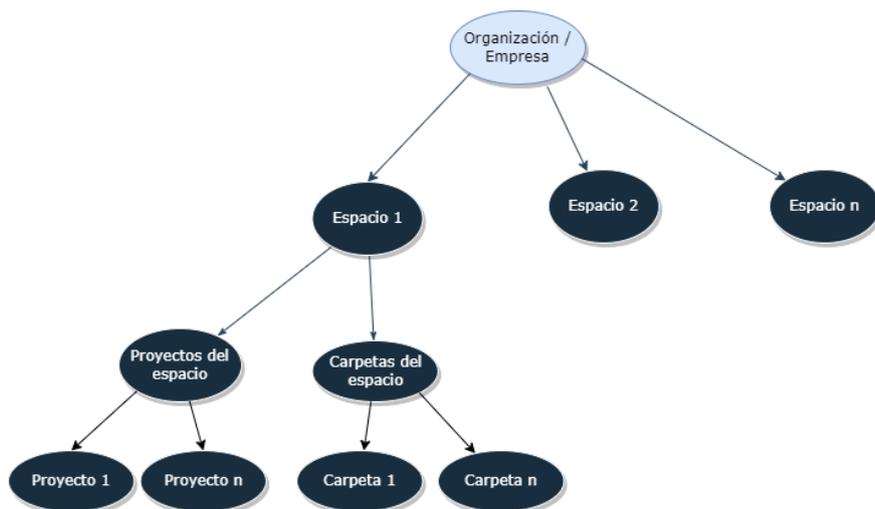


Figura 44 - Organización equipos en Wrike

7.2.4.3 Interfaz del menú principal

El menú principal de Wrike se organiza también en torno a una barra de navegación lateral y a otra superior.

En la barra de navegación superior nos encontramos con una opción de búsqueda rápida de elementos como tareas, proyectos o archivos. También tenemos un botón para crear tareas, proyectos, carpetas, espacios o invitar a personas a nuestro equipo de trabajo.

Inicio



Figura 45 - Barra de navegación superior Wrike

Por otro lado, en la barra de navegación lateral encontramos las siguientes vistas: Menú principal (donde tenemos accesos directos a la bandeja de entrada, a nuestros espacios o a elementos recientes), Bandeja de entrada (donde podemos consultar notificaciones que nos mantendrán al día en nuestras tareas y proyectos), paneles de control (funcionalidad premium para crear vistas del estado del proyecto) y mis tareas (donde puedo consultar y filtrar entre todas las tareas que me hayan sido asignadas). Además, puedo cambiar entre mis espacios y entre mis proyectos o carpetas.

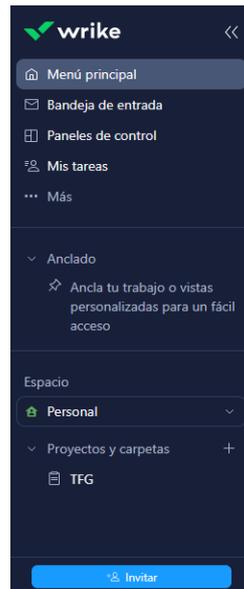


Figura 46 - Barra de navegación lateral Wrike

7.2.4.4 Revisión de funcionalidades versión gratuita

1. **Creación de tareas:** podemos crear tareas con los siguientes campos: descripción, personas asignadas, fecha de inicio y vencimiento, archivos adjuntos y subelementos. Además, podemos añadir comentarios a cada tarea.

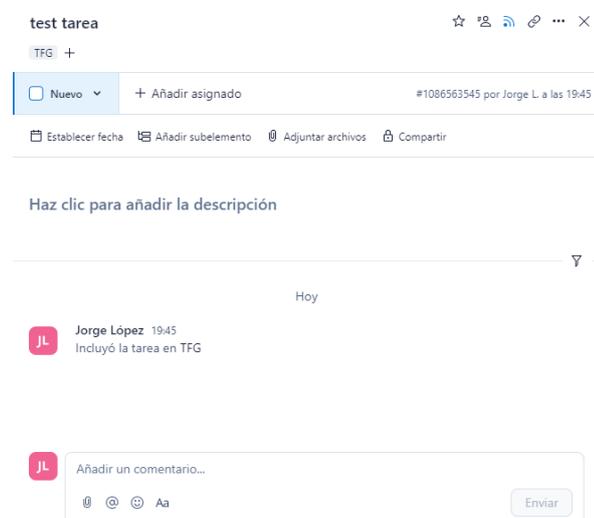


Figura 47 - Creación de tarea Wrike

2. **Filtrar tareas:** podemos filtrar tareas por diferentes campos (por estado, importancia, fecha de inicio, fecha de última modificación o por autor, entre otros).
3. **Vista de tabla, tablero y lista:** podemos visualizar las tareas en tres vistas diferentes: lista, tablero scrum o tabla.

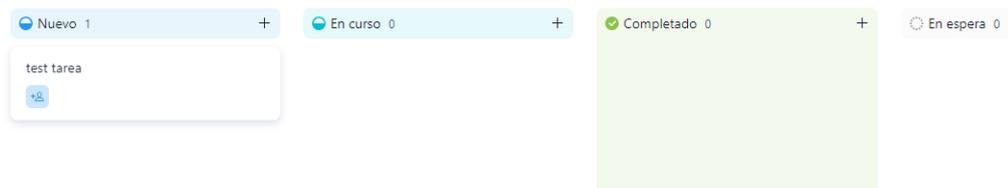


Figura 48 - Vista tablero Wrike

4. **Bandeja de entrada:** podemos consultar de manera fácil las menciones de otros miembros del equipo de trabajo o las notificaciones cuando se nos asigna una tarea o cuando nos mencionan.
5. **Integraciones:** podemos integrar Wrike con otras aplicaciones como Gmail, Outlook, Microsoft Teams, Zoom o Slack. Sin embargo, la oferta de aplicaciones que podemos integrar con el plan Free es escasa (sólo las que vemos en la figura 49).

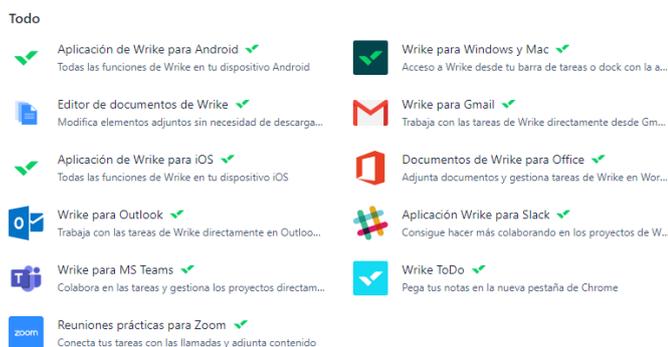


Figura 49 - Integraciones plan Free Wrike

7.2.5 iceScrum

7.2.5.1 Características principales

- **Página web oficial:** <https://www.icescrum.com/es/>
- **Compañía:** Kagilum.
- **Fecha de lanzamiento:** 2010.
- **Licencia:** GNU/GPL (código libre).
- **Planes y precios:** iceScrum pone a nuestra disposición dos opciones: contratar la herramienta en su opción **Cloud** o en su opción **On-premise**, en la que podemos integrar la aplicación en nuestros propios servidores e infraestructura. Si nos decantamos por la opción en la nube, disponemos de 4 planes: **Free** (gratuito, solo podemos crear/estar en un equipo de trabajo y solo podemos añadir hasta 3 usuarios a nuestro equipo), **Small Team** (8.90 € al mes o (u 89 € con facturación anual), nos permite añadir 7 usuarios por equipo, límite de 1 equipo de trabajo), **Teams** (el precio depende del límite de número de equipos que queramos (límite de 5 equipos), con 1 equipo el precio es de 39.90 € al mes (o 399 € con facturación anual), sin límite de usuarios por equipo) y **Business** (349 € al mes (o 3490 € con facturación anual), equipos y usuarios ilimitados).

ESTUDIO Y COMPARACIÓN DE HERRAMIENTAS DE SOPORTE DE PROYECTOS ÁGILES

	Free 0€	Small Team 8.90€/mes	Teams ★ Popular 39.90€/mes	Business 349.00€/mes
Cloud				
On-premise		Prueba gratuita Comprar ahora	Prueba gratuita Comprar ahora	Prueba gratuita Comprar ahora
Alojado en icescrum.com				
Equipos	1	1	- 1 +	Ilimitado
Usuarios	3	7	Ilimitado	Ilimitado
Proyectos privados	1	1	Ilimitado	Ilimitado
Apps & integraciones	1	3	Ilimitado	Ilimitado
Soporte	Foro	Foro	Email	Email / Teléfono / Remoto
Soporte SAFe & LeSS	✗	✗	✗	1
Infraestructura privada	✗	✗	✗	✓

Figura 50 - Planes y precios versiones Cloud iceScrum

Por otra parte, si elegimos la opción On-premise tendremos también 4 planes disponibles: **Core** (gratuito, sin límite de usuarios ni equipos), **Starter** (990 € al año (o 5990 € si elegimos comprarlo para siempre), límite de 1 equipo y usuarios ilimitados), **Enterprise** (2990 € al año (o 18000 € para siempre), equipos y usuarios ilimitados) y **Corporate** (9990 € al año (o 90000 € para siempre) sin límite de equipos ni usuarios).

	Core 0€	Starter 990€/año	Enterprise ★ Popular 2990€/año	Corporate 9990€/año
Cloud				
On-premise	Descargar	Prueba gratuita Comprar ahora	Prueba gratuita Comprar ahora	Prueba gratuita Comprar ahora
Alojado en su propio servidor				
Equipos	Ilimitado	1	Ilimitado	Ilimitado
Usuarios	Ilimitado	Ilimitado	Ilimitado	Ilimitado
Proyectos	Ilimitado	Ilimitado	Ilimitado	Ilimitado
Apps & integraciones	✗	5	Ilimitado	Ilimitado
Soporte	Foro	Email	Email / Teléfono / Remoto	Email / Teléfono / Remoto
Soporte SAFe & LeSS	✗	✗	1	Ilimitado
Infraestructura privada	✓	✓	✓	✓

Figura 51 - Planes y precios versiones On-premise iceScrum

- **Plataformas:** la versión On-Premise de iceScrum solo permite acceder a la herramienta desde una aplicación instalada en nuestro propio dispositivo. Por otra parte, la versión cloud es accesible desde un navegador web o desde un dispositivo móvil gracias a su App “iceScrum Companion”.
- **Grado de documentación / información:** Media (dispone de documentación oficial y de un foro de usuarios).

7.2.5.2 Organización de los equipos de trabajo

En iceScrum los usuarios se organizan en **equipos**. En cada organización podemos tener varios equipos, sin embargo, como ya hemos visto, si queremos tener más de un equipo tendremos que optar por los dos planes más avanzados. Dentro de cada equipo podremos crear varios **proyectos**.

En la figura 52 ilustramos la forma en la que se organizan los miembros de una empresa en iceScrum:

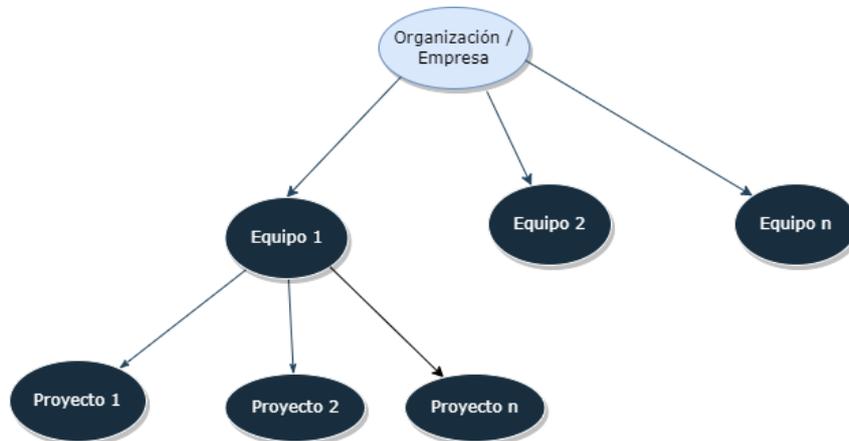


Figura 52 - Organización equipos en iceScrum

7.2.5.3 Interfaz del menú principal

Nada más entrar en iceScrum nos encontramos con la siguiente pantalla en la que podemos visualizar nuestros proyectos y diferentes “Widgets” (como “mis tareas”).

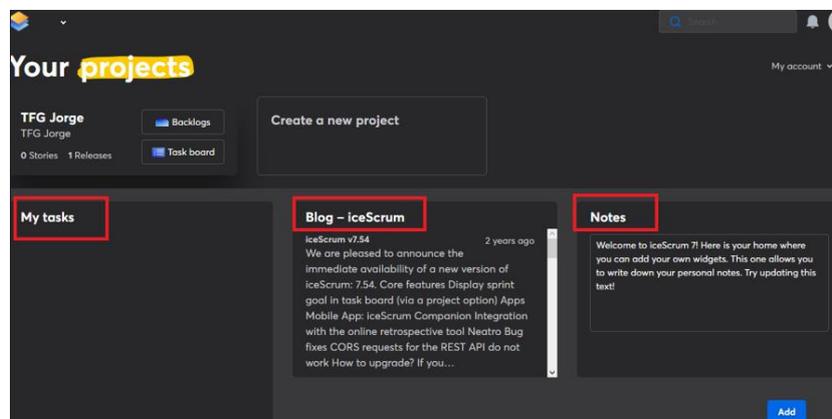


Figura 53 - Pantalla de inicio iceScrum

Una vez dentro de nuestro proyecto, nos encontramos con una barra de navegación superior en la que tenemos todas las vistas de la herramienta: dashboard (vista general del proyecto), backlogs, planning (sprints del proyecto), task board (tareas de cada sprint) y features (creación de características):

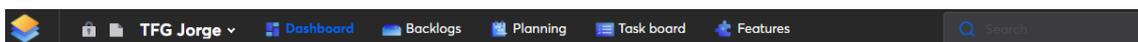


Figura 54 - Barra de navegación superior iceScrum

7.2.5.4 Revisión de funcionalidades versión gratuita

Con la versión gratuita de iceScrum podemos acceder a las siguientes funcionalidades:

1. **Vista de dashboard:** disponemos de una vista dashboard del proyecto, desde la cual podemos visualizar aspectos generales de este, como el número de historias, entregas, sprints (y sus fechas), archivos del proyecto o planificar reuniones.

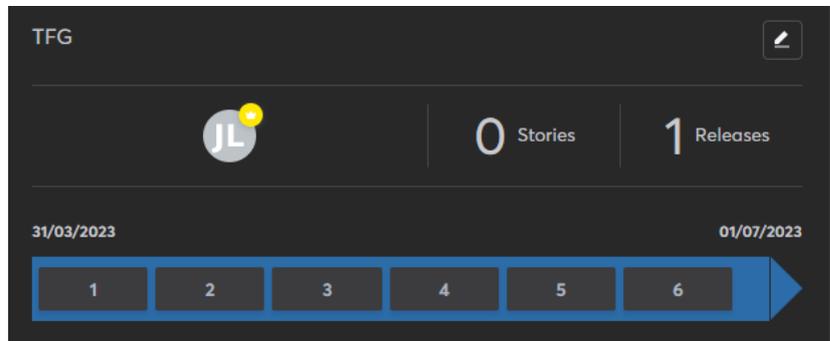


Figura 55 - Vista "Dashboard" iceScrum

2. **Indicadores / Informes:** dentro de la vista de dashboard, podemos generar diferentes gráficos asociados a indicadores como la velocidad, burndown (cantidad de trabajo pendiente) o burnup (cantidad de producto entregado), entre otros.

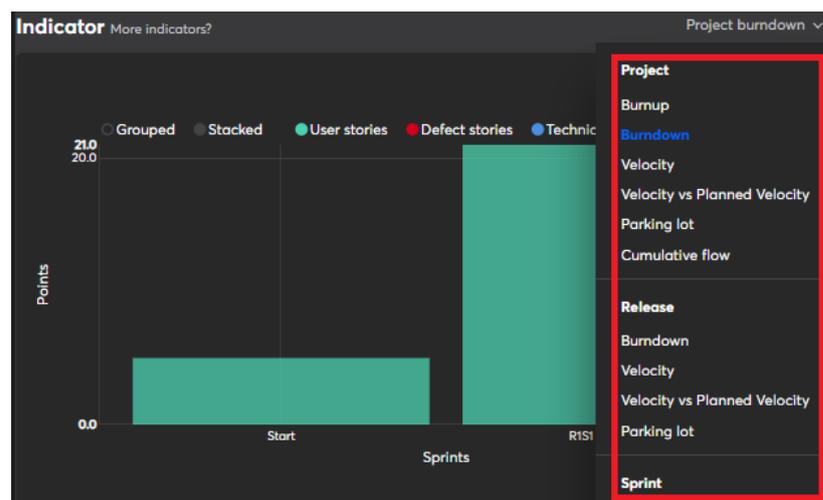


Figura 56 - Generación gráficos iceScrum

3. **Creación de historias:** podemos crear historias (según iceScrum: objetos de la pila del Producto que aportan valor a los usuarios, stakeholders y ocasionalmente al equipo mismo). [20]

Cada historia tiene un nombre, una descripción, un tipo (puede ser historia de usuario: aporta valor a los stakeholders o a los usuarios, con la creación de una nueva funcionalidad o mejora del producto, una historia de defecto: elimina valor y debe ser corregido o una historia o una historia técnica: aporta valor al equipo, por ejemplo, mejorando la calidad técnica o reduciendo el riesgo), un estado (puede ser sugerida, aceptada, estimada, planeada, en progreso, en revisión, terminada o congelada), archivos adjuntos, comentarios, características asociadas (puede no tener ninguna), criterios de aceptación (reglas de negocio que se deben cumplir para validar la historia una vez haya sido desarrollada) y tareas asociadas, entre otros campos. [20]

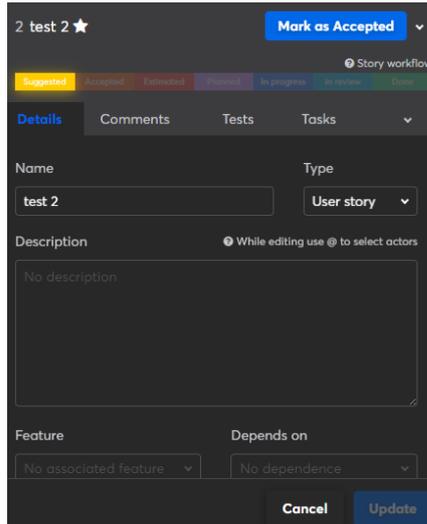


Figura 57 - Creación historia iceScrum

4. **Creación de tareas:** dentro de cada historia podemos crear las tareas que creamos oportunas. Cada tarea está compuesta por un nombre, una descripción y el tiempo para realizarla.

