



Universidad  
de Alcalá

## Trabajo Fin de Master

# Métricas en proyectos ágiles. Caso práctico: Jira cloud

**Máster Universitario en Dirección de Proyectos Informáticos**

**Presentado por:**

D<sup>a</sup>. Marina García Garrote

**Dirigido por:**

D<sup>a</sup>. M. Carmen Pagés Arévalo

Alcalá de Henares, a 15 de septiembre de 2023

**UNIVERSIDAD DE ALCALÁ**

**Escuela Politécnica Superior**

**Máster en Dirección de Proyectos  
Informáticos**

Trabajo Fin de Máster

**Métricas en proyectos ágiles. Caso práctico:  
Jira cloud**

**Autor:** Marina García Garrote  
**Director:** M.Carmen Pagés Arévalo

**TRIBUNAL:**

**Presidente:**

**Vocal 1º:**

**Vocal 2º:**

**CALIFICACIÓN:**

**FECHA: 15/09/23**

García, Marina (2023). Métricas en proyectos ágiles. Caso práctico: Jira cloud. Trabajo de Fin de Máster. Madrid: Universidad de Alcalá de Henares.

## RESUMEN

El mundo del desarrollo de software ha experimentado una transformación significativa con la adopción de metodologías ágiles. Estas metodologías han revolucionado la manera en que los equipos enfrentan la planificación, el monitoreo y la entrega de proyectos, y han resaltado la necesidad de implementar métricas efectivas para evaluar el desempeño del equipo, la calidad del producto y la satisfacción del cliente. En este contexto, esta investigación se sumerge en el ámbito de las métricas en proyectos ágiles, explorando su importancia y aplicación en el entorno de desarrollo de software actual.

Al explorar el caso práctico de Jira Cloud, se espera realizar una evaluación que permita comprender mejor cómo las métricas pueden ser aprovechadas para el éxito en la gestión de proyectos ágiles, contribuyendo así a la excelencia en la entrega de proyectos de software en la era ágil.

Además, se brindará un nuevo enfoque en la elección de las métricas, desarrollando una herramienta que ayude a la elegir la que más se adecue en cada caso al proyecto. Se buscará de esta manera que con unas sencillas preguntas se pueda orientar a los equipos a, de manera efectiva, para recopilar y analizar estas métricas, mejorando así la eficiencia y precisión en la evaluación del rendimiento del equipo.

### Palabras clave:

Desarrollo de software, métrica, metodología ágil, Jira Cloud, gestión de proyectos, rendimiento, producto, calidad.

García, Marina (2023). Métricas en proyectos ágiles. Caso práctico: Jira cloud. Trabajo de Fin de Máster. Madrid: Universidad de Alcalá de Henares.

### **ABSTRACT**

The world of software development has undergone a significant transformation with the adoption of agile methodologies. These methodologies have revolutionized the way teams approach project planning, monitoring, and delivery, and have highlighted the need to implement effective metrics to evaluate team performance, product quality, and customer satisfaction. In this context, this research dives into the field of metrics in agile projects, exploring their importance and application in the current software development environment.

By exploring the Jira Cloud use case, we hope to conduct an assessment to better understand how metrics can be leveraged for success in agile project management, thereby contributing to excellence in software project delivery in the agile era.

In addition, a new approach will be provided in the choice of metrics, developing a tool that helps choose the one that best suits the project in each case. In this way, it will be sought that with a few simple questions the teams can be guided to, effectively, collect and analyze these metrics, thus improving efficiency and precision in evaluating the team's performance.

#### **Keywords:**

Software development, metrics, agile methodology, Jira Cloud, project management, performance, product, quality.

## AGRADECIMIENTOS

---

*A Mateo, por ser mi pilar.*

---

1.	Introducción .....	7
2.	Objetivos .....	8
3.	Estado del arte .....	9
3.1.	Gestión tradicional de proyectos .....	9
3.2.	Metodologías ágiles .....	10
3.3.	Métricas en metodologías ágiles.....	12
3.3.1.	Métricas básicas .....	12
4.	Nuevas métricas aplicadas en la actualidad.....	15
4.1.	Métricas asociadas a la estimación .....	18
4.2.	Métricas asociadas a la implementación de tareas .....	19
4.3.	Métricas asociadas a la calidad .....	20
4.4.	Métricas DORA .....	21
4.5.	Conclusiones.....	24
5.	Herramientas de soporte a la gestión ágil y métricas que proporcionan.....	25
5.1.	Análisis comparativo .....	42
6.	Jira Cloud .....	44
6.1.1.	Diferencias con Jira .....	45
6.1.2.	Jira Cloud y Metodologías Ágiles.....	46
6.1.3.	Métricas en Jira Cloud .....	48
7.	Desarrollo en Jira.....	49
7.1.	Proyecto en Jira Cloud.....	49
7.2.	Métricas de proyecto en Jira Cloud.....	55
8.	Desarrollo de herramienta para elección de métricas.....	66
8.1.	Elección preguntas para el desarrollo de la herramienta .....	66
8.2.	Desarrollo .....	70
9.	Resultados .....	84
10.	Conclusiones .....	85
10.1.	Limitaciones del estudio.....	86
10.2.	Trabajos futuros .....	87
11.	Referencias bibliográficas .....	88

## 1. Introducción

En el ámbito del desarrollo de software, la metodología ágil ha revolucionado la forma en que los equipos abordan la planificación, el seguimiento y la entrega de proyectos. La naturaleza dinámica y adaptativa de las metodologías ágiles exige la implementación de métricas efectivas que permitan evaluar el desempeño del equipo, la calidad del producto y la satisfacción del cliente. Este enfoque centrado en la entrega de valor continuo ha llevado al surgimiento de herramientas tecnológicas avanzadas que facilitan la gestión ágil de proyectos, y entre ellas destaca Jira Cloud.

Este estudio pretende detallar el estado actual del mundo de las métricas en proyectos ágiles y cómo influyen en el éxito de la gestión de proyectos. Además, se realizará una evaluación de éstas en un caso práctico específico: un proyecto planificado con Jira Cloud. Se pretende comprobar cómo las métricas desempeñan un papel fundamental en la evaluación del rendimiento del equipo, la planificación efectiva y la mejora continua en entornos ágiles. Se evaluarán además distintas herramientas de soporte a la gestión ágil, plataformas líderes en la gestión de proyectos y seguimiento de problemas, y como pueden ser aprovechadas para recopilar y analizar estas métricas de manera eficiente y precisa.

A través de este trabajo, se pretende mostrar cómo las métricas en proyectos ágiles no solo permiten una comprensión más profunda del proceso de desarrollo, sino que también brindan una base sólida para la toma de decisiones informadas y la optimización constante.

## 2. Objetivos

Los objetivos, que se van a materializar en los puntos principales del estudio son los siguientes:

- Evaluación del estado del arte.
  - Dentro del estado del arte se evaluará como principal objetivo la situación actual en cuanto a métricas en gestión de proyectos mediante metodologías ágiles.
- Análisis y comparativa de las principales herramientas de soporte a la gestión ágil que existen en el mercado, centrándose en las métricas que ofrece cada una.
- Estudio de la herramienta Jira Cloud.
  - Realización de un caso de uso con la herramienta Jira Cloud que permita una evaluación de las métricas que ofrece ésta, incluyendo las conclusiones que se pueden obtener de su aplicación.
- Herramienta de soporte a la elección de métricas, que permita una selección de las métricas más adecuadas en función de las prioridades del usuario.

### 3. Estado del arte

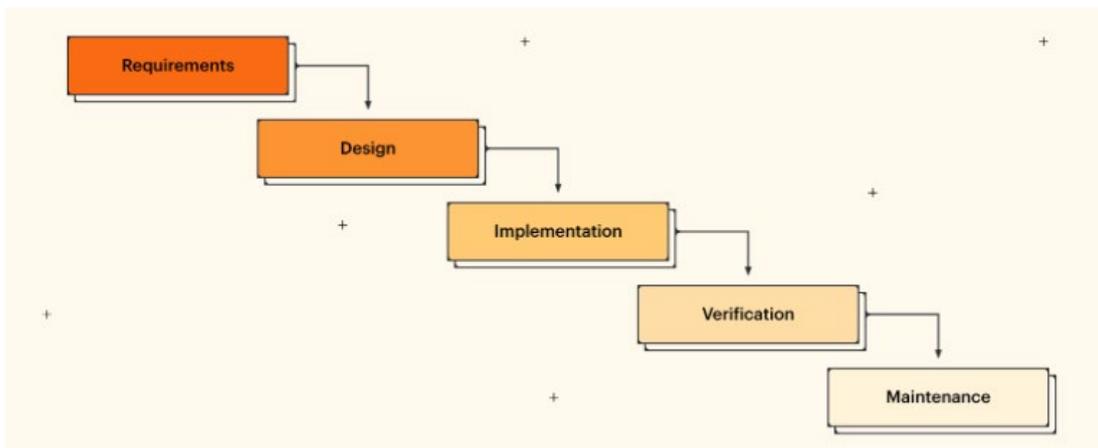
#### 3.1. Gestión tradicional de proyectos

La gestión tradicional de proyectos es una metodología donde los proyectos se ejecutan en un ciclo secuencial. Sigue una secuencia fija que incluyen principalmente los siguientes pasos:

- Iniciación.
- Planificación.
- Ejecución.
- Medición.

**Figura 1**

*Fases de la gestión tradicional de proyectos*



*Nota.* Lucidchart. (S.f.). What the waterfall project management methodology can (and can't) do for you. Lucidchart. [What the Waterfall Project Management Methodology Can Do for You | Lucidchart](#)

El enfoque de la gestión de proyectos tradicional pone especial énfasis en los procesos lineales, la documentación, la planificación por adelantado y la priorización. Con el método tradicional, el tiempo y el presupuesto son variables y los requisitos son fijos. Debido a esto a menudo existen problemas en cuanto a presupuesto y plazos de ejecución. Para cada uno de los pasos hay herramientas y técnicas definidas por el estándar que marca la metodología PMBOK (Project Management Institute, 2021).

También incluye otras metodologías como PRINCE2, común en organizaciones internacionales.

Entre los beneficios de la metodología tradicional se incluyen:

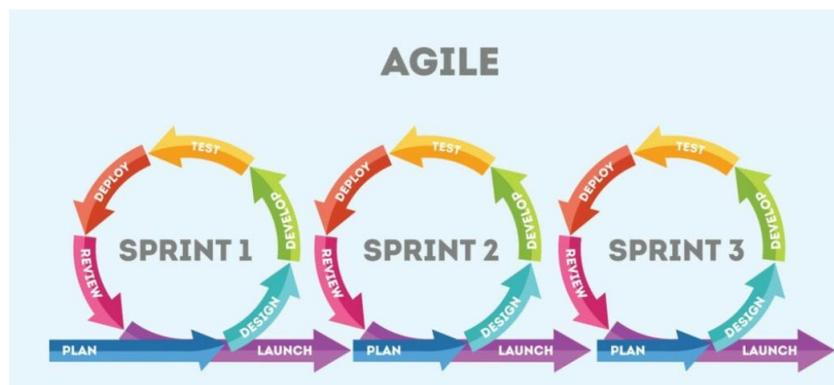
- Objetivos claramente definidos.
- Procesos controlables.
- Documentación clara.
- Mayor responsabilidad.

### 3.2. Metodologías ágiles

Las metodologías ágiles se han convertido en enfoques populares para el desarrollo de software y la gestión de proyectos en los últimos años. A diferencia de las metodologías tradicionales, que se basan en planes detallados y rigidez, las metodologías ágiles se centran en la colaboración, la adaptabilidad y la entrega incremental de productos o servicios de alta calidad, fomentando la adaptabilidad y la retroalimentación continua. Algunas metodologías ágiles comunes incluyen Scrum, Kanban, Extreme Programming (XP) y SAFe.

**Figura 2**

*Proceso de desarrollo según metodologías ágiles*



*Nota.* Toggle Team. (2022). Agile Project Management In Marketing: A Complete Guide. Togggl blog. [Agile Project Management In Marketing: A Complete Guide \(togggl.com\)](https://togggl.com)

Existe un manifiesto ágil y tiene cuatro valores importantes:

- Mayor enfoque en individuos e interacciones que en procesos y herramientas.
- El software funcionando es más importante que una documentación extensa.
- La colaboración con el cliente es más importante que la negociación contractual.

- Responder al cambio en lugar de seguir ciegamente un plan preestablecido.

Entre los beneficios más importantes de la gestión ágil de proyectos se encuentran:

- Se establecen prioridades flexibles.
- Se empieza a entregar antes.
- Costes y plazos conocidos.
- Mejora la calidad final.
- Mayor transparencia.

**Figura 3**

*Beneficios para el negocio al aplicar metodologías ágiles*



*Nota.* What Is Scaled Agile Framework: 9 Principles. (S.f.). Atc Blog. [Learn 9 Principles Of Scaled Agile Framework and Benefits \(american-technology.net\)](https://www.atc.com/blog/learn-9-principles-of-scaled-agile-framework-and-benefits)

Para poder obtener todos estos beneficios, es necesario obtener información de los datos, y conocimiento de esta información. Es por tanto imprescindible la obtención y evaluación de métricas dentro del ámbito del desarrollo del proyecto mediante metodologías ágiles.

### 3.3. Métricas en metodologías ágiles

Las métricas en Agile son herramientas utilizadas para medir y evaluar el desempeño de los equipos y proyectos ágiles. Estas métricas proporcionan datos cuantitativos que permiten a los equipos y las organizaciones comprender su rendimiento, identificar áreas de mejora y tomar decisiones informadas.

Las métricas ágiles continúan evolucionando para adaptarse a las necesidades cambiantes de los equipos y las organizaciones. Algunas tendencias emergentes incluyen el enfoque en métricas cualitativas, como la felicidad del equipo, y la integración de herramientas y técnicas de análisis de datos para una mejor visualización y seguimiento del rendimiento.

Sin embargo, también hay desafíos en la implementación efectiva de métricas en Agile. Estos desafíos incluyen la selección adecuada de métricas relevantes para cada contexto, evitar la manipulación de métricas por parte de los equipos y garantizar que las métricas se utilicen como herramientas para la mejora continua en lugar de generar presión o juicio negativo.

“En conjunto, las métricas tienen como objetivo la planificación de sprints, el proceso de reparación del software, la estimación del esfuerzo, el rendimiento del desarrollo, y los defectos de software se pueden utilizar para medir el rendimiento de los procesos de una organización” (Choraś et al. ,2020).

#### 3.3.1. Métricas básicas

Las métricas básicas de que pueden generarse en metodologías ágiles más relevantes en la actualidad son las siguientes:

- **Velocity** (Velocidad): Es la métrica más común y ampliamente utilizada en Agile y se refiere a la cantidad de trabajo completado por el equipo en cada iteración o sprint. De manera general se mide en puntos de historia, aunque puede medirse en otras unidades de medida definidas por el equipo. La velocidad se utiliza para estimar la capacidad de entrega futura y planificar las iteraciones posteriores. De esta manera se pretende establecer expectativas realistas en cuanto a las entregas futuras.