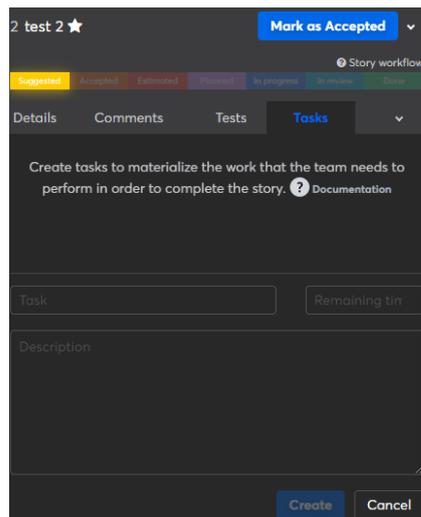


Figura 58 - Creación tarea iceScrum

5. **Vista de backlogs:** en la vista de backlogs podemos visualizar las historias según su estado: si están pendientes de ser aceptadas, es decir, están sugeridas, estarán en “sandbox”, si ya están aceptadas estarán en el “backlog”, y una vez haya sido estimado su esfuerzo, haya sido asociada a un sprint (planificada) y haya sido revisada, será marcada como “Done” (terminada).

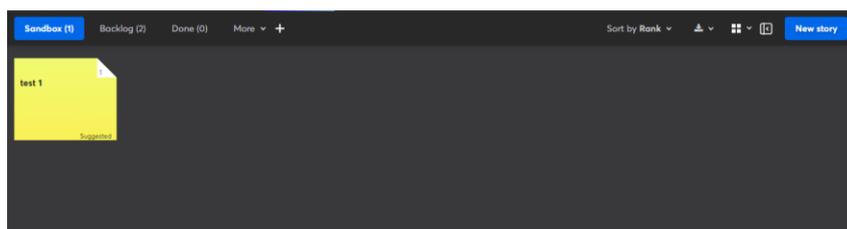


Figura 59 - Vista de "Backlogs" iceScrum

6. **Creación y vista de sprints:** desde la vista de Planning, podemos visualizar todos los sprints creados para la entrega actual. Podemos ver las historias de cada sprint y planificar los siguientes (añadir historias que hayan sido aceptadas y estimadas).

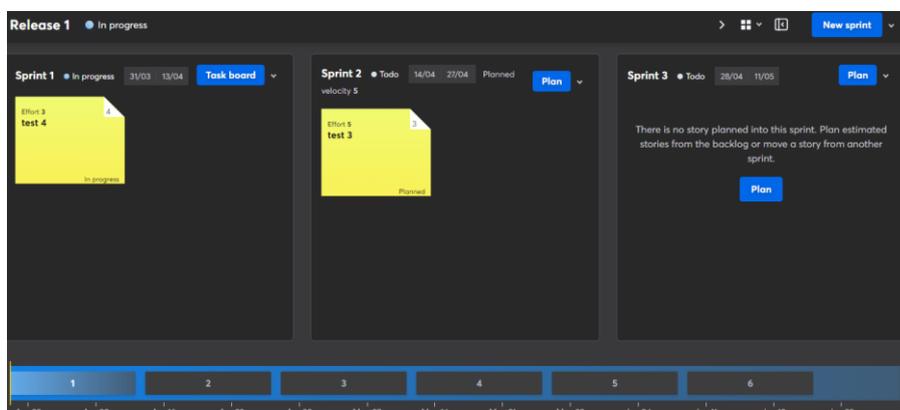


Figura 60 - Vista de sprints iceScrum

7. **Vista de tablero:** podemos visualizar el tablero de tareas de cada sprint en varias columnas según estén en progreso o terminadas.

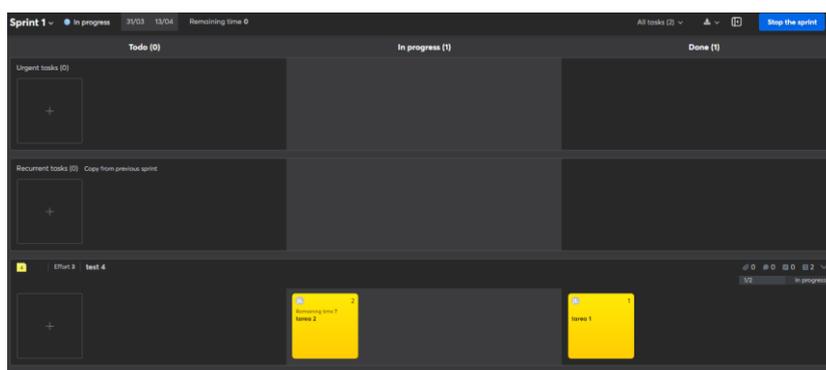


Figura 61 - Vista de tablero iceScrum

8. **Creación de características:** podemos crear características, que en iceScrum son partes del producto que aportan un valor significativo a sus usuarios y que son demasiado grandes como para que puedan ser trabajadas directamente, es por ello que cada característica puede estar asociada a varias historias. [20]

Cada característica puede tener asociada: un nombre, una descripción, un estado (en progreso, terminada), un tipo (funcional o habilitador), comentarios, archivos adjuntos, historias asociadas o un valor de negocio (representado con un número del 0 al 99), entre otros campos.

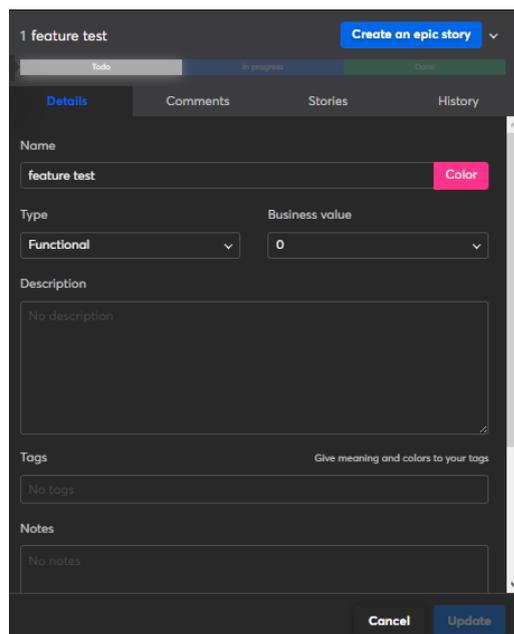


Figura 62 - Creación característica iceScrum

9. **Integraciones:** podemos integrar iceScrum con aplicaciones externas como Discord, Dropbox, Github o Google Drive. La versión gratuita está limitada a 1 sola integración.

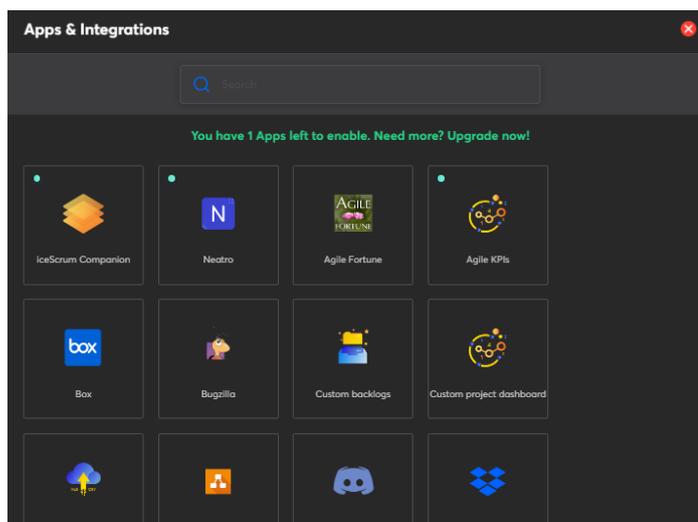


Figura 63 - Integraciones iceScrum

7.2.6 ClickUp

7.2.6.1 Características principales

- **Página web oficial:** <https://clickup.com>
- **Compañía:** ClickUp LLC.
- **Fecha de lanzamiento:** 2017.
- **Licencia:** Software propietario (no libre).
- **Planes y precios:** ClickUp ofrece cinco planes (todos en la nube, no ofrece soluciones en local): **Gratuito** (plan sin coste alguno), **Ilimitado** (5 USD por usuario al mes con facturación anual o bien 9 USD por usuario al mes con facturación mensual), **Business**

ESTUDIO Y COMPARACIÓN DE HERRAMIENTAS DE SOPORTE DE PROYECTOS ÁGILES

(12 USD por usuario al mes con facturación anual o bien 19 USD por usuario al mes con facturación mensual), **Business plus** (19 USD por usuario al mes con facturación anual o bien 29 USD por usuario al mes con facturación mensual) y **Enterprise** (contactar directamente con ellos para obtener presupuesto).

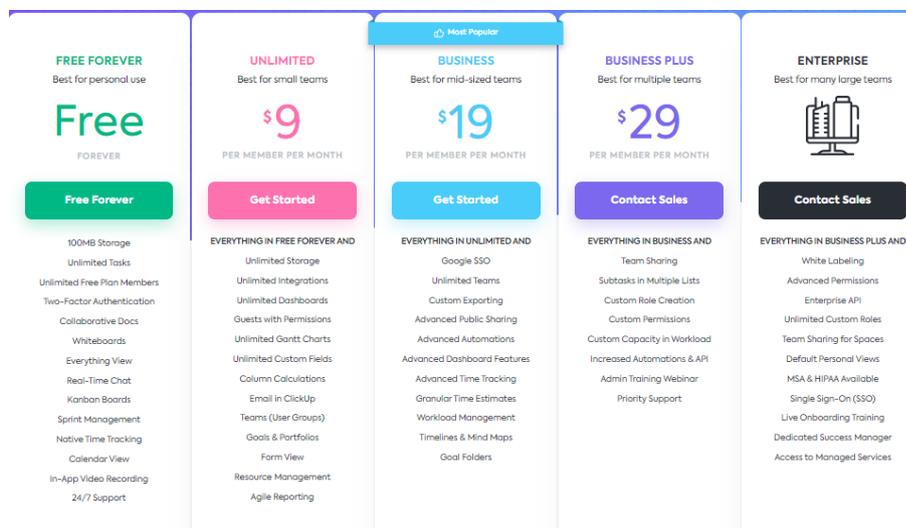


Figura 64 - Planes y precios de ClickUp

- **Plataformas:** ClickUp dispone de una versión de escritorio que puede ser descargada tanto para MAC, Linux y Windows y una aplicación para móviles (IOS y Android).
- **Grado de documentación / información:** Alta (disponen de videotutoriales, webinars grabados o en directo, cursos para obtener certificaciones y documentación oficial).

7.2.6.2 Organización de los equipos de trabajo

En ClickUp, el nivel más alto de la jerarquía de directorios es el “**Workspace**” o espacio de trabajo. En la mayoría de las situaciones, cada empresa tendrá un solo espacio de trabajo y en cada espacio de trabajo podremos crear varios **espacios** para cada uno de los equipos de trabajo (los espacios pueden ser creados de manera pública para todos los miembros del Workspace o de manera privada para nosotros mismos o personas concretas del espacio de trabajo).

Dentro de cada espacio, podemos crear **listas**. Las listas serían el equivalente de los proyectos en otras herramientas. Además, esas listas pueden estar agrupadas también en **carpetas** dentro de dicho espacio (las carpetas pueden crearse privadas, públicas para todos los miembros del espacio e incluso compartirlas mediante enlace).

En la figura 65 ilustramos la forma en la que se organizan los miembros de una empresa en ClickUp:

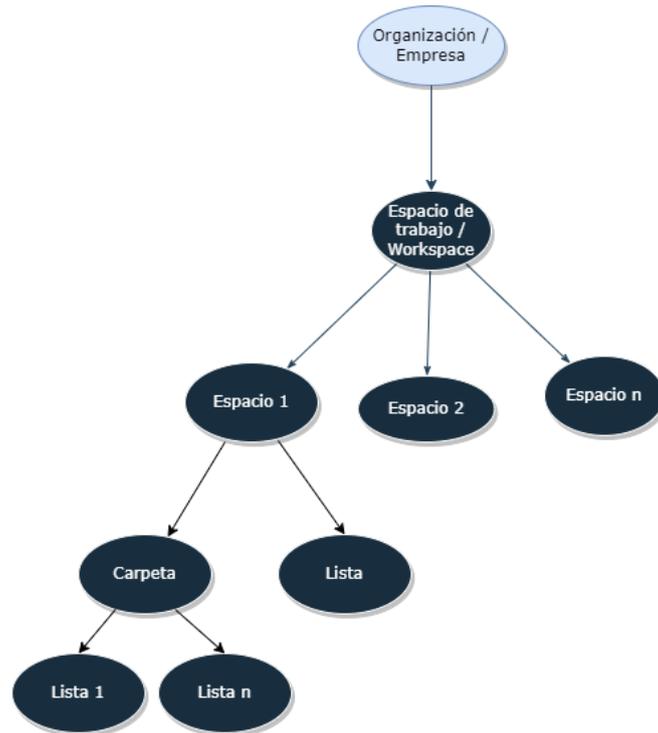


Figura 65 - Organización equipos en ClickUp

7.2.6.3 Interfaz del menú principal

El menú principal de ClickUp está conformado por una barra de navegación lateral. En dicha barra encontramos las siguientes vistas: inicio (donde podemos consultar nuestras tareas asignadas o nuestro calendario), notificaciones, pulso (en el que podemos ver el número de personas conectadas en nuestro Workspace y en qué están trabajando), metas (en el que podemos establecer diferentes metas objetivo tanto privadas como para varios miembros de nuestro Workspace), lista de espacios (donde podemos cambiar entre los diferentes espacios de nuestro Workspace), paneles (donde podemos crear paneles privados exclusivos para nosotros), y documentos (donde podemos consultar los documentos del espacio de trabajo y su localización):

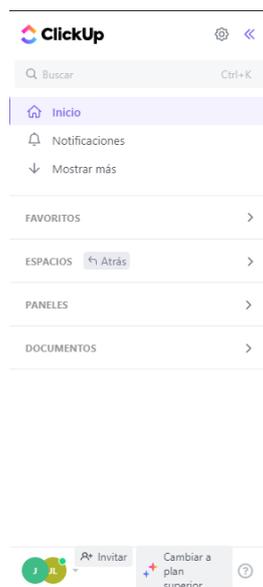


Figura 66 - Barra de navegación superior ClickUp

7.2.6.4 Revisión de funcionalidades versión gratuita

La versión gratuita de ClickUp incluye las siguientes funcionalidades:

1. **Creación de sprints:** ClickUp nos brinda la posibilidad de en vez de crear carpetas normales dentro de los sitios, crear carpetas de sprint, de esta forma podemos incorporar la funcionalidad de los sprints de scrum en ClickUp. En las carpetas de sprint, al igual que en las carpetas normales, podemos crear las listas y subcarpetas que deseemos.

Ajustes predeterminados para sprints

Mira este [breve vídeo](#) para saber más sobre la ClickApp Sprints.

Nombre del sprint (predeterminado)
{INDEX} Sprint

Duración del sprint: 2 semanas | Formato de fecha: M/D

Zona horaria del sprint: (GMT+02:00) Europe/Madrid (CEST)

Día de inicio del sprint (predeterminado): Lunes | Hora de inicio: 12:00 am

Medida de esfuerzo: Puntos de sprint

Figura 67 - Creación de sprint ClickUp

2. **Creación de tareas:** dentro de cada lista podemos crear tareas. Cada tarea en ClickUp tiene los siguientes campos: nombre, descripción, personas asignadas, subtareas, archivos adjuntos, prioridad (baja, normal, alta o urgente), fechas de inicio y límite, y dependencias (pueden ser tareas de espera: deben completarse antes de dicha tarea, de bloqueo: no pueden iniciarse hasta que se haya terminado dicha actual, o tareas relacionadas: están relacionadas entre sí, pero son independientes).

tarea 1

En List Para [person icon] 3/3

Write something or type '/' for commands

Mostrar 1 campo vacío + Añadir o modificar campos

+ Añadir subtarea + Añadir lista de control

Adjuntos Añadir

Arrastra y suelta archivos para adjuntarlos a explorar

Crear tarea

ctrl + Intro

Figura 68 - Creación de tarea ClickUp

3. **Vistas de lista, tablero y calendario:** dentro de cada lista, podemos visualizar las tareas según el estado en el que se encuentren de tres formas diferentes: en forma de lista, de tablero o de calendario. Por defecto, tenemos tres estados diferentes: pendiente, en progreso o terminado, aunque en las vistas de lista y tablero podemos añadir nuevos estados.

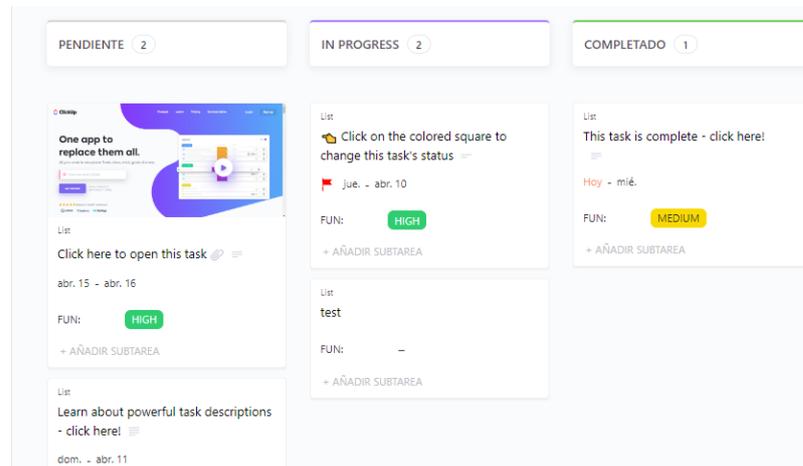


Figura 69 - Vista de tablero ClickUp

4. **Otras vistas de tarea:** tenemos también la posibilidad de añadir otras vistas adicionales para visualizar las tareas de otras formas. Algunas de las vistas de tareas que podemos añadir son: diagrama de Gantt, carga de trabajo o mapa mental.

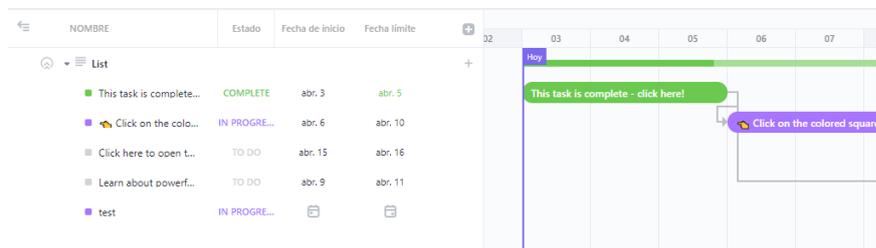


Figura 70 - Vista Diagrama de Gantt ClickUp

5. **Vistas de página:** además de vistas de tareas, podemos añadir vistas de página a nuestra lista. Algunas de las vistas de página que podemos añadir son: chat (para intercambiar mensajes con los miembros de tu lista) o whiteboard (pizarra para intercambiar ideas y resolver problemas de manera colaborativa).
6. **Creación de metas:** podemos establecer metas personales, para determinadas personas o para todo el workspace.
7. **Integraciones:** disponemos de más de 1000 aplicaciones externas para integrar con ClickUp.