Una buena velocidad en el proyecto debería ser una creciente, es decir, más puntos de historia logrados en cada iteración. Si se incrementa la velocidad significa que el equipo se está sintiendo cada vez más cómodo con el proyecto. Si la velocidad reduce puede deberse a algunos problemas técnicos o malentendidos en el proyecto que deban analizarse en las retrospectivas.

- **Burn-down chart** (Gráfico de avance): El gráfico de avance es una herramienta visual que muestra la cantidad de trabajo restante en comparación con el tiempo. Proporciona una representación clara del progreso del equipo durante un sprint y ayuda a los miembros del equipo, los Scrum Masters y los stakeholders a entender y comunicar el estado del proyecto. El gráfico muestra una línea de referencia que indica el progreso esperado y otra línea que representa el trabajo real completado. Cualquier desviación entre estas dos líneas puede ayudar a identificar posibles problemas o retrasos.
- **Lead Time** (Tiempo de entrega): El lead time es una métrica que mide el tiempo transcurrido desde que se inicia una solicitud o requerimiento hasta que se entrega al cliente o se pone en producción. Es una medida importante de la eficiencia y la rapidez del equipo en la entrega de valor al cliente. Un lead time corto indica una mayor agilidad y capacidad de respuesta del equipo.
- **Cycle Time** (Tiempo de ciclo): Es similar al lead time, pero se enfoca en el tiempo que un elemento o tarea pasa activamente en desarrollo. Mide el tiempo que transcurre desde que se inicia el trabajo en una tarea hasta que se completa. El ciclo time ayuda a identificar cuellos de botella y a mejorar la eficiencia del proceso de desarrollo. Un ciclo time más corto puede indicar una mayor eficiencia y capacidad de entrega del equipo.
- **Defect Density** (Densidad de defectos): La densidad de defectos es una métrica que mide la cantidad de defectos o errores encontrados en el software en relación con una métrica específica, como líneas de código o funciones. Esta métrica proporciona información sobre la calidad del software producido por el equipo ágil. Una baja densidad de defectos indica una mayor calidad y madurez del producto.

- **Customer Satisfaction** (Satisfacción del cliente): La satisfacción del cliente es una métrica subjetiva pero crucial en Agile. Se refiere al nivel de satisfacción general del cliente con el producto o servicio entregado por el equipo ágil. Puede obtenerse a través de encuestas, retroalimentación directa o métricas de satisfacción definidas por la organización. La satisfacción del cliente es un indicador importante de la calidad y el valor entregado por el equipo.
- **Retrospective Actions Completed** (Acciones de retrospectiva completadas): Las reuniones de retrospectiva son una parte integral de las metodologías ágiles. Durante estas reuniones, los equipos identifican acciones de mejora para abordar los desafíos y los problemas encontrados durante el sprint anterior. La métrica de acciones de retrospectiva completadas rastrea cuántas de estas acciones se han implementado con éxito en el sprint. Esta métrica ayuda a evaluar la capacidad del equipo para aprender de la retroalimentación y mejorar continuamente su rendimiento.
- **Test Coverage** (Cobertura de pruebas): La cobertura de pruebas es una métrica que mide el porcentaje de código o funcionalidades cubiertas por pruebas automatizadas. Una alta cobertura de pruebas indica una mayor confianza en la calidad del software y ayuda a identificar áreas con una cobertura insuficiente. La cobertura de pruebas es especialmente importante en Agile, donde la entrega frecuente y la capacidad de respuesta rápida requieren una sólida base de pruebas automatizadas.
- **Burn-up chart** (o gráfico de evolución): Se trata de una métrica que rastrea el progreso hacia la finalización del proyecto. En la forma más simple del gráfico hay dos líneas: una línea de trabajo total (la línea de alcance del proyecto), y una línea de trabajo terminado. En el eje vertical se sitúa la cantidad de trabajo, que se puede medir en unidades personalizadas para cada proyecto. El eje horizontal es el tiempo, medido en iteraciones.

En cada iteración puede ver la cantidad de trabajo completado y la cantidad total de trabajo. La distancia entre las dos líneas es, por lo tanto, la cantidad de trabajo restante. Cuando las dos líneas se encuentran, el proyecto estará completo.

## 4. Nuevas métricas aplicadas en la actualidad

Las métricas específicas para Agile está evolucionando constantemente y se están dando a conocer nuevas métricas en los últimos años.

Hay muchas dimensiones diferentes para medir el desempeño de los proyectos ágiles: organizacionales, de proceso, de proyecto, equipo e individuales. (Ching Beh et al., 2022).

Según este texto además se pueden dividir las métricas de software en métricas de control y métricas predictivas, las primeras asociadas a los procesos y las segundas asociadas al propio software. Esta división permite por un lado aprender y mejorar la calidad del producto de software en funcionamiento aunque la medición de estos procesos suponga un reto por ser procesos complejos e intangibles.

Ching Beh et al. (2022) se centra en su estudio en la aplicación de métricas sobre pequeñas y medianas empresas que es donde las metodologías ágiles tienen más cabida dado que se trata en su mayoría de proyectos que comienzan y finalizan rápidamente. Choraś et al. (2020) también recalca la importancia de estas métricas en las pequeñas y medianas empresas que aplican metodologías ágiles dado que su generación y evaluación puede llevar a beneficios como la reducción del tiempo de comercialización de un producto, el ahorro de costes de desarrollo y una mayor satisfacción del cliente.

En este estudio se establecen de manera empírica veinticinco métricas categorizadas en seis grupos (métricas generales, métricas de estimación de tareas, métricas de implementación de tareas, métricas de corrección de errores, métricas de prueba y otras métricas) definidas con un valor medible.

**Figura 4***Métricas candidatas para la aplicación CONTRA*

No	Category	Metric	Value
1	General	Number of development tasks	[number]
2	General	Number of completed (closed) development tasks	[number]
3	General	Number of incomplete (open) development tasks	[%] [number]
4	General	Average number of development tasks	[number]
5	General	Average time of development task in the project board – from the moment it was added, to the moment it was closed	[number/time]
6	General	Average time needed to resolve an issue	[number/time]
7	Task estimation	Effort estimation accuracy for development tasks	[%]
8	Task estimation	Average difference between estimated effort (“estimated” attribute) and real effort (“spend” attribute)	[number/time]
9	Task estimation	Number of development tasks without estimation of effort (“estimated”)	[%] [number]
10	Task estimation	Number of development tasks without real effort (“spend”)	[%] [number]
11	Task estimation	Total sum of estimated effort values (“estimated”)	[number]
12	Task estimation	Sum of effort actually spent (“spend”)	[number]
13	Task implementation	Average task implementation time based on project board	[number/time]
14	Task implementation	Average time-to-implementation of task based on project board	[number/time]
15	Task implementation	Number of tasks with unassigned “Milestone” (sprint)	[%] [number]
16	Task implementation	Number of tasks not yet assigned to any developer	[%] [number]
17	Task implementation	Number of ongoing development tasks not belonging to the current sprint	[%] [number]
18	Bug fixing	Number of development tasks with reported bugs	[%] [number]
19	Bug fixing	Average time of task correction based on project board	[number/time]
20	Bug fixing	Average time-to-correct of task based on project board	[number/time]
21	Bug fixing	Percentage of ‘non-bug’ type tasks with respect to total tasks on the board	[%] [number]
22	Testing	Average time-to-test of development tasks	[number/time]
23	Testing	Average testing time based on project board	[number/time]
24	Testing	Percentage of tasks waiting for testing	[%] [number]
25	Others	Number of merge requests without discussion / comments during the code review	[%] [number]

*Nota.* Choraś et al. (2020).

Tras el estudio de estas métricas aplicadas al proyecto CONTRA (Choraś et al., 2020) se evalúan con el equipo de desarrollo y los demás perfiles implicados en la gestión ágil, cuales son las métricas que más información proveen que sea de utilidad para realizar la gestión del proyecto, concluyendo que son las siguientes:

- **Métrica 7: Precisión de la estimación por tareas de desarrollo (por desarrollador en el proyecto en un período de tiempo específico).** Se trata de una métrica va a permitir analizar el porcentaje de acierto de la asignación y el análisis del esfuerzo de cada desarrollador con respecto a la cantidad de trabajo que se estimó sería necesario para completar una tarea.
- **Métrica 9: Número de tareas de desarrollo sin estimación del esfuerzo por proyecto por desarrollador.** Esta métrica va a permitir analizar el número de tareas que constan aun sin estimación de esfuerzo. Esto es crítico dado que estimar el esfuerzo es vital para el proceso de desarrollo de software y primordial para ayudar a los equipos a garantizar que los desarrollos son entregados a tiempo, así como permitir la gestión de los recursos que se asignan a cada tarea.
- **Métrica 10: Número de tareas de desarrollo sin valor del esfuerzo empleado por proyecto por desarrollador.** Al igual que la métrica anterior, es importante la trazabilidad del esfuerzo realizado por un desarrollador en finalizar una tarea. Por este motivo es muy relevante llevar un control de las tareas que aun no tienen indicado un esfuerzo empleado, cuando el estado de la tarea indica que ésta se encuentra ya finalizada.
- **Métrica 11: Suma total de esfuerzo estimado por proyecto por desarrollador.** En el marco de mejora continua, esta métrica será más fiable cuanto más desarrollos se hayan estimado para este proyecto y desarrollador y llevará a un análisis completo de los recursos estimados.
- **Métrica 12: Suma del esfuerzo empleado por proyecto por desarrollador.** Esta métrica va a permitir evaluar de manera más eficiente la velocidad de desarrollo del equipo. Se puede evaluar comparada con la métrica anterior, para poder evaluar cual es el porcentaje de acierto del esfuerzo por desarrollador, comparando la estimación con la suma real.
- **Métrica 18: Número de tareas con errores reportados.** Como se verá adelante las métricas asociadas a indentificar errores tienen gran importancia (Włodarski, 2020) dado que una pronta indentificación de errores llevará a una rápida subsanación de los mismos y con ello a un resultado de mayor calidad.
- **Métrica 19: Tiempo promedio para probar la tarea de desarrollo.**
- **Métrica 20: Tiempo medio de corrección de la tarea según el tablero del proyecto.**
- **Métrica 21: Porcentaje de tareas de tipo 'Non-Bug' con respecto al total de tareas en el tablero.** Esta métrica ayudará a entender por un lado la cantidad de deuda técnica a la que hace frente el equipo, así como el tiempo eficiente de desarrollo invertido en tareas que no sean correcciones.

## 4.1. Métricas asociadas a la estimación

Uno de los puntos más importantes a tener en cuenta a la hora de concretar métricas de esfuerzo es el tamaño, como indica (Usman et al., 2017, citado en R. Méndez, 2018). En este aspecto la métrica más importante dentro del mundo de las metodologías ágiles son los puntos de historia. Junto con las métricas mostradas en el punto anterior indicadas en el estudio de Choraś et al. (2020), se incluyen además:

- **Puntos de historia (Story Points).** Un punto de historia es una medida que expresa de forma relativa el tamaño de una historia de usuario (user story) o funcionalidad (Coelho y Basu, 2012). Como viene indicado en el estudio de R. Méndez, E. (2018) este es un valor que depende de muchos factores, entre ellos: como la complejidad de desarrollo, el esfuerzo requerido, la repetición de tareas similares con anterioridad (Vige, W. 2022), o el riesgo inherente al desarrollo. Los puntos de historia son la medida de tamaño más utilizada en los calculos de estimación en metodologías ágiles (Usman y Britto, 2016). Esta estimación es de gran importancia dado que dará lugar a otras métricas que servirán para evaluar el progreso del proyecto y también a realizar mejores planificaciones a posteriori. Es por ello que se dedica una reunión prácticamente en exclusiva a su estimación, mayoritariamente utilizando el sistema de Planning Poker, popularizado por Cohn (2005) utilizando comunmente la serie Fibonacci (1, 2, 3, 5, 8, 13, 21, 34. . .) y T-shirt sizes (S, M, L, XL. . .). También se están realizando enfoques desde el punto de vista de la estimación mediante técnicas de Machine Learning.
- **Mean Magnitude Relative Error (MMRE).** Permite obtener la magnitud de error relativo de la estimación realizada (Malathi, 2012).

### Figura 5

*Fórmula MMRE*

$$MMRE_i = \frac{|EE_i - ER_i|}{ER_i}$$

MMRE es la métrica utilizada de manera genera en la literatura relacionada con la estimación de esfuerzo, pero en la práctica solo suele utilizarse para realizar una comparación entre los valores estimados y los valores reales.

- **Scope Creep (requisitos añadidos sin planificar).** Traducido literalmente como desplazamiento del alcance, hace referencia a la adición continua de nuevos requisitos o características a un proyecto sin una planificación o consideración adecuada. Monitorear los gráficos de velocidad puede ayudar a identificar la variación del alcance y su impacto en la capacidad del equipo para realizar el trabajo dentro del plazo planificado. Esta métrica idealmente debe tender a cero para garantizar una correcta gestión del alcance de los Sprints.

#### 4.2. Métricas asociadas a la implementación de tareas

Durante la realización del proyecto es importante mantener la trazabilidad de las tareas para poder ver la evolución de los entregables. Junto con la métrica de Frecuencia de despliegue dentro del bloque de métricas DORA que se detallará más adelante, y las métricas 13-17 indicadas por Choraś et al. (2020), destacan:

- **Diagrama de flujo acumulado.** El objetivo de esta métrica es evaluar el flujo de trabajo e identificar cuellos de botella en su proceso. Esto es especialmente útil para los equipos Kanban, ya que podría decirse que el flujo es la métrica más importante a considerar. Sin embargo, los equipos Scrum también pueden sufrir cuellos de botella. Esto hace que este informe sea útil para todos.

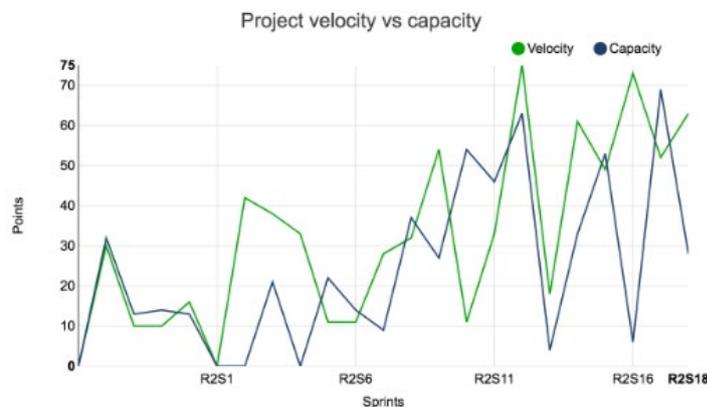
Un patrón que se puede detectar mediante esta métrica es el crecimiento incontrolado del backlog a lo largo del tiempo. Esto puede mostrar el hecho de que los propietarios de las tareas no cierren incidencias obsoletas o de que las incidencias con prioridad más baja no se traten nunca.

- **Gráfico de parking lot.** El gráfico de estacionamiento es un gráfico basado en funciones que resume el estado del proyecto mostrando el logro de cada una de sus funciones en porcentaje. Cada porcentaje representa el número de historias asociadas a una característica que se han logrado.
- **Tareas sin asignar.** Permite ver el avance del proyecto, pudiendo reflejar un número muy alto que las tareas no se desglosaron lo suficiente, o que va a desplazarse mucho trabajo al siguiente sprint.
- **Tareas bloqueadas.** Esta métrica permite ver las dependencias que se pueden tener de terceras partes, o de otras tareas, permitiendo poner el foco en resolver esos bloqueos.

- **Tareas fuera del Sprint actual.** Se trata de una métrica clave para detectar retrasos en el desarrollo, dado que da visibilidad a las tareas que se están arrastrando de Sprints anteriores.
- **Velocity variability.** Visualizado mediante el gráfico de velocidad vs. Velocidad planificada, La variabilidad de la velocidad mide la consistencia en la velocidad de un equipo. Indica la estabilidad del desempeño del equipo y ayuda a evaluar la confiabilidad de sus estimaciones y planificación.

**Figura 6**

*Gráfico de velocidad real vs velocidad planificada*



*Nota.* iceScrum. (S.f.). Indicators and reporting. iceScrum documentation. [Indicators and reporting – iceScrum](#)

- **Throughout (rendimiento).** Mide la cantidad promedio de elementos de trabajo procesados por unidad de tiempo. En un sistema Kanban, por ejemplo, como el trabajo se visualiza en tarjetas, el rendimiento se mide en función de cuántas tarjetas Kanban se terminaron en un período determinado (semanal, mensual, etc).
- **Velocity trend.** Esta métrica muestra un patrón o tendencia observada en la velocidad de un equipo a lo largo del tiempo. Ayuda a analizar el desempeño del equipo, identificar áreas de mejora y predecir su capacidad para futuros sprints.

### 4.3. Métricas asociadas a la calidad

Włodarski (2020) recalca tras estudiar de manera comparativa desarrollos realizados con metodologías ágiles y metodologías tradicionales que pese a que gestionar los proyectos con metodologías ágiles puede tener grandes beneficios, como la comunicación frecuente,

la responsabilidad conjunta o la autoorganización, también puede generarse una penalización en la calidad del producto. Esto es debido a que la entrega constante de valor al cliente y no tener la visión completa del objetivo como suele darse en una gestión más tradicional puede derivar en más errores durante el desarrollo. Por este motivo las métricas asociadas a la calidad son de gran relevancia en la gestión de proyectos con metodologías ágiles. Merecen especial atención por tanto las métricas categorizadas por Choraś et al. (2020) categorizadas como “Bug fixing” y “Testing”.

Entre las métricas encontradas respecto a la corrección de errores también se encuentran:

- **Tiempo de testing.** Se trata de la cantidad de tiempo que asigna para realizar las actividades de prueba y depuración para cada historia de usuario o característica. Se debe asignar el tiempo de prueba en función del esfuerzo de la prueba, la duración del sprint, la capacidad del equipo, las habilidades del equipo y la prioridad de la prueba.
- **Número de errores activos.** Se trata del número de errores del Sprint en curso que no se encuentran marcados como cerrados o finalizados, es decir, que aun no están corregidos. Un número muy alto de errores activos puede significar una pérdida de calidad en el producto, y un aumento de la deuda técnica progresiva.
- **Tareas obsoletas.** Esta métrica es de gran relevancia para la gestión ágil del proyecto, así como para el proyecto en sí. Hace referencia al número de tareas que no han tenido cambios en un periodo de tiempo. Puede referirse por ejemplo a tareas creadas hace más de tres meses, o tareas que cuya fecha de actualización es superior a tres semanas. Estas tareas pueden ir asociadas a productos que ya no son necesarios para el cliente, o a miembros del equipo que no están implicados en la gestión ágil del proyecto y no mantienen al día sus tareas.

#### 4.4. Métricas DORA

El equipo de Investigación y Evaluación de DevOps (DORA) se fundó en 2014 como un grupo de investigación independiente centrado en investigar las prácticas y capacidades que impulsan un alto rendimiento en la entrega de software y los resultados financieros.

El equipo de DORA es conocido por el informe *Annual State of Agile Report* que se publica desde el 2014. En 2019, Google adquirió DORA y es en 2018, cuando tres miembros del equipo de DORA, Nicole Forsgren, Jez Humble y Gene Kim, publican un libro llamado *Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performance Technology Organizations*, que relata en detalle la estrategia del grupo.

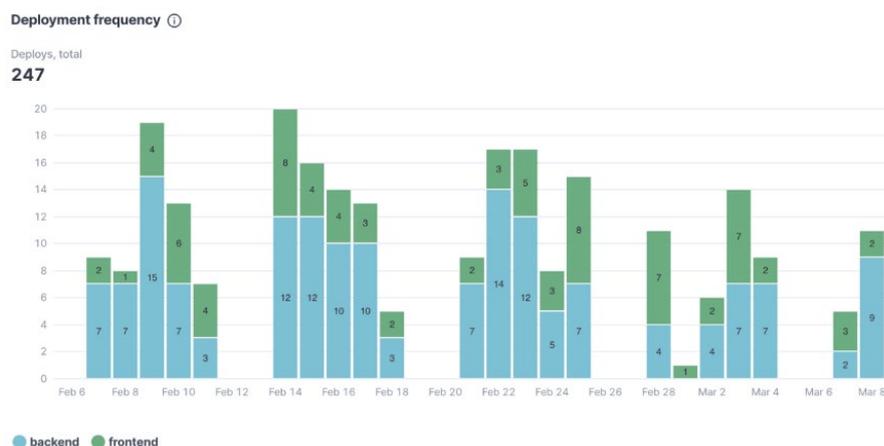
Desde que se publicó el libro *Accelerate* en 2018, es prácticamente imposible encontrar algún artículo sobre la métricas del rendimiento del desarrollo de software sin ninguna referencia a alguna de las cuatro métricas DORA que fueron popularizadas por el libro.

- **Frecuencia de despliegue.** La frecuencia de despliegue es una de las métricas más utilizadas en la que se pueden emplear para evaluar la productividad de un equipo y la eficacia general de un proyecto. Mide la frecuencia con la que un equipo despliega cambios en producción y por tanto la frecuencia con la que se entrega valor al cliente. Se puede calcular dividiendo el número total de despliegues realizados en un período de tiempo determinado (como por ejemplo, un mes, o un Sprint) por el número total de días en ese período.

A menudo se utiliza para rastrear la tasa de cambio en un proceso de desarrollo de software y puede usarse para identificar áreas potenciales de mejora. Una alta frecuencia de despliegue puede indicar un alto grado de colaboración, ciclos de retroalimentación rápidos y una cultura de experimentación e innovación. También va a indicar si las pruebas automatizadas son fiables, dado que permitirán rápidos despliegues en producción.

**Figura 7**

*Gráfico de frecuencia de despliegue*



*Nota.* Gráfico de la métrica frecuencia de despliegues de la herramienta Swarmia. [Practical guide to DORA metrics | Swarmia](#)

- **Lead Time.** Aunque descrita con anterioridad, se añade esta métrica en este punto dado que forma parte de las llamadas métricas DORA.
- **Mean time to recovery (MTTR).** Esta métrica muestra la eficiencia con la que los equipos de solucionan los problemas. Es el tiempo medio de recuperación en caso de fallo. Con ella se capta la gravedad del impacto, dado que un incidente puede provocar una interrupción significativa de las operaciones comerciales normales. Como se indicaba en el análisis de las métricas de calidad, estas son de gran importancia y deben siempre ser métricas en las que los equipos pongan el foco e intenten en este caso, reducir el tiempo al mínimo posible.

Se podría calcular como el tiempo medio de recuperación sumando el tiempo de inactividad total y dividiéndolo por el número total de incidentes que hayan ocurrido en un período particular. El tiempo de respuesta ante un incidente debe ser lo más corto posible, pero como máximo, 24 horas es una buena regla general.

Un tiempo medio bajo puede indicar un gran grado de madurez del equipo al realizar cambios más consistentes y que se están realizando buenas prácticas en cuanto a detección, prueba y monitoreo de errores.

- **Change Failure Rate (CFR).** Traducida de manera literal como tasa de fracaso del cambio o tasa de fallos, se trata de una métrica que provee información a los equipos sobre la presencia y gestión de errores. Analizar el CFR ayuda a localizar y solucionar defectos, y es una métrica fundamental para garantizar la integridad de los cambios en el código. El cálculo de esta métrica se realiza mediante la proporción de despliegues con defectos en producción sobre el total de despliegues realizados. De este modo, si un equipo despliega diez cambios y dos de ellos requirió correcciones, su Change Failure Rate será del 20%. Una alta tasa de fallos supone una pérdida de tiempo y recursos, impactando negativamente en los resultados entregados al cliente.

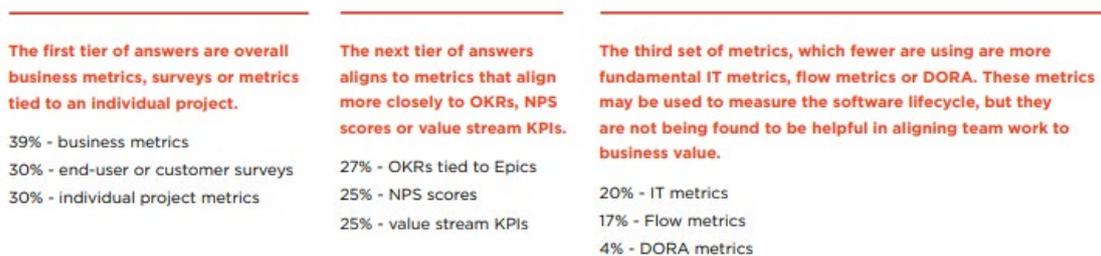
El porcentaje de despliegues defectuosos en producción varía en función de las capacidades y la experiencia del equipo. Según indica el informe *Accelerate State of DevOps* de 2021 (*2021 State of DevOps Report | Google Cloud, 2021*), el Change Failure Rate suele ser inferior al 15% en alto rendimiento. Sin embargo, en equipos con menor desempeño, esta tasa puede llegar al 30%.

## 4.5. Conclusiones

El aspecto que resalta el reporte *Annual State of Agile Report* (Digital.ai, 2022) con respecto a las métricas en metodologías ágiles es que el valor del negocio ocupa la primera posición con un 54 %, entre las métricas para medir el éxito de los proyectos ágiles, seguida de satisfacción del cliente (43 %) y la entrega a tiempo (35 %).

### Figura 8

*Extracto resultados Annual State of Agile Report*



Según la encuesta realizada el 39% métricas están orientadas a negocio, el 30% se trata de encuestas a clientes finales, y un 30% a proyectos individuales.

Destaca también que un 19% de las empresas encuestadas todavía no están seguras de como se está midiendo el valor de negocio. Esto es relevante dado que significa que aun hay una gran proporción de empresas que aun no han elegido, evaluado o dado a conocer las métricas que pueden ofrecer las metodologías ágiles.

En general las métricas evaluadas quedan divididas en dos grandes bloques que a las que van enfocadas, la velocidad y la calidad con la que se realiza el desarrollo

## 5. Herramientas de soporte a la gestión ágil y métricas que proporcionan.

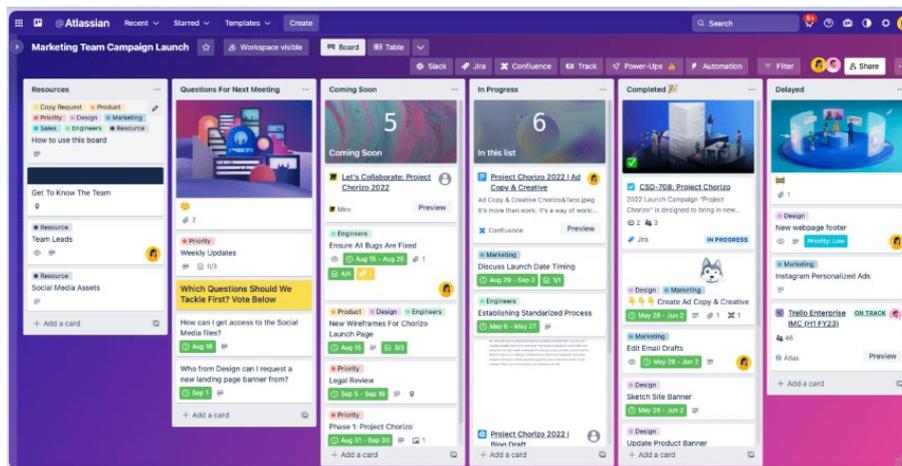
En este punto del estudio se evaluarán las más conocidas y utilizadas herramientas de soporte a la gestión ágil y las métricas que estas proporcionan. Se investigará cómo estas herramientas, diseñadas específicamente para el entorno ágil, desempeñan un papel crucial en la planificación, ejecución y seguimiento de proyectos ágiles. Además, se examinarán las métricas que estas herramientas ofrecen, las cuales son esenciales para medir el progreso del proyecto, evaluar la calidad del trabajo y tomar decisiones basadas en datos.

- **Trello.**

Trello es un de los software de gestión de trabajo con interfaz web más utilizados en la actualidad por su amplia trayectoria desde 2011 y por su facilidad de uso. Es por ello que en la gran mayoría de las fuentes consultadas consta en primer lugar entre las herramientas que facilitan la aplicación de las metodologías ágiles a la hora de gestionar proyectos.

**Figura 9**

*Tablero de Trello*



*Nota.* Jira Software. (2022). Jira Software frente a Trello.

<https://www.atlassian.com/es/software/jira/comparison/jira-vs-trello>

Trello taene gestión de tareas, gestión de recursos y gestión de documentos, pero no proporciona seguimiento del tiempo, informes y gestión de problemas.

Tiene una herramienta llamada Vista de Panel, en el que se pueden obtener métricas y estadísticas del proyecto.

Para obtener estas métricas Trello permite seleccionar en varios pasos unas limitadas opciones para mostrar un dashboard con los datos relativos al proyecto. El dashboard puede ser compartido, pero está bastante limitado en cuanto a funcionalidad.