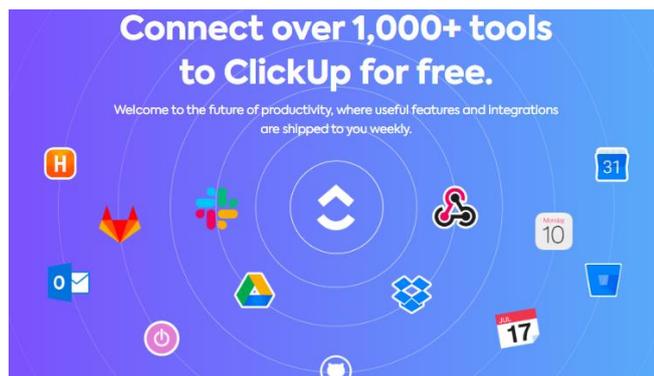


Figura 71 - Integraciones ClickUp

8. **Automatizaciones:** podemos programar automatizaciones de procesos. Disponemos de varios tipos de disparadores o condiciones (cambios de estado de tareas, fechas, cambios de persona asignada, etc.) que cuando se activen, se ejecutará de manera automática la acción que nosotros queramos (cambiar estado de la tarea, cambiar prioridad, cambiar persona asignada, crear otra subtarea, etc.).

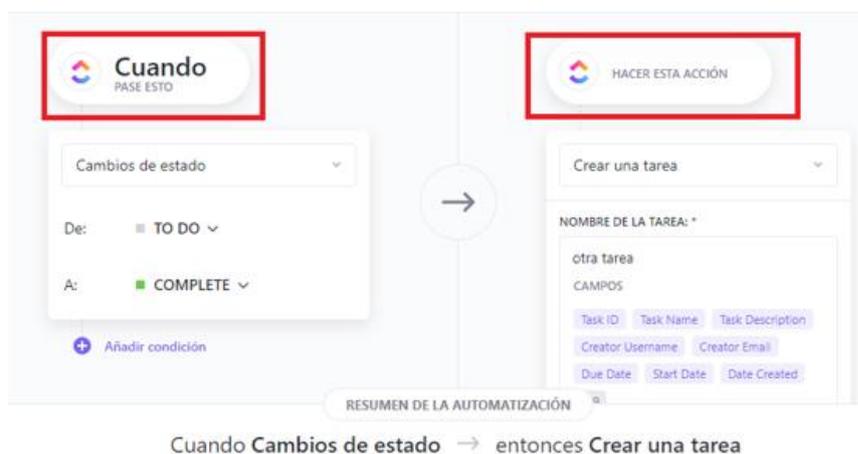


Figura 72 - Creación de automatización ClickUp

7.2.7 Taiga

7.2.7.1 Características principales

- **Página web oficial:** <https://www.taiga.io/es>
- **Compañía:** Taiga Agile LLC.
- **Fecha de lanzamiento:** 2014.
- **Licencia:** GNU Affero GPL (Código libre).
- **Planes y precios:** Taiga ofrece dos opciones: **On-Premise** o **Cloud**. Con la opción On-Premise podremos alojar la herramienta en nuestros propios servidores y/o customizar la instalación de Taiga. Disponemos de tres planes On-Premise: **Self-Managed** (gratuito, pero requiere conocimientos técnicos para instalar Taiga en tu propia infraestructura), **Automated-Hosting** (servicio de subcontratación con la compañía Elestio para el hosting e instalación de la herramienta. Precios consultar con Elestio) y **Managed for you** (para equipos que requieran soporte prioritario en la instalación y mantenimiento. Precio consultar con Taiga).

Por otro lado, tenemos las opciones Cloud, con dos planes: **Cloud Basic** (gratuito) y **Cloud con soporte premium** (70 USD al mes (o 600 USD al año con facturación anual), para los equipos que necesiten una opción en la nube, pero con soporte y asistencia premium).



Figura 73 - Planes y precios Taiga

Como vemos, uno de los valores principales de Taiga es ser una herramienta de software libre y gratuita (los planes son gratuitos salvando los servicios de instalación, mantenimiento o soporte), pero eso no quita que sea utilizada por empresas del renombre de Orange, Airbus o HP.

- **Plataformas:** Taiga sólo es accesible desde un navegador web, ya sea para acceder a la versión Cloud o a los servidores de tu compañía con las opciones On-Premise (no dispone de aplicaciones para ordenadores ni móviles).
- **Grado de documentación / información:** Media (disponen de documentación oficial, videotutoriales y de un foro en el que participan tanto miembros del equipo de Taiga como usuarios de la comunidad).

7.2.7.2 Organización de los equipos de trabajo

En Taiga, el nivel más alto de organización de equipos es el **proyecto**. Podemos crear tantos proyectos como queramos. Además, el número de usuarios por proyecto también es ilimitado en todos los planes. Esto supone una ventaja respecto a otras herramientas, las cuales en sus planes más básicos ya tienen limitaciones de proyectos y miembros.

En la figura 74 ilustramos la forma en la que se organizan los miembros de una empresa en Taiga:



Figura 74 - Organización equipos en Taiga

7.2.7.3 Interfaz del menú principal

El menú principal de Taiga se compone, por un lado, de una barra de navegación superior desde la que podemos: acceder a un dashboard de todos nuestros proyectos, movernos entre nuestros proyectos, crear un nuevo proyecto, acceder al foro de la comunidad, ver todas nuestras notificaciones o gestionar nuestro perfil:



Figura 75 - Barra de navegación superior Taiga

Por otro lado, si estamos dentro de uno de nuestros proyectos, tenemos una barra de navegación lateral para navegar entre todas las opciones que tenemos disponibles para este: epics, scrum, issues, kanban, gestionar equipo de proyecto, reunirnos con nuestro equipo o gestionar las opciones del proyecto, entre otras.

Es importante puntualizar que cuando creamos un proyecto tenemos dos plantillas por defecto: scrum o kanban. Si elegimos kanban solo tendremos la funcionalidad de kanban activada, si elegimos scrum solo tendremos las funcionalidades de scrum e issues activadas. Para activar cualquier otra opción, lo podemos hacer desde el menú de opciones del proyecto. En nuestro caso, con el fin de ver todas las funcionalidades que nos aporta Taiga, activaremos todos los módulos en un solo proyecto.

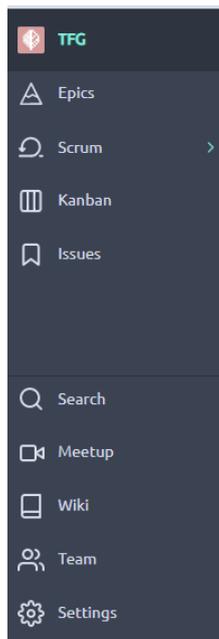


Figura 76 - Barra de navegación lateral Taiga

7.2.7.4 Revisión de funcionalidades versión gratuita

Las funcionalidades que disponemos en la versión gratuita de Taiga son las siguientes:

1. **Creación de historias:** podemos crear historias para representar los requisitos que tenemos que cumplir. Cada historia de usuario en Taiga tiene los siguientes campos: nombre, descripción, archivos adjuntos, personas asignadas, los puntos de esfuerzo asociados (de 0 a 40 puntos en diferentes campos: UX, design, front o back), fecha de vencimiento, comentarios y tareas asociadas. Además, podemos especificar si es un

requisito del usuario, un requisito del cliente o está bloqueado (por ejemplo, por tener dependencias que no se han satisfecho).

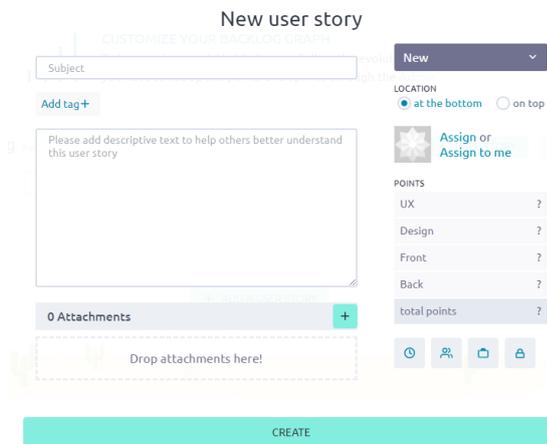


Figura 77 - Creación historia Taiga

2. **Creación de tareas:** dentro de cada historia podemos crear las tareas que consideremos necesarias para completar esta, y asignarlas a los miembros responsables.
3. **Backlog y sprints:** desde la vista “Scrum” podemos consultar el backlog con las historias de usuario no asignadas a ningún sprint. Además, desde esta misma vista, podemos crear sprints y arrastrar las historias del backlog al sprint correspondiente.

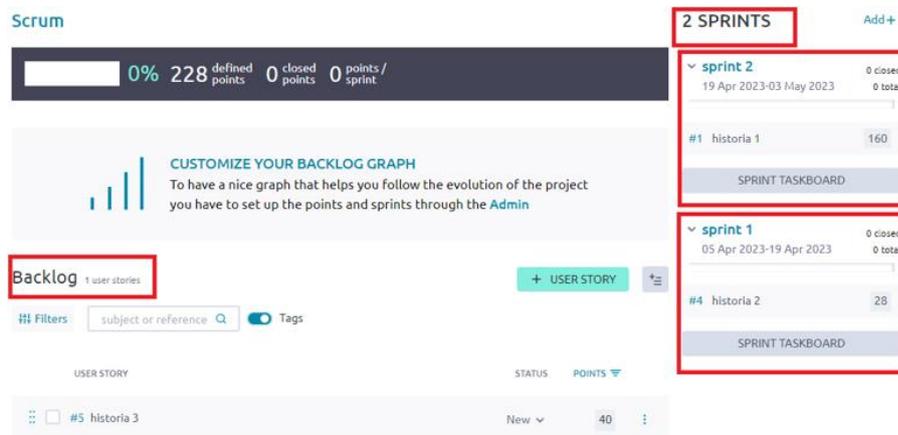


Figura 78 - Backlog y sprints Taiga

4. **Vista de tablero:** desde la vista de “Scrum” podemos consultar también un resumen de cada sprint junto con su tablero, con todas sus historias de usuario y las tareas de cada una, según su estado (cada tablero tiene las siguientes listas: new, in progress, ready for test, closed y needs info). Por otro lado, en la vista “Kanban”, podemos consultar el mismo tablero, pero con todas las historias y tareas del proyecto.

ESTUDIO Y COMPARACIÓN DE HERRAMIENTAS DE SOPORTE DE PROYECTOS ÁGILES

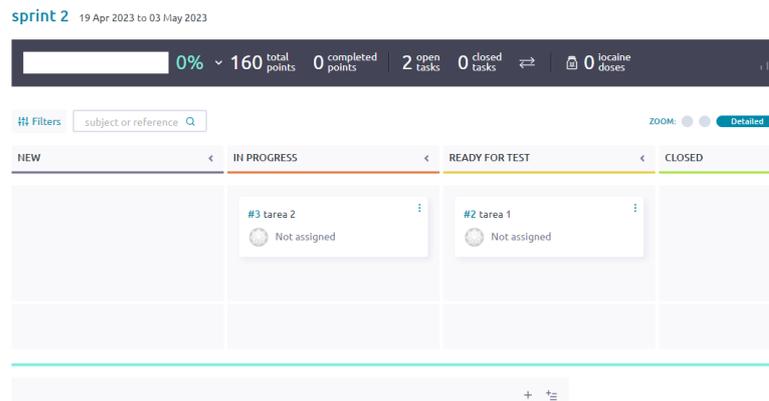


Figura 79 - Vista de tablero Taiga

5. **Creación de epics:** desde la vista de “Epics” podemos crear epics, que son colecciones de historias. Los epics en Taiga pueden agrupar historias de diferentes proyectos.
6. **Creación de issues:** desde la vista “Issues” podemos consultar y crear issues. Los issues representan incidencias del proyecto y tienen los siguientes 3 campos para ser clasificados: tipo (puede ser un bug que solucionar, una pregunta o una mejora), severidad (puede ser: deseo, menor, norma, importante o crítica) y prioridad (baja, media o alta). Además, un issue puede ser promovido a historia o ser asociado a un sprint.

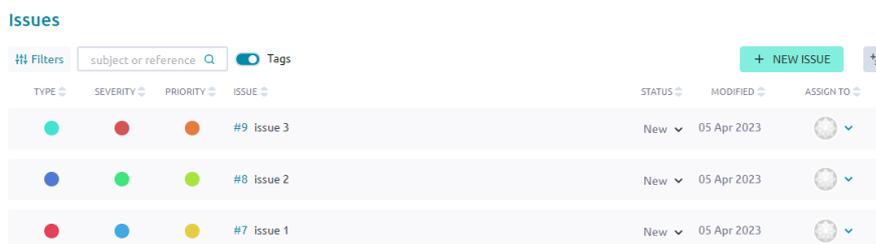


Figura 80 - Issues en Taiga

7. **Reuniones:** Taiga tiene implementada una opción para acceder directamente a la plataforma de videollamadas que tengamos configurada en ajustes del proyecto.
8. **WebHooks e integraciones:** Taiga posee un sistema webhook que permite conectar la herramienta con otras aplicaciones (un webhook envía información en tiempo real a otras aplicaciones cuando ocurre un evento. Dicho evento puede ser, por ejemplo, la creación de un sprint o de una historia en Taiga). Además, tenemos la posibilidad de integrar Taiga directamente con herramientas de repositorio como Github, Gitlab o Bitbucket, y con herramientas de mensajería como Slack.

7.2.8 Freedcamp

7.2.8.1 Características principales

- **Página web oficial:** <https://freedcamp.com>
- **Compañía:** Freedcamp Inc.
- **Fecha de lanzamiento:** 2010.
- **Licencia:** software propietario (no libre).

ESTUDIO Y COMPARACIÓN DE HERRAMIENTAS DE SOPORTE DE PROYECTOS ÁGILES

- **Planes y precios:** Freedcamp ofrece un total de cuatro planes (todos en la nube): Free (gratuito), Pro (2.49 USD por usuario al mes con facturación mensual o 1.49 USD con facturación anual), Business (8.99 USD por usuario al mes con facturación mensual o 7.49 USD al mes con facturación anual) y Enterprise (19.99 USD por usuario al mes con facturación mensual o 16.99 USD al mes con facturación anual).

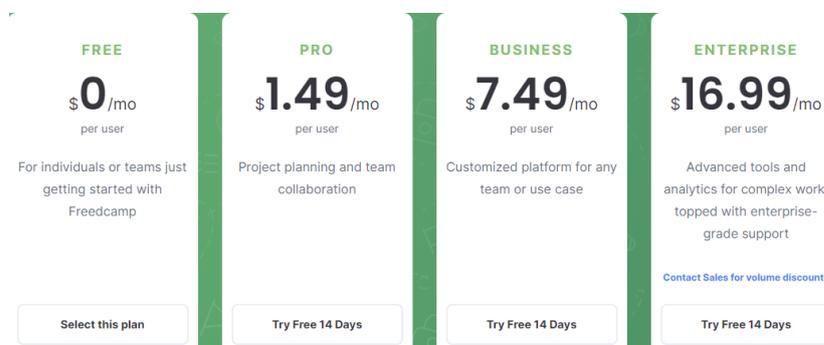


Figura 81 - Planes y precios Freedcamp

- **Plataformas:** podemos acceder a la herramienta por medio de un navegador web, por medio de su aplicación de escritorio para Windows y Mac o por medio de su App para móviles (para IOS y Android).
- **Grado de documentación / información:** Media (dentro de la herramienta, en nuestro perfil, encontramos un apartado de ayuda en el que tenemos disponibles videotutoriales sobre la herramienta, documentación oficial y un apartado de preguntas frecuentes).

7.2.8.2 Organización de los equipos de trabajo

En Freedcamp los usuarios se organizan en **Grupos** (en una organización podemos tener varios grupos). Cada grupo podemos crearlo de dos formas distintas: Simple Management System (todos los usuarios y equipos del grupo tendrán acceso a todos los proyectos de dicho grupo) o Advanced Management System (se puede tener control sobre qué usuarios tienen acceso a cada proyecto del grupo, incluidas las aplicaciones de cada proyecto) (solo disponible en planes de pago).

En el caso de que el grupo sea Simple Management System, dentro de cada grupo tenemos predefinidos un total de 5 **equipos**: Admin, Pro User, User, Guest, Read-Only. Cada uno de estos equipos tiene una serie de permisos específicos sobre los proyectos del grupo. (Un usuario puede estar en varios grupos a la vez, pero solo en un equipo de cada grupo). En el caso de que el grupo de que el grupo sea Advanced Management System, los equipos se crean a nivel de proyecto, no de grupo, y podemos crear grupos nuevos (siempre basados en esos cinco niveles de privilegios)

Por otro lado, tenemos la figura de los **Global Teams** (equipos globales), es decir, equipos a los que se les pueden asignar varios equipos específicos de diferentes grupos. Por ejemplo, si tenemos dos grupos (creados con Simple Management System): A y B, podemos crear un Global Team que tenga acceso al equipo User de A y al equipo Guest de B. Los equipos globales son útiles cuando queremos tener varias personas con los mismos privilegios en varios grupos (de esta forma no tenemos que invitarlos a cada equipo de manera individual).

Por último, como es evidente, dentro de cada grupo podemos crear tantos **proyectos** como queramos. En la figura 82, podemos apreciar la organización de los equipos en Freedcamp, con

la creación de los grupos siguiendo el sistema de gestión simple y en la figura 83 siguiendo el sistema de gestión avanzado:

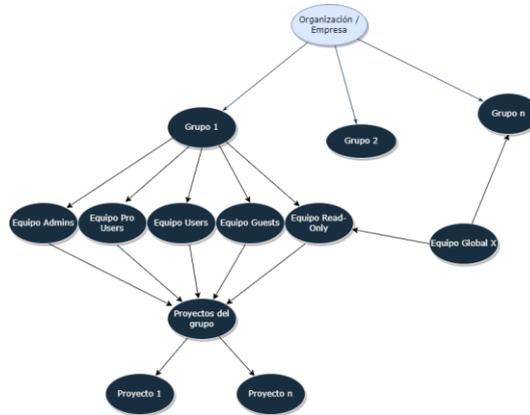


Figura 82 - Organización equipos Freedcamp (1)

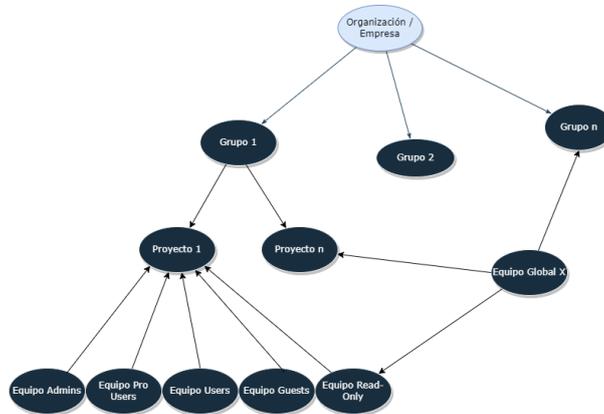


Figura 83 - Organización equipos Freedcamp (2)

7.2.8.3 Interfaz del menú principal

El menú principal de Freedcamp está compuesto por una barra de navegación lateral. Si no estamos dentro de ninguno de nuestros proyectos, podremos navegar entre los menús personales, desde los que podemos consultar información de todos nuestros proyectos: Home (vista principal con todos nuestros proyectos, nuestra agenda y nuestro trabajo pendiente), Projects (vista general de nuestros proyectos), calendario, todas nuestras tareas, hitos y cronómetros, o widgets personalizados.