**Figura 10**

*Creación de cuadro de mando en Trello*

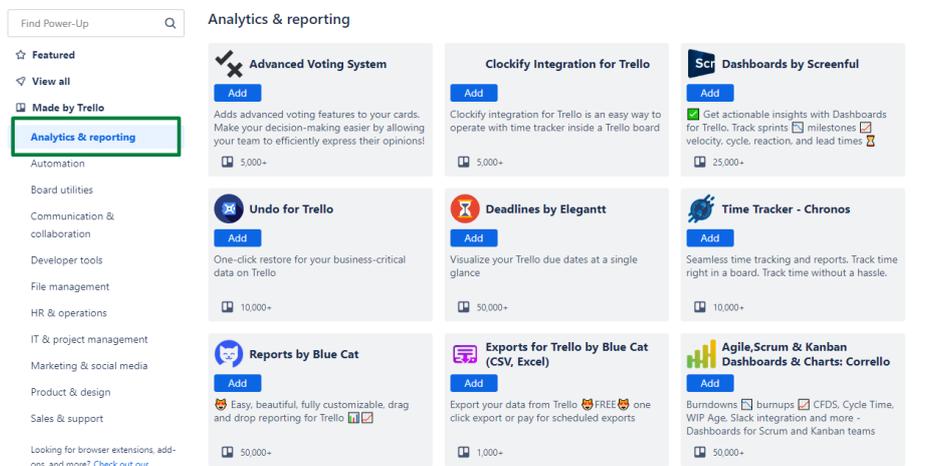


*Nota.* Joiner, B. (2021). Un resumen sobre las funciones controladas por datos de Trello. [Un resumen sobre las funciones controladas por datos de Trello](#)

Una de las ventajas de Trello es que también pueden añadirse distintos *Dashboards* mediante las herramientas que existen disponibles en *Power-ups*.

**Figura 11**

*Biblioteca de aplicaciones disponibles para Trello en el área Analytics & Reporting*



*Nota.* Trello. (S.f.). Power – Ups disponibles. [Power-Ups | Trello](#)

En la actualidad existen un total de setenta y tres herramientas enfocadas en el reporting, lo que permite realizar un control exhaustivo, y muy personalizado de los datos que arroja Trello durante la realización del proyecto.

**Figura 12**

*Gráfica de tiempo de ciclo de la aplicación Corrello*



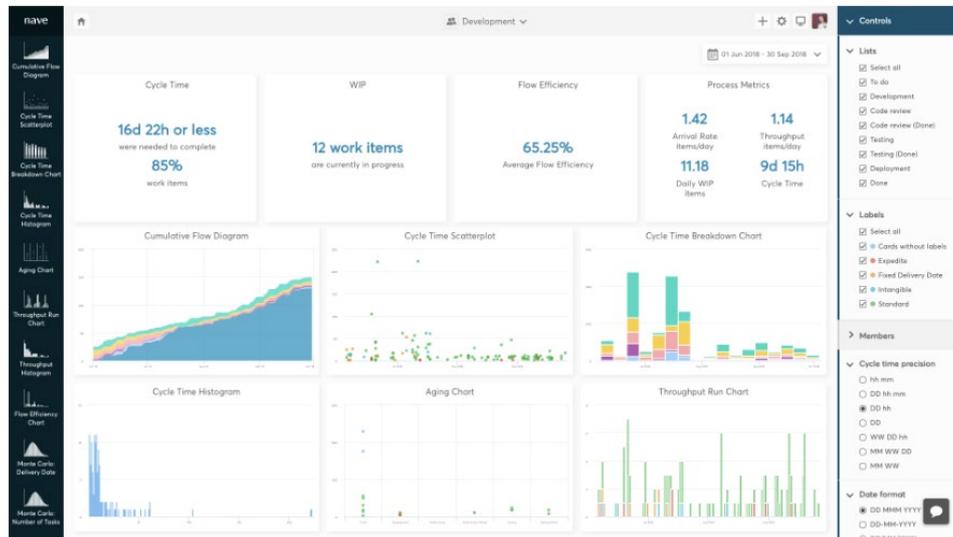
*Nota.* Trello. (S.f.). Agile, Scrum & Kanban Dashboards & Charts: Corrello.

<https://trello.com/power-ups/568c2b6c7d2758a55ac48c98/agile-scrum-kanban-dashboards-charts-corrello>

Pueden encontrarse tanto herramientas sencillas como las que permiten exportar los datos a Excel o CSV, tarjetas para visualizar cuánto tiempo lleva una tarjeta en un estado, conector para Power BI, o herramientas con *dashboards* que permiten la visualización de métricas con personalización de las mismas.

**Figura 13**

*Cuadro de mando con las métricas Tiempo de ciclo, WIP, % Eficiencia, número de items registrados al día, rendimiento de items al día, Diagrama de flujo acumulado, desglose e histograma del tiempo de ciclo y antigüedad*



Nota. Trello. (S.f.). Kanban Analytics by Nave. <https://trello.com/power-ups/5c4af277c0430d10357c87dc/kanban-analytics-by-nave>

- **Asana**

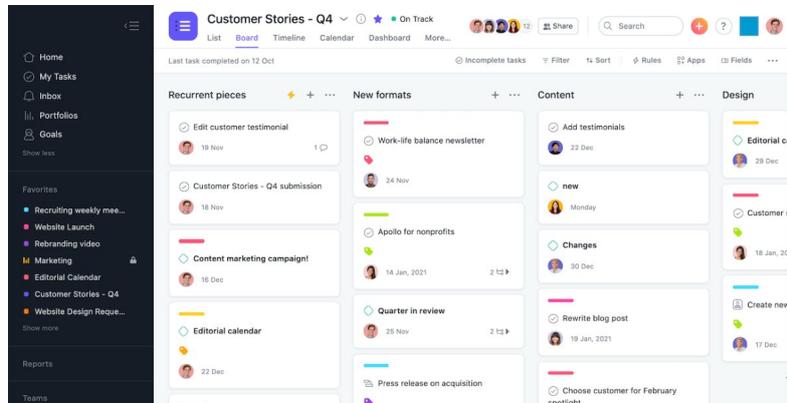
Asana es un software de gestión ágil de proyectos que se caracteriza por su versatilidad y flexibilidad. Tiene opción de compartir, planificar y rastrear el avance de las actividades asignadas a los miembros del grupo de trabajo para lo que incluye tableros, líneas de tiempo, campos personalizados, dependencias, etc.

Como indica Roselló, V. (2022) esta herramienta se creó por el co-fundador de Facebook y ha ganado popularidad en los últimos años, contando ya con 400.000 usuarios.

Además, Asana se integra con más de 100 herramientas, como Jira, GitHub o Bitbucket y cuenta con una versión gratuita para grupos de hasta 15 usuarios.

Figura 14

Tablero de la herramienta Asana



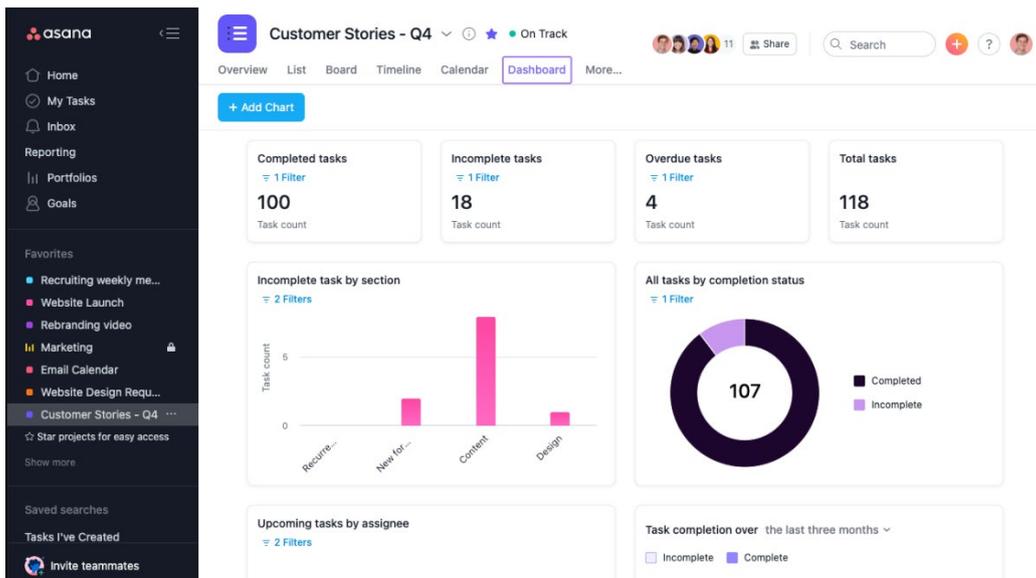
Nota. Asana. (S.f.). Explora Asana. Asana Guide.

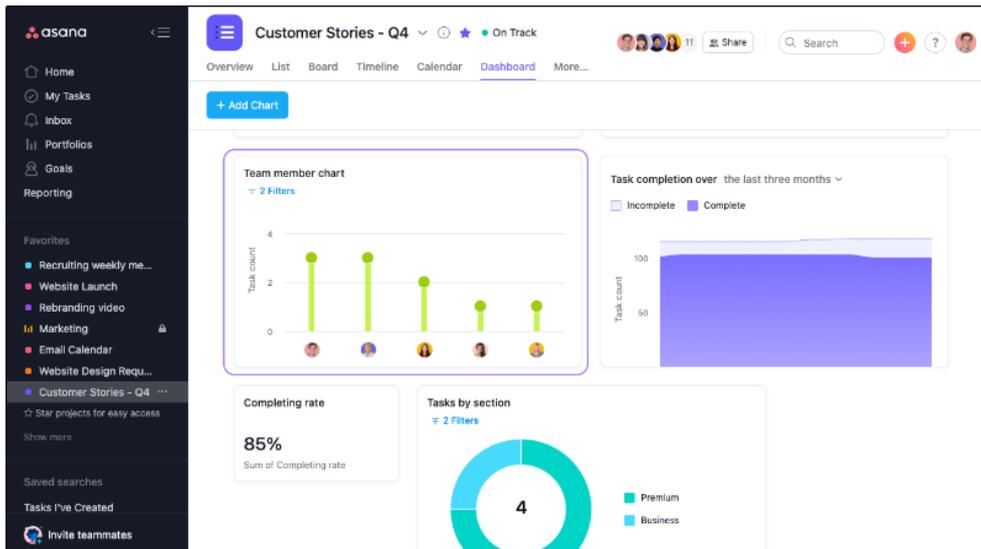
<https://asana.com/es/guide/help/fundamentals/navigating-asana>

Asana cuenta con dos apartados que permiten transformar los datos en información y métricas que permitirán a los usuarios tomar decisiones sobre sus proyectos y mantenerse informados sobre los avances de los mismos. En primer lugar se encuentra el apartado *Dashboard* dentro del proyecto y *Reporting* en el menú general.

Figura 15

Opciones *Dashboard* y *Reporting* de la herramienta Asana





Nota. Asana. (S.f.). Paneles de proyectos. Asana Guide.

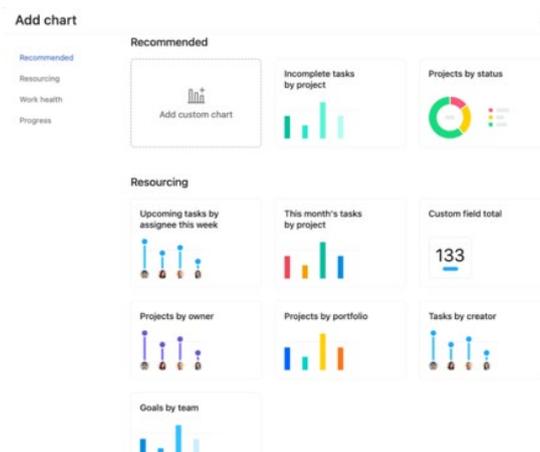
<https://asana.com/es/guide/help/projects/dashboards>

En el primero se pueden visualizar métricas generales como: tareas completadas e incompletas, tareas atrasadas, conteo de tareas por estados, tareas por responsable, un gráfico burn-up. También es posible cambiar en las gráficas la variable analizada en el eje X o la variable analizada en el caso de los campos numéricos.

En este reporte se pueden añadir más gráficas personalizadas y gráficas de valores numéricos acumulados. En el apartado *Reporting* se pueden generar gráficas con distintas métricas que son asociadas a proyectos.

### Figura 16

Opciones de reporting de métricas en Asana



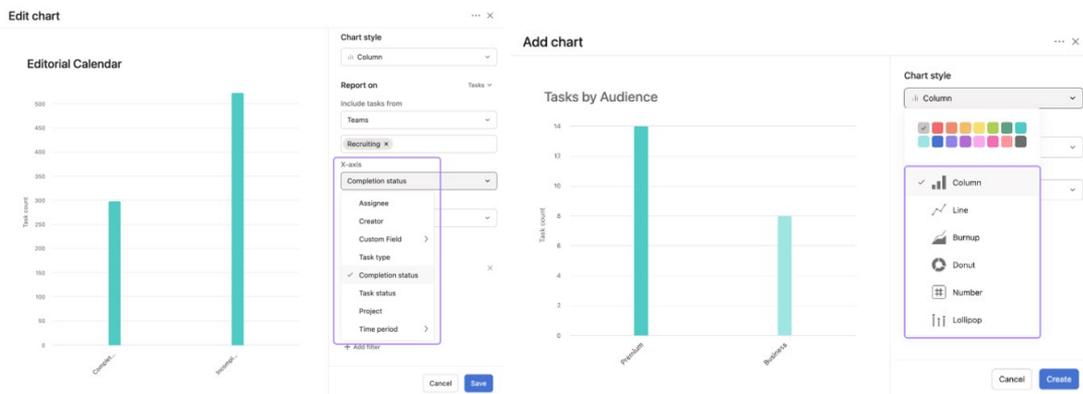
Nota. Asana. (S.f.). Informes. Asana guide.

<https://asana.com/es/guide/help/premium/reporting>

En el caso de *Reporting* Asana permite crear gráficos a partir de tareas, proyectos u objetivos, incluir tareas desde una organización, equipos, portafolios, proyectos de una persona específica o una selección de proyectos en particular. Se permite personalizar el eje X por responsable, tipo de tarea, creador, estado o tipo de la tarea, proyecto o periodo, y por campos personalizados. El eje Y se puede personalizar por cantidad de tareas, tiempo para finalización o suma o media de un campo numérico. También se permite crear filtros sobre varios campos. En algún caso alguna funcionalidad está limitada a las opciones de pago.

**Figura 17**

*Opciones de personalización de gráficos en Asana*



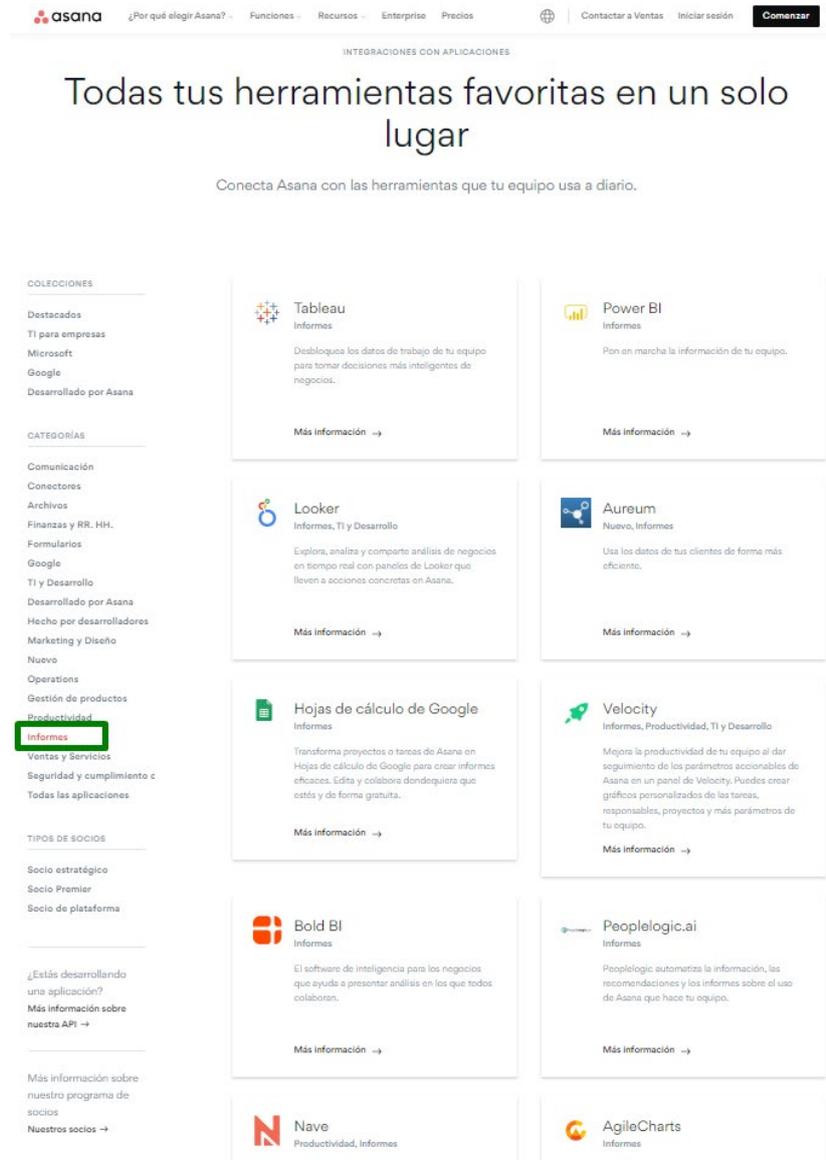
*Nota.* Asana. (S.f.). Informes. Asana guide.

<https://asana.com/es/guide/help/premium/reporting>

En Asana también existen cuarenta conectores con aplicaciones que permiten realizar *Dashboards* con métricas más personalizadas a demanda del usuario:

**Figura 18**

*Extracto de conectores de Asana con distintas apps*



*Nota.* Asana. (S.f.). Integraciones con aplicaciones. Asana guide.

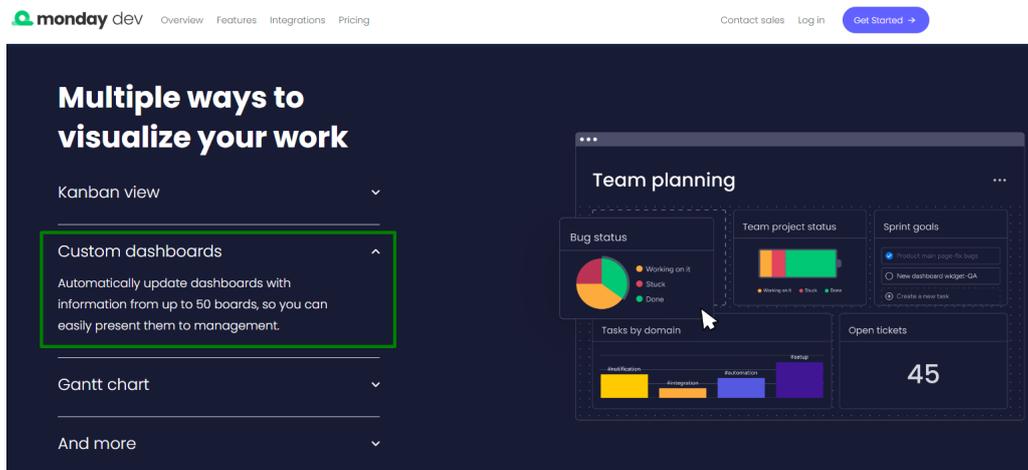
<https://asana.com/es/apps?category=informes>

- **Monday.com**

Monday es una plataforma de computación lanzada en 2014 que cuenta con más de 180.000 usuarios, que permite realizar flujos de trabajo y software de administración de proyectos, sin necesidad de programar. Ofrece una gran variedad de opciones en función del entorno para el que quiera crear el desarrollo, como por ejemplo: Marketing, Recursos Humanos, Ventas y CRM, IT, etc.

Figura 19

Opciones de personalización de dashboards en Monday Dev

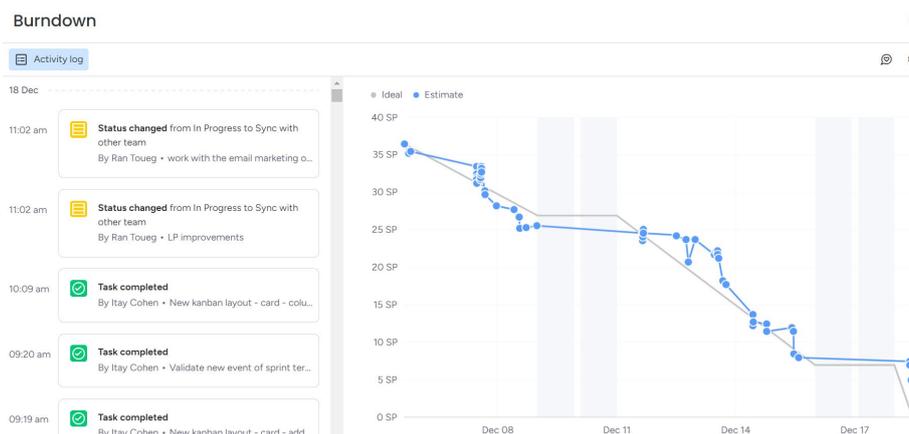


Nota. Monday. (S.f.). Monday features. Monday. <https://monday.com/dev/features>

Entre las métricas que se pueden encontrar en Monday se encuentran el gráfico de burndown que fue implementado en la herramienta en 2022, conteo de tareas por estado, y dado que es una herramienta centrada también en el Marketing, si cuenta con métricas sobre satisfacción del cliente.

Figura 20

Gráfico Burndown de la herramienta Monday

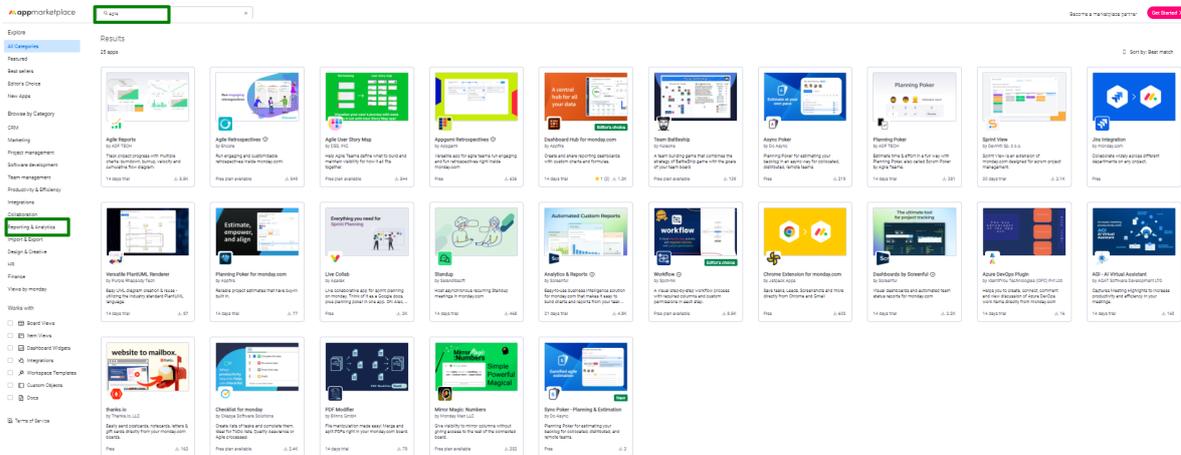


Nota. Monday. (S.f.). Burndown charts. Mondya help center, Community. [Burndown charts - Feature requests - monday Community Forum](#)

Cuenta con un marketplace en la que se pueden encontrar un total de veinticinco conectores con herramientas asociadas la gestión mediante metodologías ágiles.

Figura 21

## Herramientas con las que se puede integrar Monday para gestionar mediante metodologías ágiles



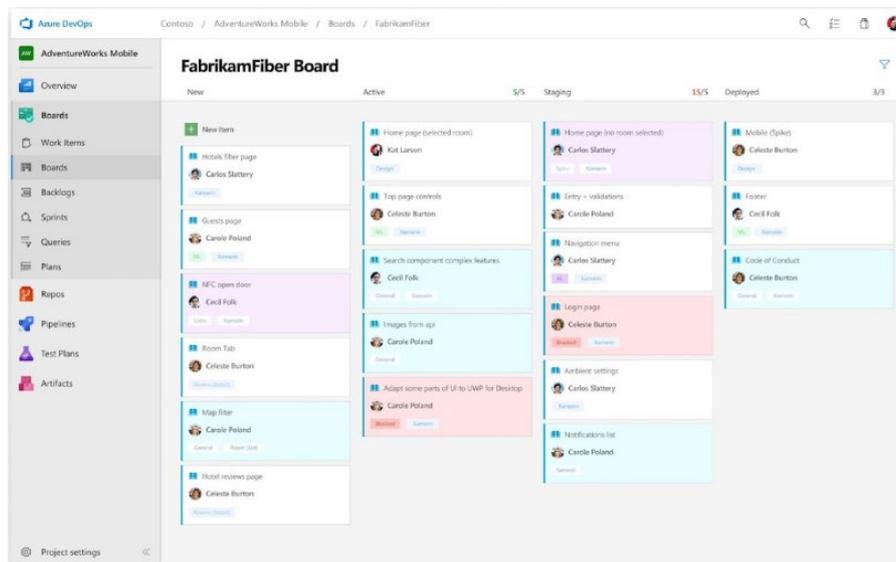
Nota. Monday. (S.f.). App Marketplace. Monday. <https://monday.com/marketplace>

- **Azure Devops**

Se trata de una de las herramientas más potentes del mercado, dado que forma parte de la Suite de Microsoft. La herramienta, que anteriormente englobaba Team Foundation Server (TFS), y Visual Studio Team System (VSTS) cuenta ya con una trayectoria de dieciocho años, lo que la sitúa en las primeras herramientas que se pueden utilizar para la gestión de proyectos, tanto para metodologías en cascada como ágiles. Cubre todo el ciclo de vida del software con herramientas para gestión de proyectos, control de versiones (permite integración con Git), gestión de requisitos, informes, pruebas, etc.

Figura 22

Tablero de Azure DevOps



Nota. AzureFunBytes - A Brief Intro To Azure Boards. Gordon, J. (2021). DevBlogs.

[AzureFunBytes - A Brief Intro To Azure Boards - Azure DevOps Blog \(microsoft.com\)](https://www.azurefunbytes.com/2021/02/02/a-brief-intro-to-azure-boards/)

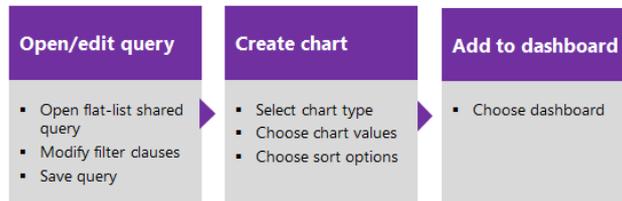
La gestión de informes mediante Azure Boards que forma parte de los componentes de la herramienta desde 2005, cuenta con un almacén de datos y un cubo *Analysis services* (SSAS). Mediante estos orígenes se pueden generar reportes mediante *Reporting Services* (SSRS), Excel, Power BI, etc. Se incluyen por defecto plantillas que incluyen métricas como *Release Burndown*, *Sprint Burndown* and Velocidad y datos sobre los errores tratados que pueden ser personalizados.

En 2012 se introdujo el componente de gráficas “ligeras”, que ofrece diagramas CFD, de velocidad y de avance en tiempo real directamente dentro de Team Web Access.

Esta personalización permite por ejemplo visualización de velocidad según esfuerzo, puntos de historia, tamaño u otro campo numérico personalizado.

**Figura 23**

Proceso de creación de gráficas “ligeras” personalizadas con Azure Boards en Azure DevOps



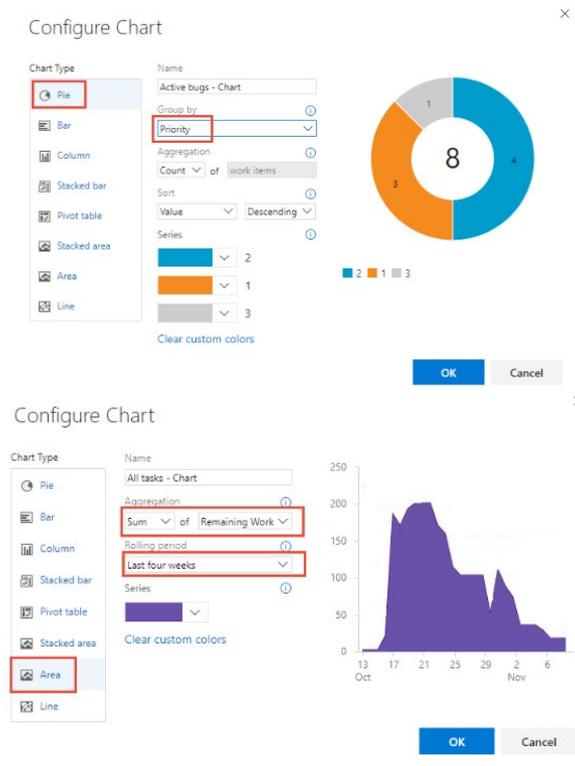
*Nota.* Azure DevOps. (S.f.). Administrar objetos de plantilla de proceso de Scrum. (2023).

Documentación Azure DevOps. [Administrar objetos de plantilla de proceso de Scrum - Azure Boards | Microsoft Learn](#)

Se incluyen estos gráficos “ligeros”, los gráficos de estado (circular, de barras, de columnas, de barras apiladas y tablas dinámicas) y de tendencia (de áreas apiladas, de líneas y de áreas).