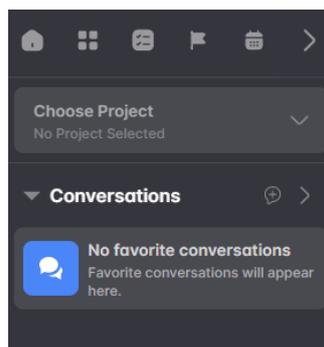


Figura 84 - Barra de navegación lateral Freedcamp (vistas generales)

Por otro lado, si seleccionamos un proyecto, tenemos acceso a las aplicaciones de dicho proyecto concreto: Tasks, Discussions, Milestones, Time, Calendar y Files.

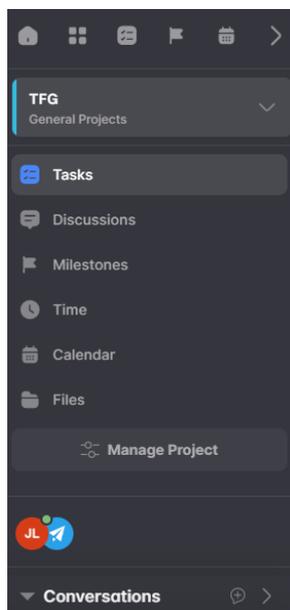


Figura 85 - Barra de navegación lateral Freedcamp (vistas del proyecto)

Además, en todo momento, estemos dentro de un proyecto o no, tenemos acceso a nuestras conversaciones, las cuales comentaremos en el siguiente punto qué nos ofrecen.

7.2.8.4 Revisión de funcionalidades versión gratuita

Las funcionalidades a las que tenemos acceso con la versión gratuita de Freedcamp son las siguientes:

1. **Creación de hitos:** podemos crear hitos (“milestones”) dentro de nuestros proyectos. Cada hito está compuesto por los siguientes elementos: un título, una descripción, una fecha de vencimiento, una prioridad (baja, media o alta), una persona o personas asignadas, archivos adjuntos, comentarios y un conjunto de tareas.
2. **Creación de tareas:** podemos crear tareas para representar el trabajo a realizar en nuestro proyecto. Cada tarea se compone de los mismos campos que los hitos y suma los siguientes: status (sin empezar, en progreso o completada) e hito al que pertenece (puede no pertenecer a ninguno). En versiones de pago podemos acceder a otros campos como, por ejemplo, subtareas.
3. **Vistas de tablero Kanban y lista:** podemos visualizar las tareas que tengamos creadas de dos formas diferentes: en forma de lista o en forma de tablero Kanban. En planes superiores, además, podemos visualizarlas en forma de tabla o en un diagrama de Gantt.

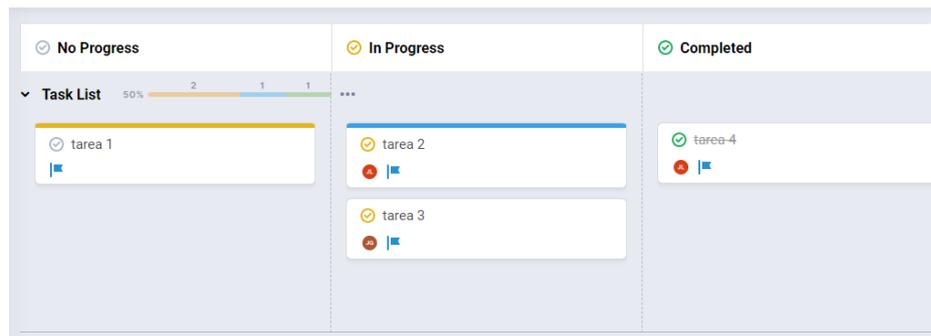


Figura 86 - Vista tablero Kanban Freedcamp

4. **Discusiones:** podemos iniciar discusiones en nuestros proyectos. Las discusiones son una forma útil de representar nuestras preocupaciones o dudas sobre el proyecto e iniciar debates con el resto de los miembros.
5. **Creación de listas:** podemos crear listas para agrupar tanto las tareas como las discusiones de nuestro proyecto.
6. **Cronómetros:** podemos controlar el tiempo que tardamos en realizar determinadas actividades del proyecto (dichas actividades pueden estar enlazadas a tareas, como en el ejemplo) estableciendo un cronómetro que podemos pausar y reanudar. Cuando terminemos la actividad pulsaremos en completar y veremos el tiempo que hemos tardado.

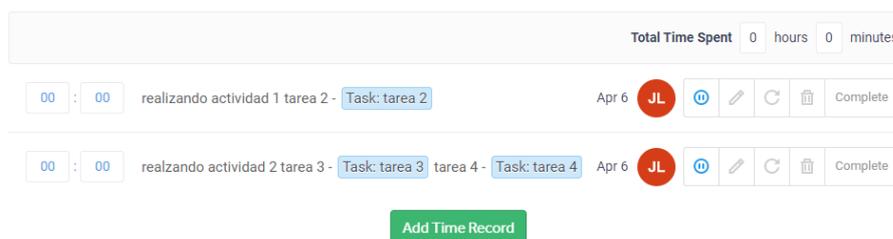


Figura 87 - Cronómetros de actividades Freedcamp

7. **Calendario:** podemos visualizar todos los elementos del proyecto, tales como tareas o hitos, en un calendario con sus respectivas fechas de vencimiento.
8. **Carpetas y archivos:** tenemos también un apartado para crear carpetas y subir todos los archivos relacionados con el proyecto.
9. **Conversaciones:** desde el apartado de discusiones, podremos crear chats con miembros concretos que agreguemos (deben ser usuarios que estén en algún grupo común con nosotros). En dichos chats podemos intercambiar mensajes, archivos e incluso realizar videollamadas.
10. **Extensiones:** podemos ampliar las funcionalidades de Freedcamp mediante complementos o extensiones. En la versión gratuita solo podemos añadir algunos pocos complementos (la mayoría ya vienen activados por defecto) como “passwords” o “tasky”. Sin embargo, en los planes de pago ya tenemos más oferta como los complementos de Dropbox o Google Drive, entre otros.

7.2.9 Monday.com

7.2.9.1 Características principales

- **Página web oficial:** <https://monday.com/lang/es/work-management>
- **Compañía:** Monday.com Inc.
- **Fecha de lanzamiento:** 2014.
- **Licencia:** software propietario (no libre).
- **Planes y precios:** antes de nada, es importante mencionar que Monday.com ofrece varios productos como “Monday dev” o “Monday sales CRM”. En nuestro caso, sólo estudiaremos su producto principal y original llamado “Monday work management”.

Monday.com ofrece un total de 5 planes (todos en la nube): **Individual** (gratuito y hasta 2 usuarios como máximo por cuenta), **Básico** (10 € por usuario al mes con facturación mensual u 8 € con facturación anual, sin límite de usuarios), **Estándar** (12 € por usuario al mes con facturación mensual o 10 € con facturación anual), **Pro** (20 € por usuario al mes con facturación mensual o 16 € con facturación anual) y **Corporativo** (consultar precio con ellos).

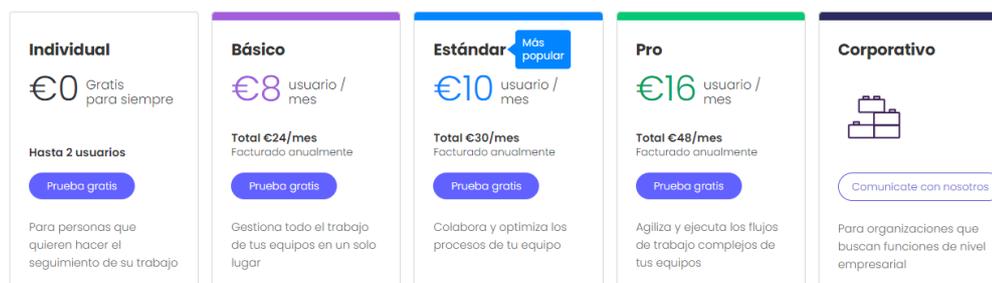


Figura 88 - Planes y precios Monday.com

- **Plataformas:** tenemos disponible una aplicación de escritorio para Mac y Windows, una App para móviles (IOS y Android) y, como es habitual, el acceso a través de un navegador web.
- **Grado de documentación / información:** Alta (Monday.com pone a nuestra disposición videotutoriales de la herramienta, un foro de la comunidad, documentación oficial y seminarios web en vivo o grabados).

7.2.9.2 Organización de los equipos de trabajo

En Monday.com, el nivel más alto en la jerarquía de organización de las personas son las **cuentas**. En una compañía, lo más común es tener una sola cuenta, aunque en empresas grandes y con un número considerable de usuarios, quizás pueda ser útil tener varias cuentas para mantener la información de la organización más separada y segura. Dentro de una cuenta, cada usuario puede tener diferentes roles de permisos dentro de esta, como, por ejemplo, tener permisos para editar y configurar la cuenta (administrador) o tener sólo permisos para visualizar (sin editar) información (espectador).

Cada cuenta se distribuye en varios **espacios de trabajo**. Los espacios de trabajo son lugares virtuales que nos permiten que los equipos de nuestra organización puedan colaborar, organizar y visualizar su trabajo de forma conjunta. Podemos crearlos abiertos o cerrados: si son abiertos no podremos controlar quién tiene acceso a él (tendrán acceso todos los miembros de la cuenta),

si son cerrados podremos especificar qué miembros de la cuenta puede acceder a su contenido (solo disponible en el plan corporativo).

Cada espacio de trabajo está compuesto por varios **tableros** y estos, a su vez, pueden estar organizados en **carpetas**. Los tableros también podemos elegir si queremos que sean públicos (accesible para todos los miembros de la cuenta), privados (solo accesible por mí o por algunos miembros de la cuenta) o compartible (para invitar a miembros de fuera de la cuenta).

Además, en Monday.com tenemos la figura de los equipos. Los equipos son grupos de personas con objetivos comunes (pueden ser usuarios que trabajan en un mismo proyecto, miembros de un mismo departamento, etc.). La figura de los equipos puede ser útil para asignar permisos o tareas de manera conjunta o para facilitar la comunicación entre esas personas.

A continuación, en la figura 89, ilustramos la forma en la que se organizan los miembros de una organización en Monday.com (la figura se ha hecho presuponiendo que el “espacio de trabajo 2” y el “tablero n” son privados y, por lo tanto, se necesita añadir miembros a ellos. Es por eso que “equipo” está unido a ambos. Si el espacio de trabajo o el tablero fuesen públicos no sería necesario añadir ningún miembro).

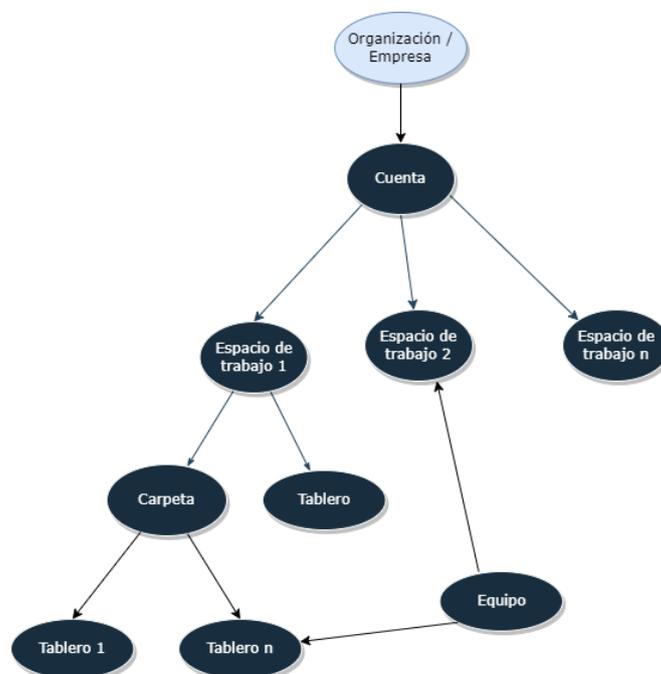


Figura 89 - Organización equipos en Monday.com

7.2.9.3 Interfaz del menú principal

El menú principal de Monday.com se organiza en torno a dos barras de navegación lateral. La primera, es la barra principal del programa y podemos acceder a diferentes vistas como la página de inicio, nuestras notificaciones y tareas, los ajustes de nuestra cuenta, podemos agregar nuevos productos, invitar nuevos miembros a nuestra cuenta o acceder a los productos que tengamos instalados (en nuestro caso, como ya hemos comentado, nos centraremos en el producto “work management”).

La segunda barra lateral podemos observarla cuando accedemos a nuestro producto “work management”. Desde esta barra, podemos acceder a los diferentes espacios de trabajo que tengamos o a los elementos que componen este.



Figura 90 - Barra principal Monday.com

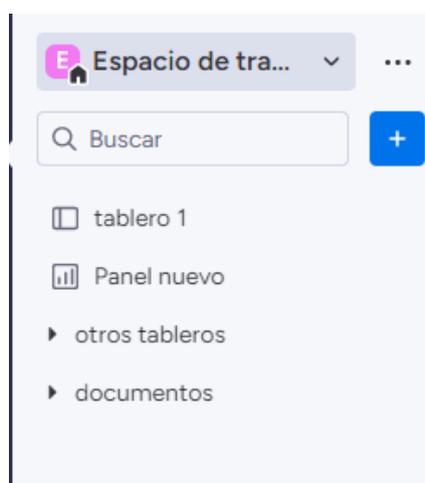


Figura 91 – Barra work management Monday.com

7.2.9.4 Revisión de funcionalidades versión gratuita

Las funcionalidades gratuitas (del producto “work management”) que podemos acceder son las siguientes:

1. **Creación de tareas:** podemos crear tareas dentro de cada tablero. Una tarea en Monday.com se compone de los siguientes campos: grupo de tareas al que pertenece, nombre, subelementos, personas asignadas, estado (en curso, detenido, listo o sin estado (blanco), fecha de vencimiento, archivos adjuntos y actualizaciones del progreso.
2. **Creación de grupos de tareas:** podemos crear grupos para agrupar las tareas por algún criterio (por ejemplo, por defecto Monday.com crea dos grupos: “esta semana” y “próxima semana”).
3. **Vistas de kanban, tabla, y tarjetas:** podemos visualizar las tareas en tres vistas diferentes: en un tablero kanban, en una tabla o en forma de tarjetas.

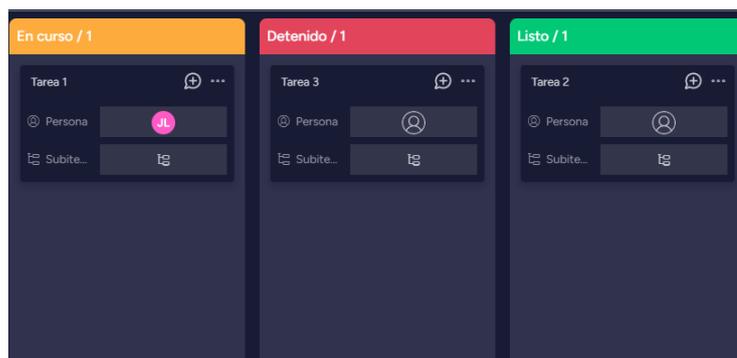


Figura 92 - Vista tablero Kanban Monday.com

4. **Creación de paneles:** dentro de nuestro espacio de trabajo podemos crear paneles. Los paneles nos permiten visualizar varios widgets de nuestros tableros (en las versiones superiores podemos conectar varios tableros y visualizarlos de manera conjunta). Para la visualización de la información, podemos agregar varios widgets como: diagrama de tareas, batería de progreso o diagrama Gantt (algunos de ellos, como el diagrama de tareas o de Gantt solo están disponibles en versiones superiores).



Figura 93 - Panel en Monday.com

5. **Creación de documentos:** dentro de nuestro espacio de trabajo podemos crear documentos. Los documentos en Monday.com disponen de varias plantillas para rellenar la información de manera simple y rápida, como por ejemplo: las notas de una reunión, la lista de tareas pendientes, las especificaciones de una función o el plan del proyecto.

The screenshot shows the 'Plan de proyecto' (Project Plan) template in Monday.com. It includes a title field 'Nombre de tu proyecto', three input fields for 'Gestor de proyectos', 'Miembros del equipo', and 'Fecha', and a section for 'Objetivos' with a checkbox. Below that is the 'Alcance' (Scope) section, which contains a table with three rows: 'Obligatorio' (Required), 'Deseable' (Desirable), and 'Fuera del alcance' (Out of scope). Each row has a checkbox and a text prompt. At the bottom, there is a 'Cronograma del proyecto' (Project Gantt chart) icon.

Categoría	Descripción
Obligatorio	<input type="checkbox"/> Agrega los requisitos obligatorios de tu proyecto
Deseable	<input type="checkbox"/> Agrega cualquier cosa que quieras, pero que no sea estrí...
Fuera del alcance	<input type="checkbox"/> Agrega todo lo que no quieras incluir

Figura 94 - Plantilla plan de proyecto Monday.com

6. **Aplicaciones:** se pueden extender las funcionalidades de la herramienta con la instalación de aplicaciones externas gratuitas (diferente a la funcionalidad de integraciones que sólo está disponible en planes superiores).

7.2.10 Zoho Sprints

7.2.10.1 Características principales

- **Página web oficial:** <https://www.zoho.com/sprints/>
- **Compañía:** Zoho Corporation.
- **Fecha de lanzamiento:** 2018.
- **Licencia:** software propietario (no libre).
- **Planes y precios:** disponemos de un total de 4 planes para usar Zoho Sprints (todas las opciones son en la nube): **Free** (gratuito, pero limitado a 3 usuarios y proyectos), **Starter** (1 € por usuario al mes (solo disponible con facturación anual), límite de 12 usuarios y 50 proyectos), **Elite** (3 € por usuario al mes con facturación mensual o 2.50 € con facturación anual, límite de 100 usuarios y proyectos ilimitados) y **Premier** (6 € por usuario al mes con facturación mensual o 5 € con facturación anual, sin límite de usuarios ni proyectos).

The screenshot displays three pricing plans for Zoho Sprints: Starter, Elite, and Premier. Each plan is presented in a vertical card with its price, billing cycle, user limit, and a 'TRY NOW' button. Below the cards, a comparison table lists features for each plan.

Plan	Precio	Billing	Usuarios	Proyectos	Almacenamiento	Plantillas
STARTER	€1	/user/month billed annually	12 users by default	50 Projects	20 GB	5 Project Templates
ELITE	€2,50	/user/month billed annually	Up to 100 users	Unlimited Projects	50 GB	10 Project Templates
PREMIER	€5	/user/month billed annually	No upper limit	Unlimited Projects	100 GB	20 Project Templates

Figura 95 - Planes y precios Zoho Sprints

- **Plataformas:** podemos acceder a Zoho Sprints a través de un navegador web y de su aplicación para móviles disponible en Android e IOS (no dispone de aplicación de escritorio para PCs).
- **Grado de documentación / información:** Alta (Zoho Sprints dispone de documentación oficial, videotutoriales y webinars tanto en vivo como grabados).

7.2.10.2 Organización de los equipos de trabajo

En Zoho Sprints, los usuarios se organizan en **Equipos** (una organización puede tener varios equipos). Cuando invitamos a un usuario a un equipo, podemos asignarle tres roles diferentes: miembro, gestor y administrador. Además, le podemos asignar a uno o varios proyectos del equipo (no tiene que estar en todos los proyectos del equipo). Por otro lado, también podemos invitar al equipo a usuarios con los roles de cliente o proveedor.

Dentro de cada equipo podemos crear **Proyectos** (dichos proyectos los podemos agrupar en grupos, para una mejor organización). Cuando invitamos a un usuario a un proyecto concreto, ese usuario será añadido también al equipo, aunque tendrá acceso solamente a ese proyecto (hasta que le asignemos otro proyecto más).

En la figura 96, representamos la forma en la se organizan los miembros de una organización en Zoho Sprints:



Figura 96 - Organización equipos Zoho Sprints

7.2.10.3 Interfaz del menú principal

El menú principal de Zoho Sprints está compuesto de una barra de navegación lateral que contiene todas las funcionalidades que podemos acceder de cada proyecto: panel de información, fuente, trabajo pendiente, epics (disponible sólo en versiones superiores), versión (disponible sólo en versiones superiores), panel, reuniones (disponible sólo en versiones superiores) e informes.

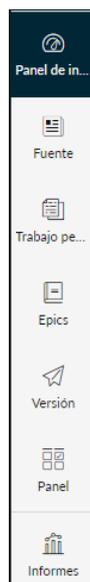


Figura 97 - Barra de navegación lateral Zoho Sprints

7.2.10.4 Revisión de funcionalidades versión gratuita

Las funcionalidades que disponemos en la versión gratuita de Zoho Sprints son las siguientes:

1. **Creación de sprints:** dentro de cada proyecto (en la vista de “trabajo pendiente”), podemos crear sprints. Cada sprint tiene los siguientes campos: nombre, descripción, propietario, usuarios y duración. Además, una vez creado el sprint, podemos acceder a diferentes vistas de este como: su panel de información con diferentes gráficos que resumen el estado de este, sus comentarios (en los que podemos mencionar a usuarios concretos y comentar aspectos del sprint) o los elementos de los que está compuesto.

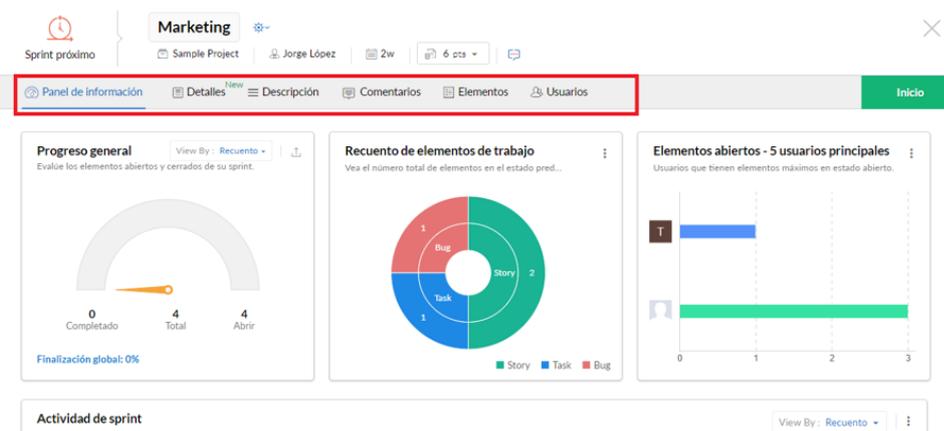


Figura 98 - Sprint creado en Zoho Sprints

2. **Creación de tareas, bugs e historias:** en la vista de “trabajo pendiente” podemos crear también elementos. Los elementos en Zoho Sprints pueden ser de tres tipos (historias, tareas o bugs). Además, cada una de estas tareas las podemos crear en “tareas pendientes” (se añadirán al backlog) o directamente en un sprint). Además, a cada elemento le podemos añadir una prioridad (baja, media o alta) o el esfuerzo requerido para completar dicho elemento.
3. **Backlog:** en la vista de “trabajo pendiente”, podemos visualizar dos columnas: “trabajo pendiente” y “trabajo pendiente del sprint”. En la primera de ellas, se encuentran los elementos (historias, tareas o bugs) que no han sido asignados todavía a ningún sprint.

ESTUDIO Y COMPARACIÓN DE HERRAMIENTAS DE SOPORTE DE PROYECTOS ÁGILES

Mientras que en la segunda columna vemos los sprints que no han sido todavía iniciados. De este modo, podemos arrastrar los elementos de la columna de la izquierda (backlog) al trabajo pendiente de los sprints (columna derecha).

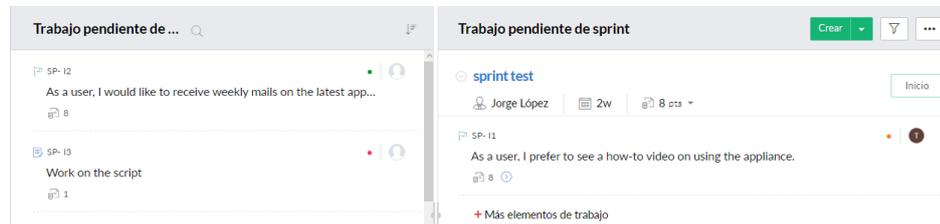


Figura 99 - Backlog Zoho Sprints

4. **Vista de tablero Kanban:** en la vista “panel” podemos visualizar los elementos del sprint activo en ese determinado momento. El tablero dispone de tres columnas: to do, in progress y done.
5. **Dashboard:** en la vista “panel de inicio” disponemos de un dashboard en el que podemos visualizar diferentes “widgets”, como el de cronograma, que muestran el estado general del proyecto.

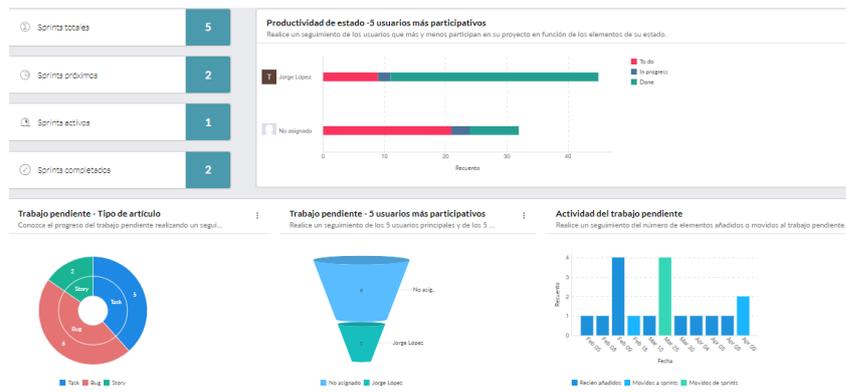


Figura 100 - Vista “Dashboard” Zoho Sprints

6. **Informes:** en la pestaña de “informes” disponemos de un total de 7 informes de sprint. Los informes a los que tenemos acceso son los siguientes: velocidad, trabajo pendiente, rendimiento, flujo acumulativo, latencia, sprints completados y sprints cancelados.
7. **Integraciones:** podemos integrar Zoho Sprints tanto con aplicaciones externas como Github u Office 365, como con aplicaciones de Zoho como Zoho Analytics o Zoho BugTracker.

7.3 Criterios de comparación

A continuación, enumeraremos y describiremos cada uno de los criterios que usaremos para comparar los planes de las herramientas vistas anteriormente (los siguientes criterios son para los planes en la nube. Para los planes On-Premise son los mismos, pero se han eliminado los criterios de aplicación para escritorio y móviles, y cambiado los criterios de: “precio por usuario y mes” a “Precio licencia anual” y “autenticación de dos factores” a “autenticación”). Para ello, dividiremos los criterios en cuatro grupos o bloques: criterios generales, almacenamiento y soporte, seguridad de los datos y funcionalidades.

La elección de los presentes criterios está basada en la revisión de los estudios [32], [33] y [35], los cuales realizan también un estudio de comparación de herramientas de soporte para

proyectos ágiles y establecen criterios para ello. Además, se añaden nuevos criterios (para que la lista sea lo más completa posible) basados en los estudios realizados en el presente trabajo de cada una de las herramientas y en la documentación disponible de cada una de ellas en sus respectivos sitios web. En base a todo ello, se ha elaborado un listado de criterios que veremos a continuación.

7.3.1 Bloque 1 – Criterios generales

1. **Precio por usuario y mes:** el precio por usuario y por mes del plan (en USD) (se tomará la facturación mensual para la comparación en los casos que se pueda. Si solo se permite la facturación anual se refleja en la tabla).
2. **Licencia de uso:** licencia de uso de la herramienta (software propietario u Open Source).
3. **Cloud/On-Premise:** tipo de solución (Cloud u On-Premise).
4. **Aplicación de escritorio:** dispone de aplicación de escritorio o no.
5. **Aplicación para móviles:** dispone de aplicación para móviles o no.
6. **Documentación:** grado de documentación e información que podemos encontrar sobre la herramienta (tales como webinars, videotutoriales, foros oficiales de la comunidad, etc.).
7. **Usuarios destinados:** tipo de usuario al que está destinado el plan de dicha herramienta por sus características (freelancers, empresa pequeña, empresa mediana o empresa grande).
8. **Mínimo de usuarios:** mínimo de usuarios para poder comprar el plan.
9. **Máximo de usuarios:** máximo de usuarios del plan.
10. **Límite de proyectos:** límite de proyectos que podemos crear (o en su defecto, en algunas herramientas, límite de tableros).
11. **División en varios equipos / espacios:** posibilidad de dividir a los miembros de la compañía o proyecto en varios equipos o espacios de trabajo.
12. **Límite de equipos / espacios:** en caso de admitir varios equipos o espacios de trabajo, cuál es el límite de estos.

7.3.2 Bloque 2 – Almacenamiento y Soporte

1. **Almacenamiento:** capacidad de almacenamiento que tenemos por usuario, proyecto o archivo.
2. **Soporte:** tipo de soporte y prioridad que tenemos derecho (por ejemplo: 24/7 y prioritario).

7.3.3 Bloque 3 – Seguridad de la información

1. **Grupos de roles y permisos:** posibilidad de establecer grupos de roles de usuario con diferentes permisos en nuestros proyectos o espacios de trabajo.
2. **Autenticación en dos factores:** posibilidad de autenticación de los usuarios en dos factores, para mayor seguridad de los datos.

7.3.4 Bloque 4 – Funcionalidades

1. **Vista tablero Kanban:** posibilidad de consultar las tareas en un tablero Kanban.
2. **Vista cronograma del proyecto:** posibilidad de consultar el cronograma del proyecto en un diagrama de Gantt o en un diagrama “timeline”.
3. **Vista de calendario:** posibilidad de consultar el trabajo pendiente en un calendario.

4. **Dependencias:** posibilidad de crear relaciones de dependencia entre tareas.
5. **Etiquetas:** posibilidad de crear etiquetas personalizadas para clasificar y organizar diferentes elementos del proyecto como tareas o historias.
6. **Campos personalizados:** posibilidad de crear campos personalizados en tareas.
7. **Comentarios:** posibilidad de dejar comentarios o notas en elementos del proyecto como tareas o sprints.
8. **Menciones a otros miembros:** posibilidad de mencionar a otros miembros del proyecto o equipo.
9. **Archivos adjuntos:** posibilidad de gestionar y adjuntar archivos o documentos en las tareas o ítems del proyecto.
10. **Generación de gráficos e informes:** posibilidad de generar informes gráficos de estado del proyecto.
11. **Tableros modificables:** posibilidad de modificar los tableros Kanban con status o columnas personalizadas.
12. **Plantillas de documentos:** posibilidad de generar documentos (tales como planes de proyecto o actas de reunión) a partir de plantillas.
13. **Plantillas de proyecto:** posibilidad de crear proyectos a partir de plantillas predeterminadas.
14. **Automatizaciones:** posibilidad de incorporar automatizaciones predeterminadas o crear las tuyas propias.
15. **Integraciones:** posibilidad de integrar la herramienta con aplicaciones externas.
16. **Importar / Exportar datos:** posibilidad de importar y exportar datos en nuestros proyectos.

7.4 Tablas comparativas

Una vez vistos y explicados los criterios que usaremos, procederemos a comparar los diferentes planes de las herramientas. Para ello, primero veremos todas las opciones Cloud disponibles y posteriormente todas las versiones On-Premise. Además, los planes Cloud, al ser más numerosos, los dividiremos en diferentes niveles: el primer nivel serán todos los planes gratuitos de cada herramienta, el segundo serán todos los segundos planes más simples de cada herramienta, y así hasta llegar hasta los planes más completos o “Premium” de cada una. En total, habrá 5 niveles, ya que es el máximo de planes que ofrecen las herramientas (Si alguna herramienta sólo tiene, por ejemplo, 4 planes en la nube, no aparecerá en la tabla de nivel 5).

ESTUDIO Y COMPARACIÓN DE HERRAMIENTAS DE SOPORTE DE PROYECTOS ÁGILES

7.4.1 Planes Cloud

7.4.1.1 Nivel 1

Criterios	Planes									
	Trello - Free	Jira - Free	Asana - Basic	Wrike - Free	TeaScrum - Free	ClickUp - Free	Taiga Cloud - Basic	FreeCamp - Free	Monday.com - Individual	Zoho Sprints - Free
ASPECTOS GENERALES										
Precio por usuario y mes	0 \$	0 \$	0 \$	0 \$	0 \$	0 \$	0 \$	0 \$	0 \$	0 \$
Licencia de uso	S. Propietario	S. Propietario	S. Propietario	S. Propietario	GNU/GPL	S.Propietario	GNU Affero GPL	S.Propietario	S.Propietario	S.Propietario
Cloud/On-Premise	Cloud	Cloud	Cloud	Cloud	Cloud	Cloud	Cloud	Cloud	Cloud	Cloud
App escritorio	✓	✗	✓	✓	✗	✓	✗	✓	✓	✗
App móviles	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓
Documentación	Alta	Alta	Alta	Alta	Media	Alta	Media	Media	Alta	Alta
Usuarios destinados	Freelancers	Freelancers	Freelancers	Freelancers	Freelancers	Freelancers	Freelancers o empresas pequeñas	Freelancers	Freelancers	Freelancers
Mínimo usuarios	No	No	No	No	No	No	No	No	No	No
Máximo usuarios	No	10	15	No	3	No	No	No	2	3
Límite proyectos	10/Workspace	Ilimitados	Ilimitados	Ilimitados	Ilimitados	100/espacio	Ilimitados	Ilimitados	3	3
División en varios equipos	✓	✓	✓	✓	✗	✓	✗	✓	✓	✓
Límite de equipos o espacios	Sí (no especificado)	Ilimitados	Ilimitados	Sí (no especificado)	No Aplica	5	No Aplica	Ilimitados	Ilimitados	Ilimitados
ALMACENAMIENTO Y SOPORTE										
Almacenamiento	10MB/file	2GB/user	100MB/file	2GB/user	No especificado	100MB/user	10GB/proyecto	10MB/file	500MB/user	500MB/user
SopORTE	Community	Community	Asistencia de la comunidad y sin atención 24h	SopORTE estándar de Wrike	Foro	24/7 Support	SopORTE de la comunidad	Standard	SopORTE estándar de Monday.com	SopORTE estándar de Zoho Sprints
SEGURIDAD DE LA INFORMACIÓN										
Grupos de roles y permisos	✗	✗	✗	✗	✗	✗	✓	✓	✗	✗
Autenticación en dos factores	✓	✗ (Requiere Atlassian Access)	✓	✗	✗	✓	✗	✗	✓	✓
FUNCIONALIDADES										
Vista tablero Kanban	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Vista cronograma del proyecto	✗	✓	✗	✗	✓	✓	✗	✗	✗	✓
Vista calendario	✗	✗	✓	✗	✗	✓	✗	✓	✗	✗
Dependencias	✗	✓	✗	✗	✓	✓	✓	✗	✗	✗
Etiquetas	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗
Campos personalizados	✗	✓	✗	✗	✗	✓	✓	✗	✓	✗
Comentarios	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Menciones a otros miembros	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓
Archivos adjuntos	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Generación de gráficos e informes	✗	✓	✓	✗	✓	✗	✓	✗	✓	✓
Tableros modificables	✓	✓	✓	✗	✗	✓	✓	✗	✓	✗
Plantillas de documentos	✗	✗	✗	✗	✗	✓	✗	✗	✓	✗
Plantillas de proyecto	✓	✓	✗	✗	✗	✓	✓	✓	✓	✗
Automatizaciones	✓	✓	✗	✗	✗	✓	✗	✗	✗	✗
Integraciones	✓	✓	✓	✓	(limitado a 1)	✓	✓	✗	✗	✓
Exportar e importar datos	✗	✓	✗	(exportar)	(exportar)	✓	✓	✓	✓	✓