**Figura 24**

Configuración de un gráfico basado en consultas y un gráfico de evolución en Azure DevOps

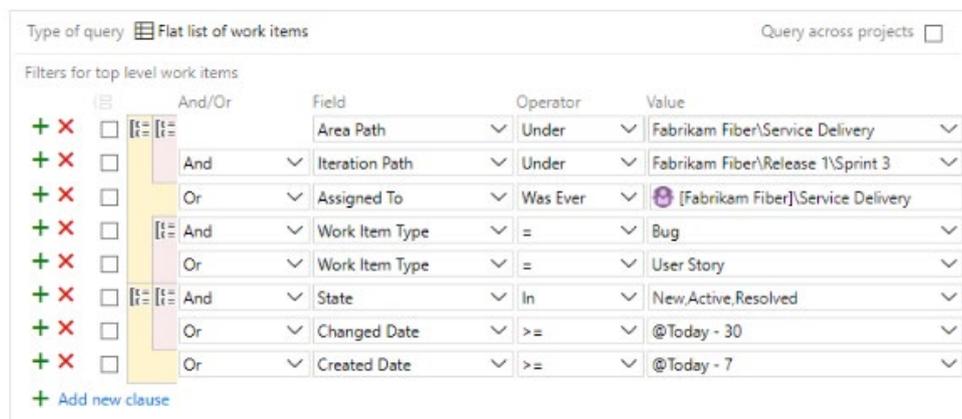


*Nota.* Azure DevOps. (2023). Seguimiento del progreso con gráficos basados en consultas de estado y tendencias. Documentación AzureDevOps. [Elemento de trabajo de estado y tendencia, gráficos basados en consultas - Azure DevOps | Microsoft Learn](#)

Para la realización de gráficos se pueden utilizar como origen las consultas planas, que se definen mediante la opción Query. Estas consultas permiten una personalización hasta el nivel más bajo según todos los campos que posee el board, lo que enriquece las métricas que se quieran obtener. Todas las gráficas pueden añadirse a un panel o dashboard para su visualización en conjunto.

**Figura 25**

*Personalización de consultas*

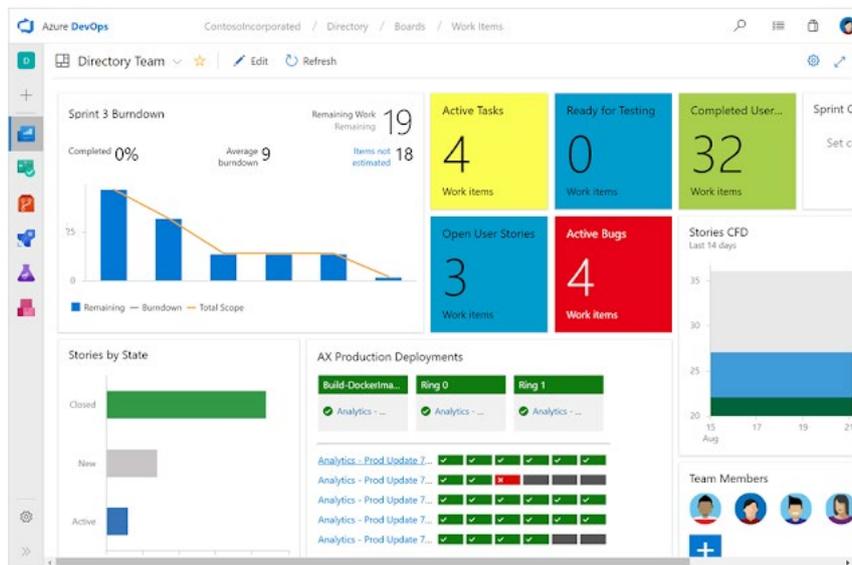


*Nota.* Azure DevOps. (2023). Definición de una consulta de elementos de trabajo en Azure Boards. Documentación Microsoft. [Definición de una consulta de elementos de trabajo con el editor de consultas en Azure Boards - Azure Boards | Microsoft Learn](#)

Estas consultas permiten una personalización máxima de métricas que se pueden obtener, dado que se puede adaptar a cada una de las necesidades de medición de los cada equipo y cada proyecto de manera individual.

Figura 26

Dashboard con métricas de un sprint en Azure DevOps



Nota. Azure DevOps. (2019). Deep dive into Azure Boards. Watson, K. Microsoft Online Guide. [Deep dive into Azure Boards | Microsoft Online Guide](#)

En Azure Boards se incluyen por defecto una serie de vistas de *Analytics*, con su descripción correspondiente, que ayudará a sacar métricas comunes según un periodo de tiempo acotado.

Figura 27

Vistas incluidas en Analytics views en Azure Board

The screenshot shows the 'Analytics views (Boards only)' page in Azure Boards. The page lists various views for different categories of work items, including Bugs, Issues, Tasks, and Work Items. Each view is described with its scope and reporting intervals.

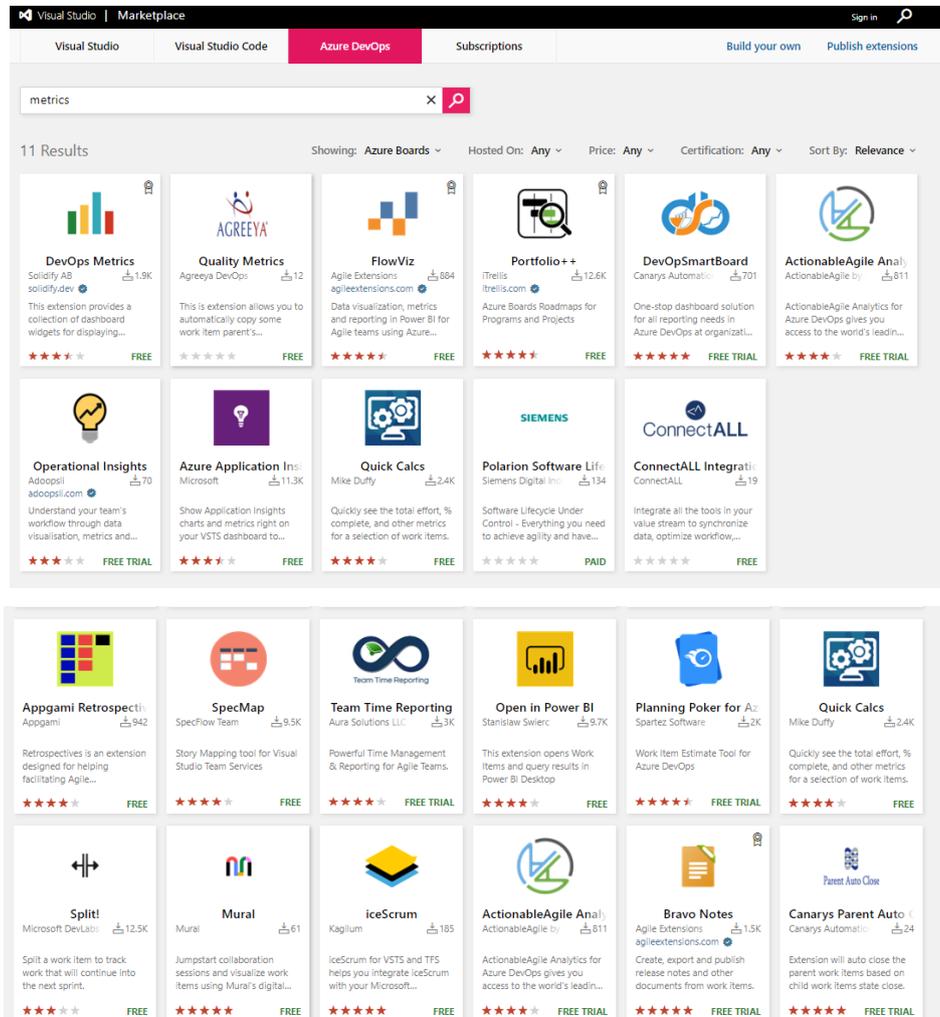
Name	Description
Analytics views let you create filtered views of simple Boards data for Power BI reporting. For help on reporting in Power BI click here.	
My Views	
Shared Views	
Bugs - All history by month	All Bugs for the entire team project. All history with monthly intervals.
Bugs - Last 26 weeks	All Bugs for the entire team project. Last 26 weeks of history with weekly intervals.
Bugs - Last 30 days	All Bugs for the entire team project. Last 30 days of history with daily intervals.
Bugs - Today	All Bugs for the entire team project. No history.
Issues - All history by month	All Issues for the entire team project. All history with monthly intervals.
Issues - Last 26 weeks	All Issues for the entire team project. Last 26 weeks of history with weekly intervals.
Issues - Last 30 days	All Issues for the entire team project. Last 30 days of history with daily intervals.
Issues - Today	All Issues for the entire team project. No history.
Tasks - All history by month	All Tasks for the entire team project. All history with monthly intervals.
Tasks - Last 26 weeks	All Tasks for the entire team project. Last 26 weeks of history with weekly intervals.
Tasks - Last 30 days	All Tasks for the entire team project. Last 30 days of history with daily intervals.
Tasks - Today	All Tasks for the entire team project. No history.
Work Items - All history by month	All work items for the entire team project. All history with monthly intervals.
Work Items - Last 26 weeks	All work items for the entire team project. Last 26 weeks of history with weekly intervals.
Work Items - Last 30 days	All work items for the entire team project. Last 30 days of history with daily intervals.
Work Items - Today	All work items for the entire team project. No history.

Nota. Azure DevOps. (2023). Acerca de las vistas de Analytics. Documentación AzureDevOps. [Acerca de las vistas de Analytics - Azure DevOps | Microsoft Learn](#)

Al igual que el resto de herramientas cuenta con integraciones con distintas aplicaciones que permiten generar distintas visualizaciones de métricas asociadas a metodologías ágiles.

**Figura 28**

*MarketPlace de Azure DevOps sobre métricas y metodologías ágiles*



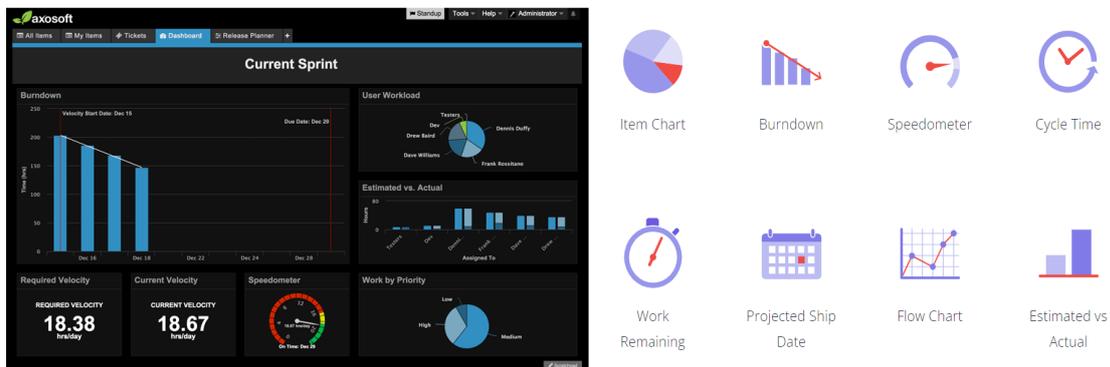
*Nota.* Azure DevOps. (S.f.). Extensions for Azure DevOps. Microsoft Marketplace. [Extensions for Visual Studio family of products | Visual Studio Marketplace](#)

- **Axosoft**

Anteriormente conocida como OnTime, se trata de un software Scrum repleto de funciones, desarrollado especialmente para equipos y proyectos de software ágiles. Permite planificación de proyectos, colaboración, y funcionalidades ágiles de gestión de bugs y seguimiento de incidentes para los equipos de desarrollo.

Figura 29

Dashboard y opciones de creación de gráficas de métricas en Axosoft



Nota. Axosoft. (S.f.). Easy agile reporting. Axosoft. [Agile Reporting - Agile Dashboard | Axosoft](#)

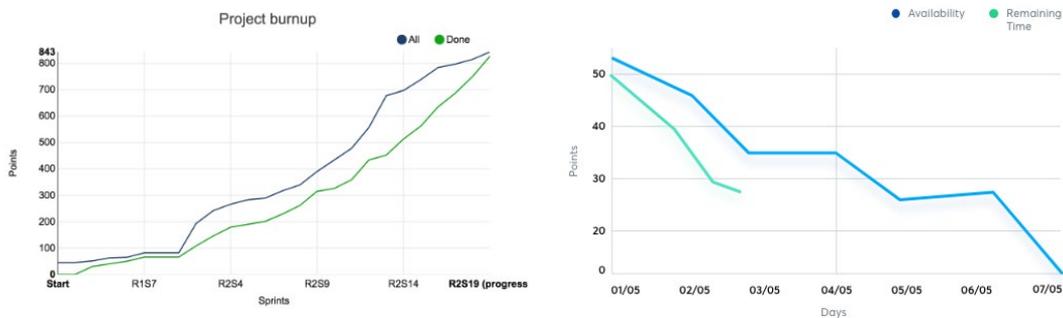
Cuenta con integraciones con las siguientes herramientas de reporting. Axosoft Report Manager, Axosoft Stopwatch, Clockify, QlikView, Timeify y Trello.

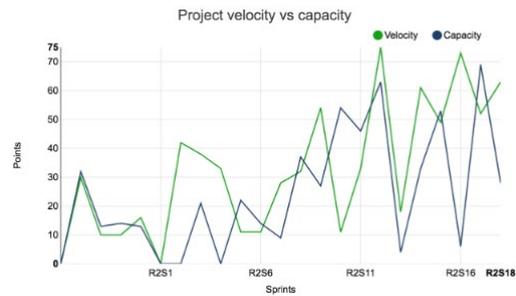
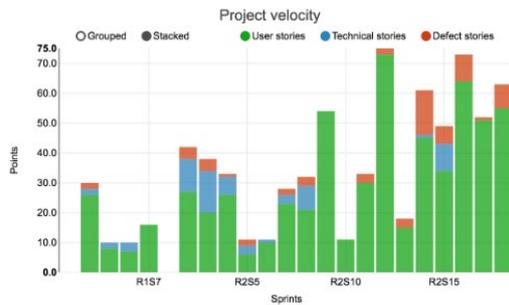
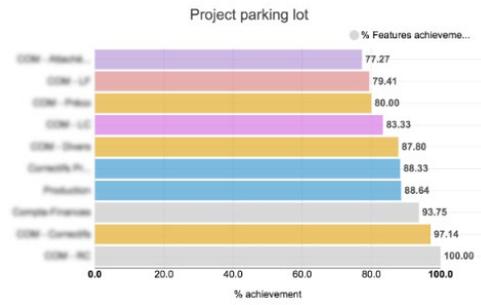
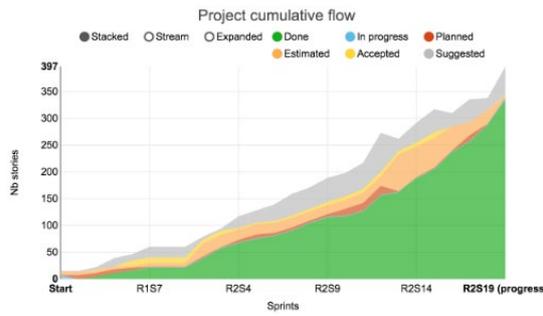
- **iceScrum**

Ofrece gestión de proyectos mediante SAFE y LeSS. Cuenta con los siguientes gráficos predefinidos para evaluar métricas ágiles: burndown, burnup, parking lot, diagrama de flujo acumulado, velocidad y velocidad vs capacidad. Aunque se trata de buenas gráficas y métricas relevantes, no permite personalización ni del tipo de gráfico de los elementos mostrados en los ejes.

Figura 30

Gráficos de burnup, burndown, diagrama de flujo acumulado, parking lot, velocidad y velocidad vs capacidad de la herramienta iceScrum





Nota. iceScrum. (S.f.). Indicators and reporting. iceScrum documentation. [Indicators and reporting – iceScrum](#)

Dentro de las aplicaciones con las que tiene integración se encuentra por ejemplo Agile KPIs, que incluye las métricas de *Cycle Time*, *Lead Time* y *Throughput* (rendimiento).

## 5.1. Análisis comparativo

En primer lugar surge la comparativa más relevante para este estudio que es Jira vs Trello. Ambas de Atlassian.

Se puede utilizar una combinación de ambas para crear desde cero o vincular las incidencias que se tengan en Jira con las tarjetas de Trello por ejemplo. Aunque la gestión de tarjetas es posible realizarla con una interfaz muy similar e igualmente sencilla mediante Jira Cloud.

Asana al igual que Trello permite realizar gráficas sobre valores numéricos acumulados. Asana sin embargo permite mayor personalización tanto de tipo de gráfica como de valores que pueden seleccionarse para los distintos ejes, aunque en algún caso los valores o filtros están limitados a las opciones de pago de la aplicación.

Monday pese a ser una herramienta muy completa no centra sus desarrollo en la visualización de métricas. Sin embargo al igual que el resto, posee integración con distintas aplicaciones para su extracción. Uno de sus puntos fuertes es que se trata de una herramienta que si puede ofrecer un nivel alto de métricas enfocadas a la satisfacción del cliente.

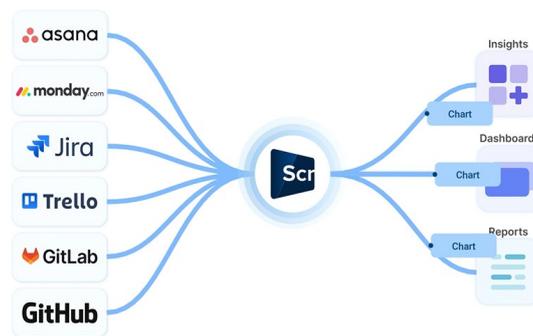
En el caso de Azure Boards dentro de Azure DevOps se trata de una de las mejores y más completas herramientas dado que forma parte del paquete de Microsoft. Su desarrollo es de los más completos, cuenta con una amplia documentación así como con una comunidad de gran tamaño dado el número de usuarios en todo el mundo que utilizan esta aplicación. Por otra parte, al ser un producto de estas características, cuenta con pocos sistemas que permitan su integración, al contrario de sus competidoras. Sin embargo cuenta con el conector directo de Power BI y conexión mediante OData, que permite diseñar reportes e informes completos y muy personalizables, dado que igualmente Power BI es una de las herramientas más completas y utilizadas del mercado.

Axosoft, aunque no es la herramienta más potente de entre las comparadas, si que destaca por la sencillez con la que están presentadas las opciones de métricas y gráficas. Éstas no cuentan con grandes opciones de personalización, pero para equipos que se inician en metodologías ágiles y que tengan necesidades muy sencillas en cuanto a métricas puede ser una gran opción.

En muchos casos, dada la existencia de aplicaciones que generan reportes y *dashboards* donde se pueden calcular métricas personalizadas, la gran ventaja que aportan estas aplicaciones radica que poder utilizar la información de varios orígenes de datos para poder gestionar equipos o proyectos que no trabajen con una única herramienta de gestión. Este es el caso por ejemplo de la aplicación Screenful (Scr).

**Figura 31**

*Integración de distintas herramientas con Scr para reporting*



*Nota.* Screenful (S.f.). Universal Productivity Data Platform. Screenful. <https://screenful.com/>

[¿Qué diferencias hay entre Jira Work Management y Trello? \(deiser.com\)](#)

Llama especialmente la atención que en muchas ocasiones las aplicaciones analizadas pueden enlazarse entre ellas. Esto aunque en un principio puede no tener mucho sentido, dado que es de esperar que en una compañía se utilice una única herramienta de gestión ágil, tiene sentido porque precisamente estas metodologías se basan en la flexibilidad. Por tanto puede ser especialmente útil que un gestor de proyectos tenga varios proyectos y equipos y que cada uno utilice una herramienta de gestión que más se adapte a sus necesidades. Por ese motivo tendría gran interés poder tener una imagen global de todas las métricas con un “lenguaje universal” en el que poder visualizar el estado de distintos proyectos.

## 6. Jira Cloud

Jira Cloud es una herramienta de gestión de proyectos y seguimiento de problemas basada en la nube desarrollada por Atlassian. Está diseñada para ayudar a los equipos a planificar, rastrear y administrar sus proyectos de manera colaborativa. Jira Cloud se utiliza ampliamente en el desarrollo de software y en otros campos que requieren la gestión eficiente de tareas y proyectos.

Entre las características principales de Jira Cloud cabe destacar:

- **Gestión de proyectos:** Jira Cloud permite a los equipos crear proyectos y organizar tareas en tableros, listas o vistas de calendario. Los proyectos pueden configurarse con estructuras personalizadas según las necesidades del equipo.
- **Seguimiento de problemas:** Los problemas, también conocidos como tickets, representan las tareas, errores, mejoras o solicitudes de función que deben ser abordados por el equipo. Jira Cloud proporciona un sistema de seguimiento de problemas flexible y personalizable, que permite asignar problemas a miembros del equipo, establecer prioridades, adjuntar archivos, agregar comentarios y realizar un seguimiento del progreso.
- **Tableros Kanban y Scrum:** Jira Cloud ofrece tableros Kanban y Scrum que permiten a los equipos visualizar y gestionar su trabajo de manera efectiva. Los tableros Kanban se centran en la gestión visual del flujo de trabajo, mientras que los tableros Scrum se basan en sprints y ofrecen características específicas para la planificación y el seguimiento ágil.
- **Integraciones y complementos:** Jira Cloud se integra con una amplia gama de herramientas y servicios, como Confluence (herramienta de colaboración), Bitbucket (plataforma de control de versiones), Slack (herramienta de comunicación) y muchas más. Además, hay una amplia variedad de complementos disponibles en el Marketplace de Atlassian para personalizar y ampliar las capacidades de Jira Cloud según las necesidades específicas del equipo.
- **Informes y métricas:** Jira Cloud ofrece una serie de informes y métricas para ayudar a los equipos a evaluar su rendimiento y tomar decisiones informadas. Estos informes incluyen el seguimiento de tiempo, la velocidad del equipo, el flujo de trabajo de los problemas y muchas otras métricas relevantes para la gestión de proyectos y el seguimiento del progreso.

- Acceso en la nube: Jira Cloud es una solución basada en la nube, lo que significa que no es necesario instalar ni mantener servidores locales. Esto permite a los equipos acceder a Jira desde cualquier lugar y dispositivo con conexión a Internet, lo que facilita la colaboración remota y el acceso rápido a la información del proyecto.

En resumen, Jira Cloud es una herramienta de gestión de proyectos y seguimiento de problemas basada en la nube que ofrece funcionalidades amplias y personalizables para equipos que desean gestionar eficientemente sus proyectos y tareas.

### 6.1.1. Diferencias con Jira

Jira es una suite de herramientas de gestión de proyectos y seguimiento de problemas desarrollada por Atlassian, que incluye varias ediciones, como Jira Software, Jira Service Management (anteriormente conocido como Jira Service Desk) y Jira Core. Entre Jira y Jira Cloud destacan las siguientes diferencias:

- Jira Cloud: Jira Cloud se refiere específicamente a la versión basada en la nube de Jira, donde los usuarios acceden a la herramienta a través de un navegador web y no se requiere instalación local o mantenimiento de servidores. Proporciona una experiencia de usuario simplificada y actualizaciones automáticas.
- Jira Server y Jira Data Center: Estas son las versiones autohospedadas de Jira, que requieren la instalación y configuración en servidores locales o en la infraestructura de la organización. A diferencia de Jira Cloud, estas versiones permiten un mayor control y personalización, pero también requieren la responsabilidad de la gestión y el mantenimiento del servidor.
- Disponibilidad de características: Jira Cloud y Jira Server/Data Center ofrecen características similares, pero puede haber algunas diferencias en cuanto a las funcionalidades disponibles. Algunas características pueden estar disponibles primero en Jira Cloud antes de ser implementadas en las versiones autohospedadas, y viceversa. Es importante consultar la documentación de Atlassian para conocer las diferencias específicas entre las ediciones.
- Integraciones y complementos: Tanto Jira Cloud como Jira Server/Data Center admiten una amplia gama de integraciones y complementos a través del Marketplace de Atlassian.

Sin embargo, algunos complementos pueden estar diseñados específicamente para una edición en particular y no ser compatibles con otras ediciones. Por lo tanto, es importante verificar la disponibilidad de los complementos requeridos para la edición de Jira que se esté utilizando.

- Escalabilidad y rendimiento: Jira Cloud se beneficia de la infraestructura escalable y robusta de la nube, lo que permite una mayor capacidad de adaptación y rendimiento para gestionar proyectos y equipos de diferentes tamaños. Jira Server/Data Center, por otro lado, ofrece la posibilidad de escalar horizontalmente mediante la adición de nodos para manejar una mayor carga de usuarios y proyectos.
- Actualizaciones y mantenimiento: En Jira Cloud, las actualizaciones y el mantenimiento de la plataforma son responsabilidad de Atlassian, lo que significa que los usuarios reciben automáticamente las últimas actualizaciones y correcciones de seguridad. En Jira Server/Data Center, los administradores son responsables de gestionar las actualizaciones y el mantenimiento, lo que implica una mayor dedicación y control en comparación con Jira Cloud.

En general, la elección entre Jira Cloud y Jira Server/Data Center depende de los requisitos y las preferencias específicas de cada equipo u organización. Jira Cloud es ideal para aquellos que buscan una solución lista para usar y sin preocupaciones de infraestructura, mientras que Jira Server/Data Center ofrece más control y personalización, pero requiere más responsabilidad en términos de instalación y mantenimiento.

Para el desarrollo del proyecto de simulación un proyecto ágil se utilizará la herramienta gratuita Jira Cloud.

### 6.1.2. Jira Cloud y Metodologías Ágiles

Jira Cloud tiene una estrecha relación con las metodologías Agile, ya que se ha desarrollado específicamente para apoyar la gestión ágil de proyectos y el seguimiento de tareas. A continuación, se describen algunas formas en las que Jira Cloud se relaciona con las metodologías Agile:

- Metodologías ágiles incorporadas: Jira Cloud proporciona plantillas y flujos de trabajo predefinidos que se alinean con las metodologías ágiles Scrum y Kanban.

Estos flujos de trabajo se pueden personalizar según las necesidades específicas del equipo, lo que permite una implementación ágil más rápida y eficiente.

- Seguimiento y visualización del trabajo: Jira Cloud ofrece tableros Kanban y tableros Scrum que permiten a los equipos visualizar y gestionar su trabajo de manera efectiva. Estas vistas proporcionan una representación visual del flujo de trabajo y permiten a los miembros del equipo arrastrar y soltar tareas entre las diferentes etapas del proceso. Esto facilita la colaboración, el seguimiento del progreso y la identificación rápida de cuellos de botella o tareas bloqueadas.
- Gestión de historias y tareas: En Jira Cloud, los equipos pueden crear historias de usuario, tareas y subtareas para representar las unidades de trabajo en su proyecto. Estos elementos se pueden priorizar, asignar a miembros del equipo y rastrear su progreso a medida que avanzan a través del flujo de trabajo. Esto facilita la descomposición de proyectos en elementos más manejables y proporciona una visión clara del trabajo pendiente y completado.
- Sprint planning y seguimiento de velocidad: Jira Cloud permite a los equipos planificar y realizar un seguimiento de los sprints en Scrum. Los equipos pueden crear sprints, asignar tareas a ellos y establecer objetivos y fechas de finalización. Además, Jira Cloud calcula automáticamente la velocidad del equipo, que es la cantidad de trabajo completado en cada sprint. Esta información ayuda a los equipos a estimar y planificar futuros sprints de manera más precisa.
- Informes y métricas ágiles: Jira Cloud proporciona una variedad de informes y métricas para evaluar el rendimiento del equipo y el progreso del proyecto. Estas se desarrollarán en el siguiente punto.

En resumen, Jira Cloud está diseñado para brindar soporte a las metodologías ágiles al ofrecer características y funcionalidades específicas para la gestión ágil de proyectos. Desde la planificación de sprints hasta el seguimiento del progreso y la visualización del trabajo, Jira Cloud se ha convertido en una herramienta popular para equipos que adoptan enfoques ágiles en su trabajo.

### 6.1.3. Métricas en Jira Cloud

Estos informes y métricas que se pueden obtener en Jira Cloud son fundamentales para tener una visión clara y objetiva del trabajo realizado, identificar áreas de mejora y tomar decisiones informadas. A continuación, se detallan algunos de los informes y métricas clave que ofrece Jira Cloud en el contexto ágil en el apartado *Reports*:

- **Sprint Burndown Chart (Gráfico de avance):** Jira Cloud proporciona una representación gráfica del progreso del equipo a lo largo del tiempo y ayuda a identificar si el equipo está en camino de completar el trabajo planificado para el sprint o el proyecto.

**Figura 32**

*Sprint Burndown Chart*



*Nota.* <https://support.atlassian.com/jira-software-cloud/docs/view-and-understand-the-burndown-chart/>

- **Velocity (Velocidad):** Jira Cloud calcula automáticamente la velocidad del equipo en función de las historias de usuario, tareas o puntos de historia completados en un sprint.

**Figura 33**

*Velocity Chart*



*Nota.* <https://www.techagilist.com/agile/jira/velocity-chart-in-jira-stabilize-and-improve-velocity/>

- **Cumulative flow diagram:** Especialmente útil para los equipos Kanban, ya que podría decirse que el flujo es la métrica más importante a considerar. Sin embargo, los equipos Scrum también pueden sufrir cuellos de botella, lo que hace que este informe sea útil para todos.
- **Cycle time report:** Como se indica en la documentación de Jira “Nuestro tiempo de ciclo es el tiempo que le toma a un problema pasar desde el primer compromiso hasta la última implementación de producción según el desarrollo y los datos de CI/CD.”
- **Control de calidad:** Jira Cloud permite realizar un seguimiento de la calidad del trabajo realizado por el equipo mediante métricas como los defectos reportados, las pruebas realizadas y los errores corregidos.