Tabla 4 - Comparación planes Cloud nivel 1

ESTUDIO Y COMPARACIÓN DE HERRAMIENTAS DE SOPORTE DE PROYECTOS ÁGILES

7.4.1.2 Nivel 2

Criterios	Planes									
	Trello - Standard	Jira - Standard	Asana - Premium	Wrike - Team	TeaScrum - Small Team	ClickUp - Unlimited	Taliga Cloud - Premium	Freecamp - Pro	Mombam.com - Básico	Zoho Sprints - Starter
ASPECTOS GENERALES										
Precio por usuario y mes	6 \$	7.75 \$	13.49 \$	9.80 \$	9.82 \$ (7 usuarios)	9 \$	70 \$ (fijo)	2.49 \$	8.82 \$	1.1 \$ (fact. Anual)
Licencia de uso	S. Propietario	S. Propietario	S. Propietario	S. Propietario	GNU/GPL	S.Propietario	GNU Affero GPL	S.Propietario	S.Propietario	S.Propietario
Cloud/On-Premise	Cloud	Cloud	Cloud	Cloud	Cloud	Cloud	Cloud	Cloud	Cloud	Cloud
App escritorio	✓	✗	✓	✓	✗	✓	✗	✓	✓	✗
App móviles	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓
Documentación	Alta	Alta	Alta	Alta	Media	Alta	Media	Media	Alta	Alta
Usuarios destinados	Empresas pequeñas	Empresas pequeñas o medianas	Empresas pequeñas	Freelancers o empresas pequeñas	Freelancers	Empresas pequeñas	Empresas pequeñas o medianas	Empresas pequeñas	Empresas pequeñas	Freelancers o empresas pequeñas
Mínimo usuarios	No	No	No	2	No	No	No	No	No	No
Máximo usuarios	No	35.000	No	25	7	No	No	No	No	12
Límite proyectos	Ilimitados	Ilimitados	Ilimitados	Ilimitados	Ilimitados	200/espacio	Ilimitados	Ilimitados	Ilimitados	50
División en varios equipos	✓	✓	✓	✓	✗	✓	✗	✓	✓	✓
Límite de equipos o espacios	Sí (no especificado)	Ilimitados	Ilimitados	Ilimitados	No Aplica	Ilimitados	No Aplica	Ilimitados	Ilimitados	Ilimitados
ALMACENAMIENTO Y SOPORTE										
Almacenamiento	250MB/file	250GB/user	100MB/file	2GB/user	No especificado	Ilimitado	10GB/proyecto	25MB/file	5GB/user	20GB/user
SopORTE	Local Business Hours	Local Business Hours	Asistencia de la comunidad y sin atención 24h	SopORTE estándar de Wrike	Foro	24/7 y Live Chat Support	SopORTE y asistencia premium	Premium	Asistencia al cliente prioritaria	SopORTE estándar de Zoho Sprints
SEGURIDAD DE LA INFORMACIÓN										
Grupos de roles y permisos	✗	✓	✓	✗	✗	✗	✓	✓	✗	✗
Autenticación en dos factores	✓	✗ (Requiere Atlassian Access)	✓	✗	✗	✓	✗	✗	✓	✓
FUNCIONALIDADES										
Vista tablero Kanban	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Vista cronograma del proyecto	✗	✓	✓	✓	✓	✓	✗	✗	✗	✓
Vista calendario	✗	✗	✓	✓	✗	✓	✗	✓	✗	✗
Dependencias	✗	✓	✓	✓	✓	✓	✓	✗	✗	✗
Etiquetas	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Campos personalizados	✓	✓	✓	✓	✗	✓	✓	✗	✓	✗
Comentarios	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Menciones a otros miembros	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓
Archivos adjuntos	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Generación de gráficos e informes	✗	✓	✓	✓	✓	✗	✓	✗	✓	✓
Tableros modificables	✓	✓	✓	✓	✗	✓	✓	✗	✓	✓
Plantillas de documentos	✗	✗	✗	✓	✗	✓	✗	✗	✓	✗
Plantillas de proyecto	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓
Automatizaciones	✓	✓	✓	✓	✗	✓	✗	✗	✗	✗
Integraciones	✓	✓	✓	✓	✓ (limitado a 3)	✓	✓	✓	✗	✓
Exportar e importar datos	✗	✓	✗	✓	✓	✓	✓	✓	✓	✓

Tabla 5 - Comparación planes Cloud nivel 2

ESTUDIO Y COMPARACIÓN DE HERRAMIENTAS DE SOPORTE DE PROYECTOS ÁGILES

7.4.1.3 Nivel 3

Criterios	Planes									
	Trello - Premium	Jira - Premium	Asana - Business	Wrike - Business	IceScrum - Teams	CheckUp - Business		Freecamp - Business	Monday.com - Estándar	Zoho Sprints - Elite
ASPECTOS GENERALES										
Precio por usuario y mes	12.50 \$	12.71 \$	30.49 \$	24.80 \$	44 \$ (para 1 equipo)	19 \$		8.99 \$	11.02 \$	3.30 \$
Licencia de uso	S. Propietario	S. Propietario	S. Propietario	S. Propietario	GNU/GPL	S. Propietario		S. Propietario	S. Propietario	S. Propietario
Cloud/On-Premise	Cloud	Cloud	Cloud	Cloud	Cloud	Cloud		Cloud	Cloud	Cloud
App escritorio	✓	✗	✓	✓	✗	✓		✓	✓	✗
App móviles	✓	✓	✓	✓	✓	✓		✓	✓	✓
Documentación	Alta	Alta	Alta	Alta	Media	Alta		Media	Alta	Alta
Usuarios destinados	Empresas pequeñas o medianas	Empresas medianas	Empresas medianas	Empresas pequeñas o medianas	Empresas pequeñas o medianas	Empresas medianas		Empresas medianas	Empresas pequeñas o medianas	Empresas pequeñas o medianas
Mínimo usuarios	No	No	No	5	No	No		No	No	No
Máximo usuarios	No	35.000	No	200	No	No		No	No	100
Límite proyectos	Ilimitados	Ilimitados	Ilimitados	Ilimitados	Ilimitados	400/espacio		Ilimitados	Ilimitados	Ilimitados
División en varios equipos	✓	✓	✓	✓	✓	✓		✓	✓	✓
Límite de equipos o espacios	Sí (no especificado)	Ilimitados	Ilimitados	Ilimitados	5	Ilimitados		Ilimitados	Ilimitados	Ilimitados
ALMACENAMIENTO Y SOPORTE										
Almacenamiento	250MB/file	Ilimitado	100MB/file	5GB/user	No especificado	Ilimitado		100MB/file	20GB/user	50GB/user
SopORTE	24/5 Premium	24/7 Premium	Asistencia personalizada y sin atención 24h	SopORTE estándar de Wrike	Email	24/7 y Live Chat Support		Advanced	Asistencia al cliente prioritaria	SopORTE estándar de Zoho Sprints
SEGURIDAD DE LA INFORMACIÓN										
Grupos de roles y permisos	✓	✓	✓	✓	✓	✗		✓	✗	✗
Autenticación en dos factores	✓	✗ (Requiere Atlassian Access)	✓	✗	✗	✓		✗	✓	✓
FUNCIONALIDADES										
Vista tablero Kanban	✓	✓	✓	✓	✓	✓		✓	✓	✓
Vista cronograma del proyecto	✓	✓	✓	✓	✓	✓		✓	✓	✓
Vista calendario	✓	✗	✓	✓	✗	✓		✓	✓	✗
Dependencias	✗	✓	✓	✓	✓	✓		✓	✓	✗
Etiquetas	✓	✓	✓	✓	✓	✓		✓	✓	✓
Campos personalizados	✓	✓	✓	✓	✗	✓		✓	✓	✗
Comentarios	✓	✓	✓	✓	✓	✓		✓	✓	✓
Menciones a otros miembros	✓	✓	✓	✓	✗	✓		✓	✓	✓
Archivos adjuntos	✓	✓	✓	✓	✓	✓		✓	✓	✓
Generación de gráficos e informes	✓	✓	✓	✓	✓	✓		✓	✓	✓
Tableros modificables	✓	✓	✓	✓	✗	✓		✓	✓	✓
Plantillas de documentos	✗	✗	✗	✓	✗	✓		✗	✓	✗
Plantillas de proyecto	✓	✓	✓	✓	✗	✓		✓	✓	✓
Automatizaciones	✓	✓	✓	✓	✗	✓		✗	✓	✗
Integraciones	✓	✓	✓	✓	✓	✓		✓	✓	✓
Exportar e importar datos	✓	✓	✗	✓	✓	✓		✓	✓	✓

Tabla 6 - Comparación planes Cloud nivel 3

ESTUDIO Y COMPARACIÓN DE HERRAMIENTAS DE SOPORTE DE PROYECTOS ÁGILES

7.4.1.4 Nivel 4

Criterios	Planes									
	Trello - Enterprise	Jira - Enterprise	Asana - Enterprise	Wrike - Enterprise	TeaScrum - Business	ClickUp - Business Plus		Freecamp - Enterprise	Monday.com - Pro	Zoho Sprints - Premier
ASPECTOS GENERALES										
Precio por usuario y mes	17.50 \$ (fact. anual)	13.99 \$	No disponible	No disponible	385 \$ (para todos los equipos)	29 \$		19.99 \$	17.64 \$	6.60 \$
Licencia de uso	S. Propietario	S. Propietario	S. Propietario	S. Propietario	GNU/GPL	S. Propietario		S. Propietario	S. Propietario	S. Propietario
Cloud/On-Premise	Cloud	Cloud	Cloud	Cloud	Cloud	Cloud		Cloud	Cloud	Cloud
App escritorio	✓	✗	✓	✓	✗	✓		✓	✓	✗
App móviles	✓	✓	✓	✓	✓	✓		✓	✓	✓
Documentación	Alta	Alta	Alta	Alta	Media	Alta		Media	Alta	Alta
Usuarios destinados	Empresas medianas o grandes	Empresas grandes	Empresas grandes	Empresas medianas o grandes	Empresas medianas o grandes	Empresas medianas o grandes		Empresas grandes	Empresas medianas	Empresas medianas o grandes
Mínimo usuarios	50	801	A Consultar	5	No	No		A consultar	No	No
Máximo usuarios	No	No	No	No	No	No		No	No	No
Límite proyectos	Ilimitados	Ilimitados	Ilimitados	Ilimitados	Ilimitados	1000/espacio		Ilimitados	Ilimitados	Ilimitados
División en varios equipos	✓	✓	✓	✓	✓	✓		✓	✓	✓
Límite de equipos o espacios	Ilimitados	Ilimitados	Ilimitados	Ilimitados	Ilimitados	Ilimitados		Ilimitados	Ilimitados	Ilimitados
ALMACENAMIENTO Y SOPORTE										
Almacenamiento	250MB/file	Ilimitado	100MB/file	10GB/user	No especificado	Ilimitado		250MB/file	100GB/user	100GB/user
Suporte	24/7 Enterprise Admin	24/7 Enterprise	Asistencia personalizada y con atención 24h	Suporte estándar de Wrike	Email / Teléfono / Remoto	24/7, Live Chat y Priority Support		Lightning fast	Asistencia al cliente prioritaria	Suporte estándar de Zoho Sprints
SEGURIDAD DE LA INFORMACIÓN										
Grupos de roles y permisos	✓	✓	✓	✓	✓	✓		✓	✗	✓
Autenticación en dos factores	✓	✓	✓	✓	✗	✓		✓	✓	✓
FUNCIONALIDADES										
Vista tablero Kanban	✓	✓	✓	✓	✓	✓		✓	✓	✓
Vista cronograma del proyecto	✓	✓	✓	✓	✓	✓		✓	✓	✓
Vista calendario	✓	✗	✓	✓	✗	✓		✓	✓	✗
Dependencias	✗	✓	✓	✓	✓	✓		✓	✓	✗
Etiquetas	✓	✓	✓	✓	✓	✓		✓	✓	✓
Campos personalizados	✓	✓	✓	✓	✗	✓		✓	✓	✓
Comentarios	✓	✓	✓	✓	✓	✓		✓	✓	✓
Menciones a otros miembros	✓	✓	✓	✓	✗	✓		✓	✓	✓
Archivos adjuntos	✓	✓	✓	✓	✓	✓		✓	✓	✓
Generación de gráficos e informes	✓	✓	✓	✓	✓	✓		✓	✓	✓
Tableros modificables	✓	✓	✓	✓	✗	✓		✓	✓	✓
Plantillas de documentos	✗	✗	✗	✓	✗	✓		✗	✓	✗
Plantillas de proyecto	✓	✓	✓	✓	✗	✓		✓	✓	✓
Automatizaciones	✓	✓	✓	✓	✗	✓		✗	✓	✗
Integraciones	✓	✓	✓	✓	✓	✓		✓	✓	✓
Exportar e importar datos	✓	✓	✓	✓	✓	✓		✓	✓	✓

Tabla 7 - Comparación planes Cloud nivel 4

ESTUDIO Y COMPARACIÓN DE HERRAMIENTAS DE SOPORTE DE PROYECTOS ÁGILES

7.4.1.5 Nivel 5

Criterios	Planes									
	Wrike - Panacea		ClickUp - Enterprise		Monday.com - Corporativo					
ASPECTOS GENERALES										
Precio por usuario y mes				No disponible		No disponible			No disponible	
Licencia de uso				S. Propietario		S. Propietario			S. Propietario	
Cloud/On-Premise				Cloud		Cloud			Cloud	
App escritorio				✓		✓			✓	
App móviles				✓		✓			✓	
Documentación				Alta		Alta			Alta	
Usuarios destinados				Empresas grandes		Empresas grandes			Empresas grandes	
Mínimo usuarios				5		A Consultar			A consultar	
Máximo usuarios				No		No			No	
Límite proyectos				Ilimitados		Ilimitados			Ilimitados	
División en varios equipos				✓		✓			✓	
Límite de equipos o espacios				Ilimitados		Ilimitados			Ilimitados	
ALMACENAMIENTO Y SOPORTE										
Almacenamiento				15GB/user		Ilimitado			100GB/user	
SopORTE				SopORTE estándar de Wrike		24/7, Live Chat y Priority Support			SopORTE de primer nivel	
SEGURIDAD DE LA INFORMACIÓN										
Grupos de roles y permisos				✓		✓			✓	
Autenticación en dos factores				✓		✓			✓	
FUNCIONALIDADES										
Vista tablero Kanban				✓		✓			✓	
Vista cronograma del proyecto				✓		✓			✓	
Vista calendario				✓		✓			✓	
Dependencias				✓		✓			✓	
Etiquetas				✓		✓			✓	
Campos personalizados				✓		✓			✓	
Comentarios				✓		✓			✓	
Menciones a otros miembros				✓		✓			✓	
Archivos adjuntos				✓		✓			✓	
Generación de gráficos e informes				✓		✓			✓	
Tableros modificables				✓		✓			✓	
Plantillas de documentos				✓		✓			✓	
Plantillas de proyecto				✓		✓			✓	
Automatizaciones				✓		✓			✓	
Integraciones				✓		✓			✓	
Exportar e importar datos				✓		✓			✓	

Tabla 8 - Comparación planes Cloud nivel 5

ESTUDIO Y COMPARACIÓN DE HERRAMIENTAS DE SOPORTE DE PROYECTOS ÁGILES

7.4.2 Planes On-Premise

Criterios	Planes									
	Jira - Data Center	IceScrum - Core	IceScrum - Starter	IceScrum - Enterprise	IceScrum - Corporate	Tabgo - Self Managed	Tabgo - Automated Hosting	Tabgo - Managed for you		
ASPECTOS GENERALES										
Precio licencia anual	Desde 42.000 \$	0 \$	1093 \$	3302 \$	11038 \$	0 \$	A Consultar con Elestio	No disponible		
Licencia de uso	S. Propietario	GNU/GPL	GNU/GPL	GNU/GPL	GNU/GPL	GNU Affero GPL	GNU Affero GPL	GNU Affero GPL		
Cloud/On-Premise	On-Premise	On-Premise	On-Premise	On-Premise	On-Premise	On-Premise	On-Premise (gestionado por Elestio)	On-Premise		
Documentación	Alta	Media	Media	Media	Media	Media	Media	Media		
Usuarios destinados	Empresas grandes	Freelancers	Empresas pequeñas	Empresas medianas	Empresas grandes	Freelancers, empresas pequeñas	Empresas pequeñas o medianas	Empresas medianas o grandes		
Mínimo usuarios	500	No	No	No	No	No	No	No		
Máximo usuarios	No	No	No	No	No	No	No	No		
Límite proyectos	Ilimitados	Ilimitados	Ilimitados	Ilimitados	Ilimitados	Ilimitados	Ilimitados	Ilimitados		
División en varios equipos	✓	✓	✗	✓	✓	✗	✗	✗		
Límite de equipos o espacios	Ilimitados	Ilimitados	No Aplica	Ilimitados	Ilimitados	No Aplica	No Aplica	No Aplica		
ALMACENAMIENTO Y SOPORTE										
Almacenamiento	Autogestionado	Autogestionado	Autogestionado	Autogestionado	Autogestionado	Autogestionado	Autogestionado	Autogestionado		
SopORTE	24/7 Support	Foro	Email	Email / Teléfono / Remoto	Email / Teléfono / Remoto	SopORTE de la comunidad	SopORTE de la comunidad y de Elestio	SopORTE prioritario		
SEGURIDAD DE LA INFORMACIÓN										
Grupos de roles y permisos	✓	✗	✗	✓	✓	✓	✓	✓		
Autenticación	Autogestionado	Autogestionado	Autogestionado	Autogestionado	Autogestionado	Autogestionado	Autogestionado	Autogestionado		
FUNCIONALIDADES										
Vista tablero Kanban	✓	✓	✓	✓	✓	✓	✓	✓		
Vista cronograma del proyecto	✓	✓	✓	✓	✓	✗	✗	✗		
Vista calendario	✗	✗	✗	✗	✗	✗	✗	✗		
Dependencias	✓	✓	✓	✓	✓	✓	✓	✓		
Etiquetas	✓	✓	✓	✓	✓	✓	✓	✓		
Campos personalizados	✓	✗	✗	✗	✗	✓	✓	✓		
Comentarios	✓	✓	✓	✓	✓	✓	✓	✓		
Menciones a otros miembros	✓	✗	✗	✗	✗	✓	✓	✓		
Archivos adjuntos	✓	✓	✓	✓	✓	✓	✓	✓		
Generación de gráficos e informes	✓	✓	✓	✓	✓	✓	✓	✓		
Tableros modificables	✓	✗	✗	✗	✗	✓	✓	✓		
Plantillas de documentos	✗	✗	✗	✗	✗	✗	✗	✗		
Plantillas de proyecto	✓	✗	✗	✗	✗	✓	✓	✓		
Automatizaciones	✓	✗	✗	✗	✗	✗	✗	✗		
Integraciones	✓	✗	(limitado a 5)	✓	✓	✓	✓	✓		
Exportar e importar datos	✓	(exportar)	✓	✓	✓	✓	✓	✓		

Tabla 9 - Comparación planes On-Premise

8. DESARROLLO DE APLICACIÓN WEB

En el siguiente apartado, veremos el desarrollo de una aplicación web que tendrá como objetivo, mediante un cuestionario, ayudar a las organizaciones a decantarse por una herramienta de soporte de proyectos ágiles (y su plan concreto) u otra (de las estudiadas en el presente trabajo).

NOTA: Todo el código de la aplicación está documentado y se puede consultar toda la documentación Java generada en la carpeta “JavaDoc” del proyecto.

8.1 Resumen Técnico

Para realizar la aplicación se ha utilizado el IDE “**IntelliJ IDEA**” y el servidor web “**Apache Tomcat**” y los lenguajes: **Java** (versión 17, para la lógica de negocio), **HTML** (para la creación de las vistas), **JavaScript** (para el contenido interactivo de las páginas) y **CSS** (para el diseño gráfico del front-end). Además, se ha utilizado el framework web “**Spring**”, el framework “**Bootstrap**” para obtener mejores resultados en el front-end de la aplicación y el motor de plantillas Java “**Thymeleaf**” para mostrar la información de nuestro modelo en las vistas de una forma sencilla.

Las preguntas del cuestionario han sido diseñadas en base a la comparación de herramientas realizada anteriormente. Se han seleccionado aquellas características que se han considerado relevantes para elegir o descartar un plan u otro, mientras que aquellas características que no añaden un factor diferencial en la elección (como, por ejemplo, si incorporan un tablero Kanban o no, ya que todos los planes, desde los gratuitos, lo incluyen) se ha descartado su incorporación al cuestionario.

8.2 Creación del proyecto y estructura de directorios

En primer lugar, lo primero que se ha hecho ha sido generar el proyecto mediante la herramienta “**spring initializr**” a la cual se puede acceder desde el siguiente enlace: <https://start.spring.io/>. En dicha herramienta debemos seleccionar el tipo de proyecto (en nuestro caso Maven), el lenguaje (Java), la versión de Spring Boot (3.1.0), diversos metadatos del proyecto como la versión de Java (17 en nuestro caso) y, además, debemos añadir las dependencias que queremos utilizar (Spring Web y Thymeleaf). Una vez generado el proyecto, podemos abrirlo con un IDE de nuestra elección y empezar a generar los directorios que necesitaremos para nuestra aplicación.

Nuestra aplicación está compuesta por dos directorios diferenciados que cuelgan de /src/main, que son: “java” y “resources”. En el directorio “java” encontramos los directorios de: “modelo” (que incluye todas las clases de la lógica de nuestra aplicación: Cuestionario, Herramienta y Plan), “repositorios” (clases que guardan toda la información relativa a las herramientas y la información generada por el usuario en el cuestionario), “servicios” (encargados de acceder a los métodos de nuestros repositorios) y “controladores” (encargados de la comunicación entre las vistas y la lógica de nuestra aplicación). Además, dentro del directorio “java” encontramos la clase principal que es la encargada de iniciar la aplicación. Por otro lado, tenemos el directorio “resources”, en el que encontramos, por un lado, la carpeta “templates” con todas las vistas (páginas HTML) de la aplicación y la carpeta “static” por otro lado. Dicha carpeta “static” incluye tanto los ficheros CSS utilizados, como los archivos JS, como las imágenes de la aplicación.

Como vemos, nuestra aplicación sigue el patrón de diseño MVC (Modelo-Vista-Controlador) con el que buscamos una clara separación de las responsabilidades en el código (el modelo se

encarga de los datos y de la lógica de la aplicación, la vista de la representación visual de los datos y de la interacción con el usuario y el controlador que actúa como intermediario entre ambos).

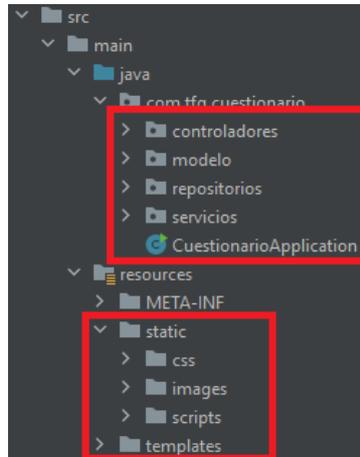


Figura 101 - Estructura de repositorios

8.3 Explicación detallada de los componentes del proyecto

Una vez explicada la estructura principal de nuestro proyecto (su esqueleto), procederemos a verla más en detalle:

- Comenzaremos con las clases de nuestro **modelo**. En primer lugar, la clase “Herramienta” tiene los atributos: “nombre”, “logo” y “sitioWeb”. El objetivo de esta clase es representar cada una de las herramientas que hemos estudiado en el presente trabajo y que están presentes en el cuestionario (por ejemplo: Jira, Trello, Asana, etc.). Por otro lado, la clase “Plan”, que está compuesta por los atributos: “herramienta”, “nombre”, “características”, “coincidencias” y “puntuación”, representa cada uno de los planes de los que está compuesto cada herramienta (por ejemplo: Trello-Free o Trello-Enterprise). Por último, la clase “Cuestionario” tiene los atributos: “respuestas” y “respuestas_texto” y representa el cuestionario que está siendo rellenado por el usuario. Los atributos de cada una de las clases los iremos viendo y entendiendo a medida que vayamos viendo el funcionamiento de la aplicación.
- Por otro lado, tenemos los **repositorios**. Encontramos un repositorio por cada una de las clases de nuestro modelo (es decir, tres repositorios en total). En los repositorios de “Herramienta” y “Plan” encontramos un ArrayList con todas las herramientas y planes respectivamente, para poder acceder a ellos (se ha hecho de este modo, descartando la conexión a una base de datos, ya que la aplicación no necesita compartir datos entre usuarios, sino que es un mero cuestionario). Además, encontramos los métodos para cargar las herramientas y los planes por los que está conformado el cuestionario en dichos Arrays. En el repositorio de planes, cargamos tanto la herramienta a la que pertenece, como su nombre, como sus características (algunas características son numéricas, otras booleanas y otras ArrayList, por lo que se guardan en un Hashmap de tipo <Integer,Object>) (todas las características han sido extraídas de las tablas comparativas establecidas en el apartado 7.4), dejando vacíos los atributos de “coincidencias” y “puntuación” que se rellenarán más adelante en base a lo introducido por el usuario. Por último, en el repositorio de “Cuestionario”, tenemos una instancia de dicha clase para poder acceder a sus datos cuando sea necesario.

- En tercer lugar, tenemos los **servicios**. También tenemos una clase servicio por cada clase del modelo. Cada una de estas clases tiene una instancia de su respectivo repositorio (por ejemplo, el servicio de Cuestionario guarda una instancia de “CuestionarioRepositorio”, para acceder a todos sus métodos). El objetivo de estas clases es el de proveer estos métodos de los repositorios a los controladores, para que estos no tengan acceso directo a estos.
- Y, por último, los **controladores**. Tenemos un controlador por cada una de las vistas principales de nuestro proyecto (tres en total: una para inicio, otra para cuestionario y otra para resultados). Los controladores, como hemos comentado, hacen uso de los servicios y acceden al modelo de la aplicación para proveer de datos a las vistas. A la vez que recogen los datos introducidos en las vistas por el usuario para que sean procesados por el modelo.
- Por otro lado, tenemos las **vistas**. Las vistas hacen uso del framework “Bootstrap” para mejorar el aspecto de nuestra interfaz de usuario y lograr, además, que sea “responsive”, es decir, que se adapte a diferentes dispositivos y tamaños de pantalla (esto se logra gracias a su estructura de rejilla conformada por filas y columnas que podemos apreciar en el código de las vistas). Por otro lado, hacen uso también del motor de plantillas “Thymeleaf” que nos proporciona una manera sencilla de recoger la información de los formularios o presentar los datos en las tablas (pudiendo insertar también reglas condicionales). Además, algunas vistas incorporan sus propias hojas de CSS para tocar ciertos aspectos del diseño y sus propios archivos de JavaScript para, por ejemplo, revisar que todas las preguntas han sido respondidas.

8.4 Implementación de las funcionalidades

Una vez visto más en detalle cómo está estructurada nuestra aplicación, veremos cómo se han implementado las funcionalidades más destacadas.

En primer lugar, comentaremos la **recogida de los datos del cuestionario**. Cuando el usuario rellena la primera pregunta que se le muestra (si requiere una herramienta en la nube u On-Premise), se le dirige hacia un conjunto de preguntas u otro dependiendo del tipo de herramienta que solicita (ya que los planes Cloud y On-Premise tienen preguntas diferentes debido a su naturaleza). Las preguntas del cuestionario están dentro de un formulario, el cual se envía una vez que se han rellenado todas las preguntas correctamente (un JavaScript comprueba esto último). Cada pregunta del formulario se envía como un elemento del HashMap “resultados”, de modo que, por ejemplo, la pregunta “i” se envía en el formulario como el campo “respuestas[i]”.

```
<div class="form-group mt-4">
  <label for="2">2. ¿Cuántas licencias (una por cada usuario) necesitaría contratar?</label>
  <input type="number" class="form-control mt-2 mx-4" style="width: 400px;" id="2" placeholder="Número de licencias" th:field="**{respuestas[2]}"/>
</div>
```

Figura 102 - Recogida de los datos introducidos en la pregunta 2

En segundo lugar, una vez recogidos los datos del cuestionario, el programa los procesa y realiza tres acciones: generar el ArrayList de “coincidencias” para cada plan, calcular su puntuación y generar las respuestas en texto que mostraremos al usuario en los resultados.

Para **generar el ArrayList de coincidencias**, se mira si se ha seleccionado un plan en la nube u On-Premise. En función de esto, se obtienen todos los planes de un tipo u otro y se calculan sus coincidencias para cada uno de ellos. El proceso del cálculo de este Array se realiza de la siguiente manera: si lo introducido por el usuario coincide con la característica del plan (por ejemplo, si el usuario solicita que tenga automatizaciones y el plan lo tiene) se añade un 1 al array. Por el contrario, si lo introducido no está en las características del plan se introduce un 0.

Hay que destacar que solo se comparan los atributos que solicita el usuario, es decir, si el usuario ha introducido en el cuestionario que no requiere de integraciones, ese atributo no se compara y, por lo tanto, no se tiene en cuenta para el cálculo de las puntuaciones (hay algunos atributos como el precio, que estarán siempre en la comparación porque no son preguntas de “Sí” o “No”).

Para el **cálculo de las puntuaciones de cada plan**, es tan sencillo como sumar todos los puntos que ha obtenido el plan en el Array de coincidencias y dividir esta suma de puntos entre el total que se pueden sacar. De esta forma obtendremos un número que, multiplicado por 100, nos dirá el porcentaje de coincidencia de ese plan con lo que está buscando el usuario. Hay que puntualizar que hay 2 preguntas que puntúan 4 puntos en vez de 1 en el caso de coincidir, debido a su importancia en la elección. Estas dos preguntas son: el tipo de usuarios al que está destinado el plan (ya que se quiere evitar que se recomiende un plan, por ejemplo, a un freelancer y que dicho plan esté enfocado a empresas grandes) y el número de usuarios o licencias requeridas (ya que si el usuario requiere 1000 licencias y el límite de dicho plan son 100, hay que penalizar con mayor severidad dicho aspecto que otros, ya que es menos salvable).

```
public void calcularPuntuacion(){
    double puntuacion_total_posible = coincidencias.size() - 2 + 8;
    double suma = 0;
    for (int i=0;i<coincidencias.size();i++){
        if (i==1 || i==2){
            if (coincidencias.get(i) == 1){
                suma = suma + 4;
            }
        }
        else{
            if (coincidencias.get(i) == 1){
                suma = suma + 1;
            }
        }
    }
    double puntuacionCalculada = (suma / puntuacion_total_posible) * 100;
    puntuacionCalculada = Math.round(puntuacionCalculada * 100) / 100d;
    this.puntuacion = puntuacionCalculada;
}
```

Figura 103 - Método encargado de calcular la puntuación de cada plan

Y para la **generación de las respuestas en texto**, que se mostrarán al usuario en los resultados, se rellena un ArrayList de Strings en función de lo solicitado por el usuario. Hay un total de 5 respuestas que están siempre presentes, que son: el tipo de plan, el tipo de usuario, el número de usuarios, el presupuesto y el número de proyectos simultáneos. Para cada una de estas cinco preguntas se genera un texto personalizado para la respuesta (por ejemplo, si el usuario ha introducido que tiene un presupuesto de 100\$, el texto generado es: “Presupuesto de 100\$”). Para el resto de las respuestas, se mira si son “true” (se requiere esa funcionalidad) o “false” (no se requiere). En caso de ser “true”, se añade el texto correspondiente a dicha característica (por ejemplo, si el usuario ha indicado que quiere que tenga aplicación de escritorio, se añade el texto de: “Con aplicación de escritorio”), y en caso de ser “false” no se añade nada al array.

```
respuestas_texto.add("Herramienta Cloud"); //0
respuestas_texto.add("Para "+respuestas.get(1)); //1
respuestas_texto.add(respuestas.get(2)+" licencias"); //2
respuestas_texto.add("Presupuesto de: "+respuestas.get(3)+" $"); //3
respuestas_texto.add("Para "+respuestas.get(4)+" proyectos simultáneos"); //4
if (respuestas.get(5).equals("true")){ respuestas_texto.add("Con aplicación de escritorio");} //5
if (respuestas.get(6).equals("true")){ respuestas_texto.add("Con aplicación para móviles");} //6
if (respuestas.get(7).equals("true")){ respuestas_texto.add("Open Source");} //7
```

Figura 104 - Extracto del método encargado de generar las respuestas en texto

Una vez tenemos calculado todo lo mencionado anteriormente, estamos listos para **presentarle los resultados al usuario**. Para ello le dirigimos a la vista “resultados” con el cuestionario realizado (que contiene las respuestas en texto generadas) y con los planes Cloud u On-Premise, según lo requerido (dichos planes contienen el array de 1s y 0s generado con las coincidencias de cada uno y su puntuación. Además, los planes se ordenan antes de enviarlos según su puntuación en orden de mayor a menor). En la vista “resultados”, primero se le muestran al usuario los 3 planes que han obtenido una mayor puntuación y posteriormente se le muestran el resto (ordenados de mayor a menor puntuación). De todos los planes, se muestra una tabla que justifica sus resultados. Dicha tabla está conformada por las respuestas en texto que hemos generado anteriormente y el array de 1s y 0s de cada plan con las coincidencias, de modo que, para cada respuesta, si el array de ese plan tiene un 1 en esa posición se sustituye por un “tick” verde haciendo referencia a que ese plan cumple con esa condición requerida. Y, por el contrario, si el array en esa posición tiene un 0, se sustituye por una cruz roja, haciendo referencia a que esa condición no la cumple el plan.

```
<table class="table table-hover">
  <thead>
    <tr>
      <th th:each="encabezado : ${cuestionario.respuestas_texto}" th:text="${encabezado}" style="..."></th>
      <th> Puntuación (% de coincidencia) </th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th scope="row"> </th>
      <td th:each="coincidencia : ${planes[0].coincidencias}">
        <span th:text="${coincidencia == 1 ? '&#9989;' : '&#10060;'}"></span>
      </td>
      <th scope="row" th:text="${planes[0].puntuacion} + ' %'"></th>
    </tr>
  </tbody>
</table>
```

Figura 105 - Tabla que muestra los resultados del mejor plan

9. CONCLUSIONES Y TRABAJO FUTURO

9.1 Conclusiones

Como conclusión del trabajo, hay que destacar que se han cumplido ampliamente los objetivos marcados al inicio de este: hemos entendido la idea y conceptos en los que están sustentados las metodologías ágiles y ligeras, viendo en qué situaciones es mejor aplicar uno u otro y cuáles son sus principales modelos. También hemos expuesto las necesidades de soporte a las que se exponen los proyectos ágiles y, por lo tanto, qué beneficios nos aporta el uso de estas herramientas.

Posteriormente, también se han estudiado diez herramientas seleccionadas. Durante dichos estudios, una de las prioridades ha sido mostrar en todo momento todas las posibilidades que nos ofrecen estas herramientas, pero mostrando también los puntos débiles de cada una (ya que, al fin y al cabo, uno de los objetivos es compararlas y eso se consigue no solo viendo sus bondades, sino también sus lados débiles).

Se ha realizado una comparación exhaustiva de todos los planes de las herramientas, en el que se ha intentado exponer la mayor cantidad de criterios posibles para que dicha comparación sea lo más completa y eficaz posible. Y, por último, se ha realizado la aplicación web para ayudar a seleccionar de entre las herramientas estudiadas. Se ha conseguido transformar de manera eficaz la comparación realizada, en un cuestionario fácil de realizar para el usuario, a la par que la aplicación canaliza de manera correcta dicho cuestionario en unos resultados intuitivos y precisos.

9.2 Trabajo futuro

A continuación, proponemos una serie de trabajo futuro con relación al presente TFG:

1. **Ampliar el número de herramientas estudiadas:** con el objetivo de ampliar el abanico de herramientas que están disponibles en la aplicación web, se puede ampliar el número de herramientas de soporte estudiadas, añadirles a la comparación e incorporarlas al cuestionario de la aplicación web.
2. **Incorporar material multimedia a la aplicación:** con el objetivo también de mejorar la aplicación web, se podría añadir material multimedia en cada herramienta (como, por ejemplo, demostraciones de las funcionalidades que se comparan en el cuestionario) para facilitar la decisión de elección del usuario.
3. **Buscar otras vías para facilitar la selección de herramientas de soporte:** se podrían buscar otras alternativas (al cuestionario implementado en el presente trabajo) que ayuden de igual forma a las organizaciones, a comparar y decantarse por una herramienta u otra. Una de estas alternativas podría ser realizar pruebas de concepto con cada herramienta, es decir, desarrollar un proyecto piloto (basado en las características y procesos de trabajo de una empresa en particular) con cada herramienta para evaluar su rendimiento y capacidades. De esta forma, si la organización se asemeja a las características y forma de trabajo del proyecto piloto probado, puede ver si encaja o no en su contexto (podrían desarrollarse tantos proyectos piloto como se quisiese, basados en diferentes criterios como el tamaño de la empresa o su forma de trabajo, para que la probabilidad de que las compañías encuentren su símil sea mayor).

10. ANEXOS

10.1 Manual de instalación de la aplicación web

Podemos instalar y probar la aplicación web siguiendo **dos opciones** diferentes:

Abrir el proyecto de la aplicación web (“cuestionario”) en un IDE y ejecutar el archivo principal del proyecto o la opción más simple y que se recomienda: **ejecutar el ejecutable .jar de la aplicación que se proporciona en el proyecto.**

1. Para la primera opción (ejecutar el proyecto en un IDE) será necesario tener instalado en el equipo Java (JDK y mínimo versión 17) (si se utiliza una versión de java superior, es posible que se tenga que cambiar en las propiedades del proyecto) y un servidor web que nos permita compilar y ejecutar el código (en este caso, para la programación de la aplicación se ha utilizado Apache Tomcat).

El proceso puede variar ligeramente según el IDE utilizado (por ejemplo, en algunos IDE es necesario configurar el servidor web utilizado o en el caso de IntelliJ IDEA lleva su propio servidor Tomcat embebido, por lo que no necesitaremos instalarlo). En nuestro caso, se ha utilizado el IDE IntelliJ IDEA, en el cual basta con abrir el proyecto, localizar la clase principal (CuestionarioApplication.java) y ejecutarla (teniendo JDK 17 mínimo instalado en el equipo). Una vez que se haya iniciado la aplicación, podremos utilizarla escribiendo en un navegador web: **“localhost:8080”** para acceder al puerto 8080 de nuestro equipo (la aplicación nos redirigirá automáticamente a la vista de inicio).

A continuación, se muestran capturas del proceso de ejecución utilizando IntelliJ IDEA:

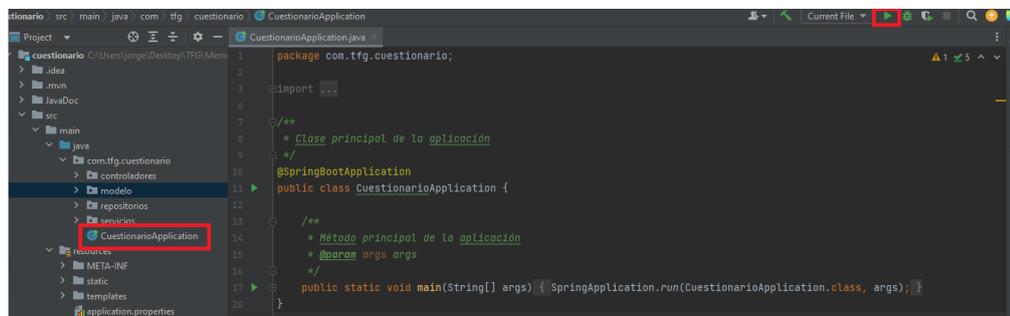


Figura 106 - Ejecución de la clase principal del programa

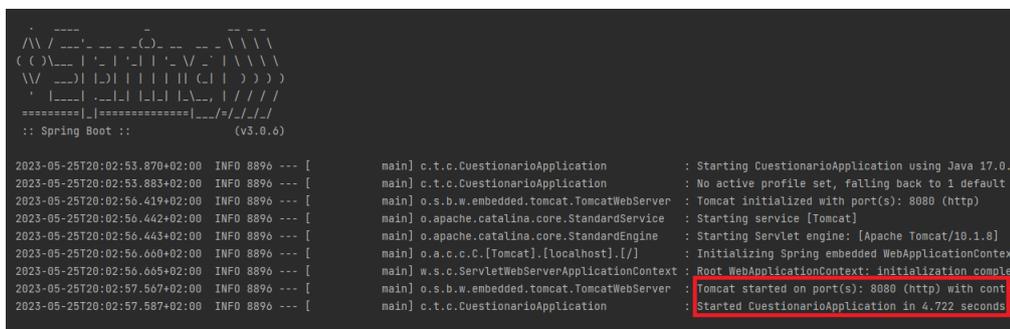


Figura 107 - Aplicación iniciada y lista para usar



Figura 108 - "localhost:8080" en un navegador para usar la aplicación

- Para la segunda opción y más recomendable al ser la más sencilla, solo necesitaremos tener instalado en nuestro equipo una versión de Java superior a la 17 (no necesitaremos tener instalado ningún servidor web).

Para cerciorarnos de la versión Java que se tiene instalada en el equipo (este procedimiento de comprobación también sirve para la opción 1, aunque para esa opción debemos asegurarnos de que sea el JDK lo que tenemos instalado) podemos escribir en un terminal de comandos: “**java -version**”, con ello comprobaremos la versión instalada en el equipo.

```
C:\WINDOWS\system32>java -version
java version "17.0.5" 2022-10-18 LTS
Java(TM) SE Runtime Environment (build 17.0.5+9-LTS-191)
Java HotSpot(TM) 64-Bit Server VM (build 17.0.5+9-LTS-191, mixed mode, sharing)
```

Figura 109 - Comprobación de la versión Java instalada

(Es probable que no tengamos configurado Java de manera correcta en nuestro equipo si no nos aparece un mensaje como el de la figura anterior. Debemos asegurarnos de tener bien configuradas las variables de entorno del sistema)

Una vez comprobada la versión de Java instalada, deberemos abrir un terminal de comandos (CMD o similar) y situarnos en la carpeta “target” de nuestro proyecto (para ello podemos copiar la ruta de dicha carpeta y ejecutar el comando: “**cd ruta**”). Una vez situados en “target” deberemos ejecutar el siguiente comando: “**java -jar cuestionario-0.0.1-SNAPSHOT.jar**”.

```
C:\Users\jorge\Desktop\TFG\Memoria\cuestionario\target>java -jar cuestionario-0.0.1-SNAPSHOT.jar
Spring
:: Spring Boot :: (v3.0.6)
2023-05-25T20:20:35.164+02:00 INFO 14188 --- [main] c.t.c.CuestionarioApplication : Starting CuestionarioApplication v0.0.1-SNAPSHOT using Java
7.0.5 with PID 14188 (C:\Users\jorge\Desktop\TFG\Memoria\cuestionario\target\cuestionario-0.0.1-SNAPSHOT.jar started by Jorge in C:\Users\jorge\Desktop\TFG\Memoria\cu
estionario\target)
2023-05-25T20:20:35.171+02:00 INFO 14188 --- [main] c.t.c.CuestionarioApplication : No active profile set, falling back to 1 default profile: "
default"
2023-05-25T20:20:37.639+02:00 INFO 14188 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (http)
2023-05-25T20:20:37.656+02:00 INFO 14188 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2023-05-25T20:20:37.657+02:00 INFO 14188 --- [main] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.8]
2023-05-25T20:20:37.889+02:00 INFO 14188 --- [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2023-05-25T20:20:37.894+02:00 INFO 14188 --- [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 256
ms
2023-05-25T20:20:38.866+02:00 INFO 14188 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context path ""
2023-05-25T20:20:38.901+02:00 INFO 14188 --- [main] c.t.c.CuestionarioApplication : Started CuestionarioApplication in 4.689 seconds (process r
unning for 5.718)
```

Figura 110 - Ejecución del archivo .jar proporcionado

Hecho esto, la aplicación se iniciará y podremos, al igual que en la opción 1, acceder al puerto 8080, escribiendo “**localhost:8080**” en un navegador web cualquiera (para detener la aplicación bastará con ejecutar control+c y detener el proceso).

10.2 Manual de usuario de la aplicación web

Al arrancar la aplicación web, el usuario puede visualizar una **página principal de bienvenida** en la que se muestran las 10 herramientas disponibles para comparar en el cuestionario (se muestra tanto el nombre como el logo de cada una). Además, el usuario puede acceder al sitio web oficial de cada herramienta si hace click en “Visitar sitio web” en la herramienta correspondiente (si se pasa el ratón por encima de la herramienta, se observa cómo el texto cambia del nombre de la herramienta a “Visitar sitio web” indicando al usuario dicha funcionalidad) (la página web de la herramienta se abrirá en una pestaña nueva).



Figura 111 - Página principal de bienvenida

Para **empezar el cuestionario**, el usuario debe hacer click en el botón situado en la parte inferior de la ventana “Realizar Cuestionario”. Una vez iniciado el cuestionario, empieza por una primera pregunta en la que se debe indicar si se quiere una herramienta Cloud u On-Premise, ya que los itinerarios de preguntas son diferentes.

Figura 112 - Pregunta inicial del cuestionario

Deberemos responder a la pregunta y pulsar en “Continuar”. Una vez definido el tipo de herramienta que queremos, se dirige al usuario a la **continuación del cuestionario**, en la que se muestran el resto de preguntas (un total de 24 preguntas si se ha seleccionado “Cloud” o 20 si se ha seleccionado “On-Premise”). El usuario debe responder a todas las preguntas: si no se rellenan todas las preguntas, el programa no dejará avanzar.

Rellena todas las siguientes cuestiones:

- ¿Cuál es el tamaño de su empresa?
 - Soy Freelancer
 - Empresa Pequeña (aproximadamente, hasta 50 empleados)
 - Empresa Mediana (aproximadamente, entre 50 y 250 empleados)
 - Empresa Grande (aproximadamente, más de 250 empleados)
- ¿Cuántas licencias (una por cada usuario) necesitaría contratar?
- ¿Qué presupuesto mensual tiene para todos los usuarios seleccionados (en USD)?
- En cuántos proyectos necesita trabajar a la vez su empresa?
- ¿Necesita que la herramienta disponga de aplicación de escritorio?

Figura 113 - Resto de preguntas del cuestionario

Encontramos tres tipos de preguntas: de opción múltiple, de introducción de un valor numérico y de “Sí / No”. Además, las preguntas 2, 3 y 4 de ambos itinerarios (las preguntas de introducción de un valor numérico) tienen las siguientes restricciones de entrada:

1. Para la pregunta 2, se debe introducir un número entero mayor que cero.
2. Para la pregunta 3, se debe introducir un número entero mayor o igual que cero.
3. Para la pregunta 4, se debe introducir un número entero mayor que cero.

Soy Freelancer
 Empresa Pequeña (aproximadamente, hasta 50 empleados)
 Empresa Mediana (aproximadamente, entre 50 y 250 empleados)
 Empresa Grande (aproximadamente, más de 250 empleados)

Cuántas licencias:

Qué presupuesto mensual tiene para todos los usuarios seleccionados (en USD)?

En cuántos proyectos necesita trabajar a la vez su empresa?

Figura 114 – Aviso de error si las preguntas no se rellenan correctamente

Una vez rellenas todas las preguntas correctamente, el usuario debe hacer click en el botón situado en la parte inferior de la ventana “Finalizar y ver resultados” que le llevará a la siguiente y última pantalla.

En la última pantalla, el usuario podrá consultar los **resultados de su cuestionario**. Se le mostrarán las 3 herramientas que mayor puntuación han obtenido (entendiendo por puntuación al porcentaje de coincidencia entre lo solicitado y lo que entrega dicha herramienta) junto con

una tabla en la que se muestra qué aspectos de todo lo que se ha solicitado (las características que no se solicitan se descartan y no se muestran) cumple dicho plan y qué aspectos no los cumple. Si por ejemplo, el usuario ha introducido que quiere que el plan tenga aplicación de escritorio, se mostrará con un tick verde si lo tiene o una cruz roja si no lo tiene.

1º

Monday.com - Estándar



Herramienta Cloud	Para Empresa_Pequeña	40 licencias	Presupuesto de: 700 \$	Para 100 proyectos simultáneos	Con aplicación de escritorio	Open Source	Buen almacenamiento	Vista de cronograma	Vista de calendario	Depender entre tar
✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓

Coincidencia: 96.0 %

Figura 115 – Mejor plan para el cuestionario realizado de ejemplo

Además, en la parte inferior de la pantalla, se muestran los resultados que han obtenido el resto de herramientas junto con su respectiva puntuación. Hay que puntualizar que solo se muestran las herramientas del tipo que ha solicitado el usuario (si solicitó una herramienta Cloud, sólo se muestran herramientas Cloud).

Y estos han sido los resultados de las otras herramientas:

	Herramienta Cloud	Para Empresa_Pequeña	40 licencias	Presupuesto de: 700 \$	Para 100 proyectos simultáneos	Con aplicación de escritorio	Open Source	Buen almacenamiento	Vista de cronograma	Vista de calendario	Dependencias entre tareas
Asana - Premium	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓
Wrike - Business	✓	✓	✓	✗	✓	✓	✗	✗	✓	✓	✓
Jira - Standard	✓	✓	✓	✓	✓	✗	✗	✓	✓	✗	✓
Taiga - Basic	✓	✓	✓	✓	✓	✗	✓	✓	✗	✗	✓
Taiga - Premium	✓	✓	✓	✓	✓	✗	✓	✓	✗	✗	✓
Wrike -	✓	✓	✗	✓	✓	✓	✗	✗	✓	✓	✓

Figura 116 - Resultados del resto de planes

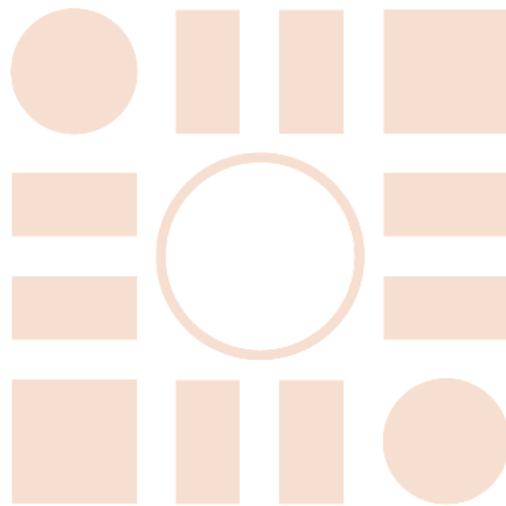
Una vez se han observado los resultados, **se puede volver a la página principal de bienvenida y realizar de nuevo el cuestionario** (los datos del cuestionario anterior se resetean), pulsando en el botón “Volver al Menú de Inicio”.

11. REFERENCIAS

- [1] J. Cervantes Gómez y M. d. C. Gómez Fuentes, «Taxonomía de los modelos y metodologías de desarrollo software más utilizados,» *Universidades*, vol. 62, n° 52, pp. 37-47, 2012.
- [2] R. G. Figueroa, C. J. Solís y A. A. Cabrera, «Metodologías tradicionales vs. metodologías ágiles,» *Universidad Técnica Particular de Loja, Escuela de Ciencias de la Computación*, vol. 9, n° 1, pp. 1-10, 2008.
- [3] C. Gil Aros, « Metodología en los sistemas y aplicaciones basadas en la web,» *Avances: Investigacion en Ingeniería*, vol. 1, n° 8, pp. 83-87, 2008.
- [4] J. Zumba Gamboa, «Evolución de las Metodologías y Modelos utilizados en el Desarrollo de Software,» *INNOVA Research Journal*, vol. 3, n° 10, pp. 20-33, 2018.
- [5] E. Herrera Uribe y L. E. Valencia Ayala, «Del manifiesto ágil sus valores y principios,» *Scientia et technica*, vol. 13, n° 34, pp. 381-386, 2007.
- [6] PMI, «Pulse of the Profession,» 2017.
- [7] A. Navarro Cadavid, J. D. Fernández Martínez y J. Morales Vélez, «Revisión de metodologías ágiles para el desarrollo de software,» *Prospectiva*, vol. 11, n° 2, pp. 30-39, 2013.
- [8] A. SHRIVASTAVA, « A systematic review on extreme programming,» *Journal of Physics: Conference Series*, vol. 1969, n° 1, p. 012046, 2021.
- [9] T. Chow y D.-B. Cao, «A survey study of critical success factors in agile software projects,» *Journal of systems and software*, vol. 81, n° 6, pp. 961-971, 2008.
- [10] N. R. Darwish y N. M. Rizk, «Multi-dimensional success factors of agile software development projects,» *International Journal of Computer Applications*, vol. 118, n° 15, 2015.
- [11] C. d. I. p. Wikimedia, «Trello - Wikipedia, la enciclopedia libre,» 14 07 2016. [En línea]. Available: <https://es.wikipedia.org/wiki/Trello>. [Último acceso: 25 03 2023].
- [12] «Manage Your Team's Projects From Anywhere | Trello,» [En línea]. Available: <https://trello.com/es>. [Último acceso: 25 03 2023].
- [13] C. d. I. p. Wikimedia, «Jira - Wikipedia, la enciclopedia libre,» 20 03 2010. [En línea]. Available: <https://es.wikipedia.org/wiki/Jira>. [Último acceso: 26 03 2023].
- [14] «Jira | Software de seguimiento de proyectos e incidencias,» [En línea]. Available: <https://www.atlassian.com/es/software/jira>. [Último acceso: 26 03 2023].
- [15] C. d. I. p. Wikimedia, «Asana (software) - Wikipedia, la enciclopedia libre,» 27 10 2013. [En línea]. Available: [https://es.wikipedia.org/wiki/Asana_\(software\)](https://es.wikipedia.org/wiki/Asana_(software)). [Último acceso: 27 03 2023].
- [16] «Gestiona en línea el trabajo, los proyectos y las tareas de tu equipo • Asana,» [En línea]. Available: <https://asana.com/es>. [Último acceso: 27 03 2023].
- [17] C. d. I. p. Wikimedia, «Wrike - Wikipedia, la enciclopedia libre,» 12 03 2013. [En línea]. Available: <https://es.wikipedia.org/wiki/Wrike>. [Último acceso: 29 03 2023].
- [18] «Software de gestión de proyectos versátil y eficaz | Wrike,» [En línea]. Available: <https://www.wrike.com/es/>. [Último acceso: 29 03 2023].
- [19] «iceScrum,» [En línea]. Available: <https://www.icescrum.com/es/>. [Último acceso: 31 03 2023].
- [20] «Características, historias y tareas – iceScrum,» [En línea]. Available: <https://www.icescrum.com/es/documentation/features-stories-tasks/>. [Último acceso: 31 03 2023].
- [21] «ClickUp | One app to replace them all,» [En línea]. Available: <https://clickup.com>. [Último acceso: 03 04 2023].

-
- [22] «Taiga: Tu herramienta de gestión de proyectos ágil y opensource: Kanban & Scrum,» [En línea]. Available: <https://www.taiga.io/es>. [Último acceso: 05 04 2023].
- [23] «Taiga Documentation,» [En línea]. Available: <https://docs.taiga.io>. [Último acceso: 05 04 2023].
- [24] «Free Project Management Software - | Freedcamp,» [En línea]. Available: <https://freedcamp.com>. [Último acceso: 06 04 2023].
- [25] «monday.com | Una nueva forma de trabajar,» [En línea]. Available: <https://monday.com/lang/es/>. [Último acceso: 08 04 2023].
- [26] C. d. l. p. Wikimedia, «monday.com - Wikipedia, la enciclopedia libre,» 01 07 2021. [En línea]. Available: <https://es.wikipedia.org/wiki/Monday.com>. [Último acceso: 08 04 2023].
- [27] <https://support.monday.com/hc/es>. [En línea].
- [28] C. d. l. p. Wikimedia, «Zoho Corporation - Wikipedia, la enciclopedia libre,» 04 03 2014. [En línea]. Available: https://es.wikipedia.org/wiki/Zoho_Corporation. [Último acceso: 09 04 2023].
- [29] «Online Agile Project Management Software | Zoho Sprints,» [En línea]. Available: <https://www.zoho.com/sprints/>. [Último acceso: 09 04 2023].
- [30] C. d. p. d. Wikimedia, «Freedcamp - Wikipedia,» 13 01 2015. [En línea]. Available: <https://en.wikipedia.org/wiki/Freedcamp>. [Último acceso: 06 04 2023].
- [31] C. d. l. p. Wikimedia, «Taiga (administrador de proyectos) - Wikipedia, la enciclopedia libre,» 24 11 2020. [En línea]. Available: [https://es.wikipedia.org/wiki/Taiga_\(administrador_de_proyectos\)](https://es.wikipedia.org/wiki/Taiga_(administrador_de_proyectos)). [Último acceso: 05 04 2023].
- [32] M. Taheri y S. M. Sadjadi, «A Feature-Based Tool-Selection Classification for Agile Software Development,» *SEKE*, pp. 700-704, 2015.
- [33] A. Mihalache, «Project management tools for agile teams,» *Informatica Economica*, vol. 21, n° 4, pp. 85-93, 2017.
- [34] M. Manole y M.-Ş. Avramescu, « Comparative Analysis of Agile Project Management Tools,» *Academy of Economic Studies. Economy Informatics*, vol. 17, n° 1, pp. 25-31, 2017.
- [35] D. Özkan y A. Mishra, « Agile Project Management Tools: A Brief Comprative View,» *Cybernetics and Information Technologies*, vol. 19, n° 4, pp. 17-25, 2019.

Universidad de Alcalá
Escuela Politécnica Superior



ESCUELA POLITECNICA
SUPERIOR



Universidad
de Alcalá