El resto de gráficos y métricas disponibles se personalizarán con el caso práctico.

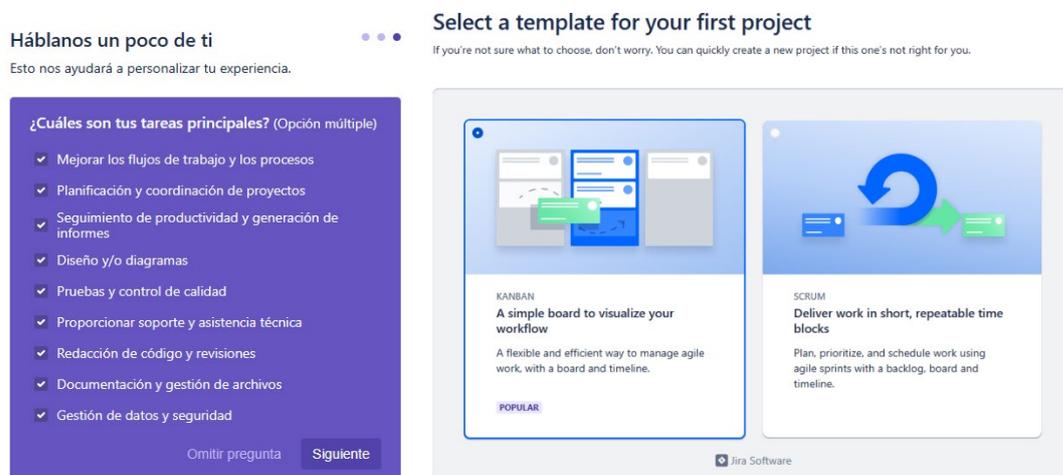
## 7. Desarrollo en Jira

### 7.1. Proyecto en Jira Cloud

Para la aplicación de las métricas que puede ofrecer la herramienta Jira Cloud, se crea un proyecto ejemplo siguiendo los siguientes pasos.

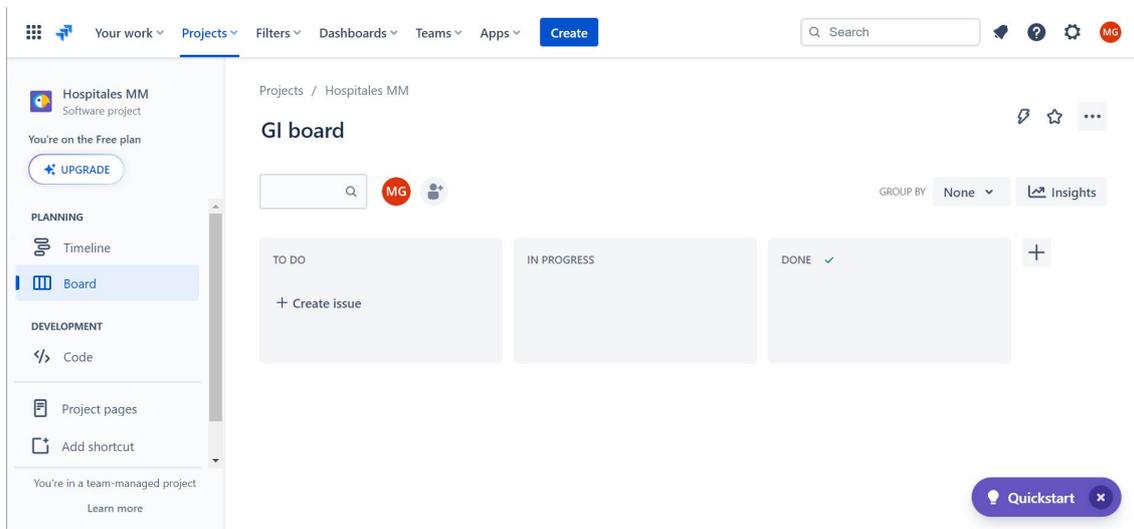
**Figura 34**

#### *Configuración proyecto Jira Cloud*



**Figura 35**

*Configuración proyecto Jira Cloud*



**Figura 36**

*Configuración proyecto Jira Cloud*

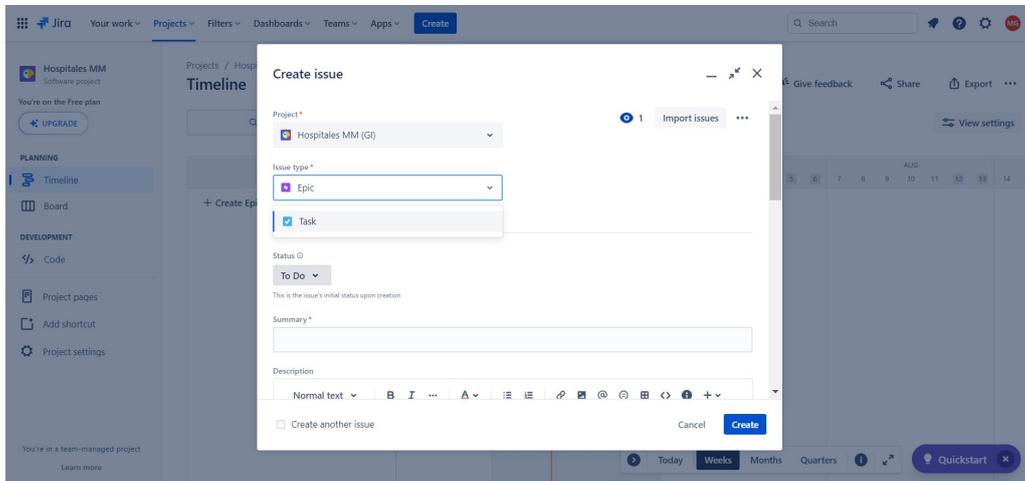


Figura 37

Configuración proyecto Jira Cloud

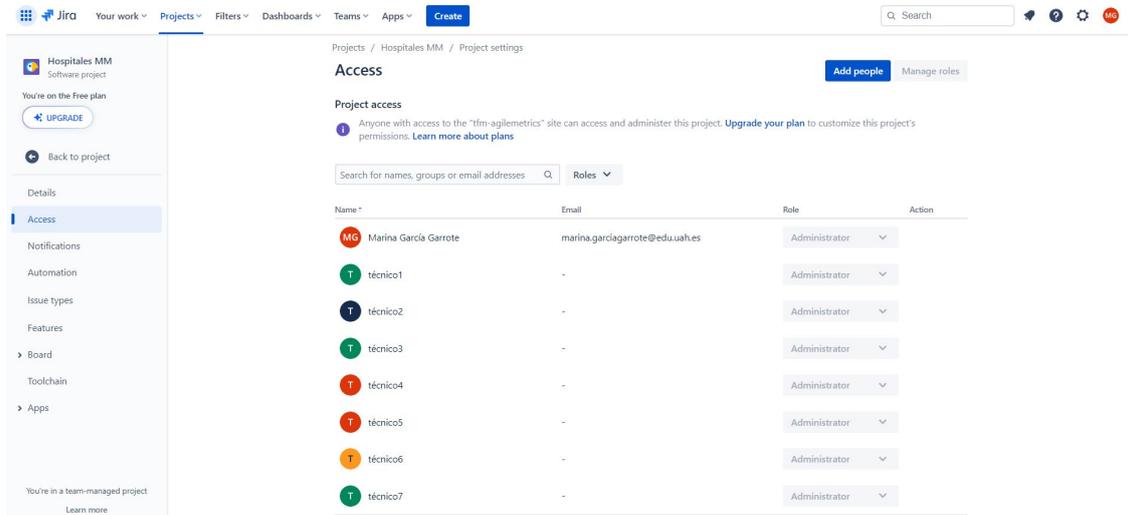
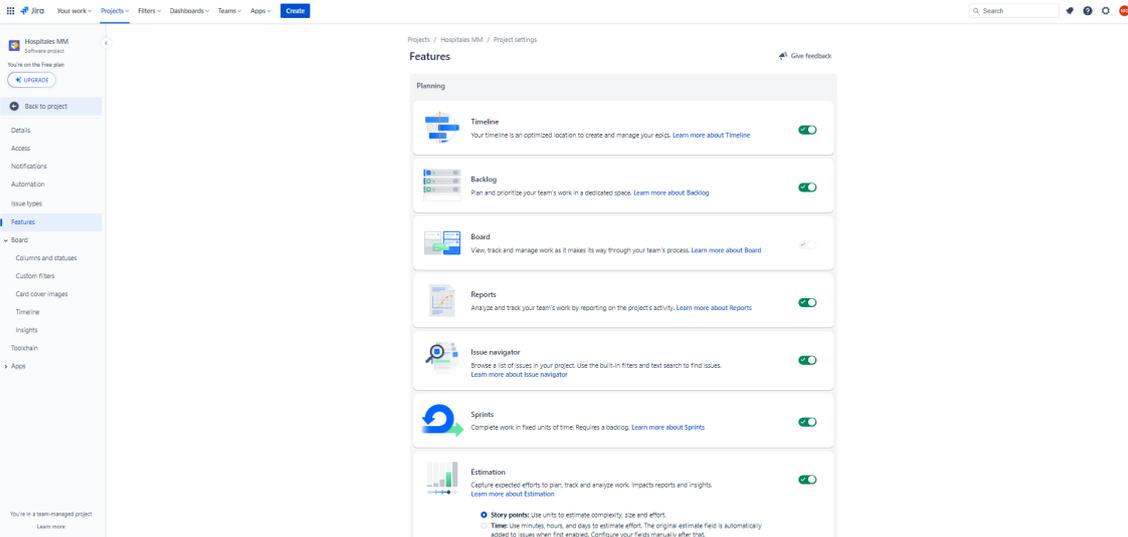


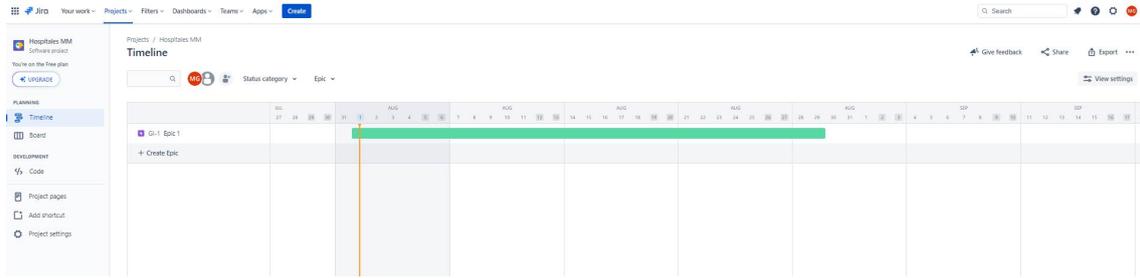
Figura 38

Configuración proyecto Jira Cloud. Habilitación de la opción "Reports"



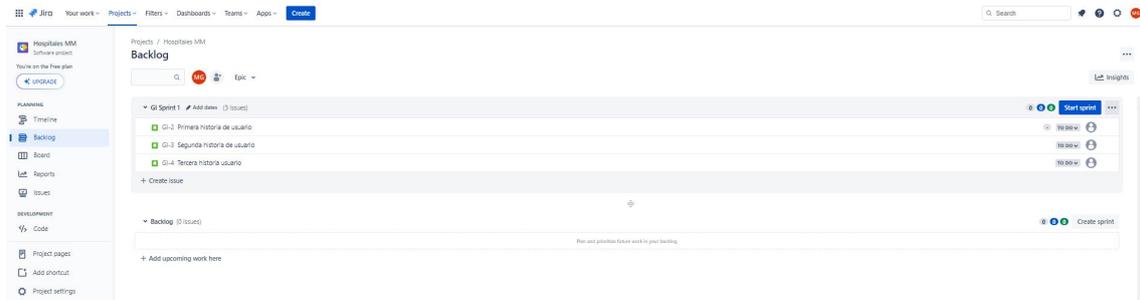
**Figura 39**

*Configuración proyecto Jira Cloud. Se crea la Epic 1*



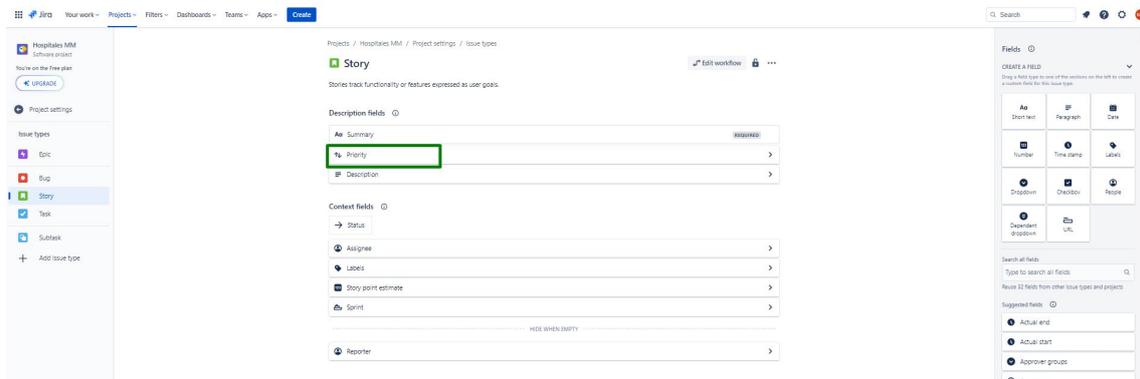
**Figura 40**

*Configuración proyecto Jira Cloud. Se crean las historias de usuario*



**Figura 41**

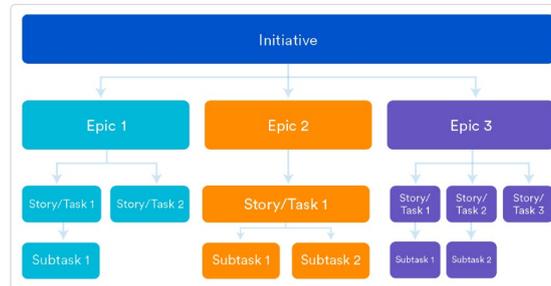
*Configuración proyecto Jira Cloud. Se añaden los campos necesarios a demanda*



La organización del trabajo se divide según el esquema general utilizado por la herramienta Jira. Como suele ser habitual, aunque puede definirse de muchas otras maneras teniendo en cuenta las necesidades de los equipos y proyectos, se dividirá el trabajo en una épica que contendrá varias historias de usuario, y a su vez estas contendrán subtareas.

**Figura 42**

*Organización de tareas de la iniciativa*



*Nota.* Rehkopf, M. (S.f.). Historias de usuario. Documentación Jira.

<https://www.atlassian.com/es/agile/project-management/user-stories>

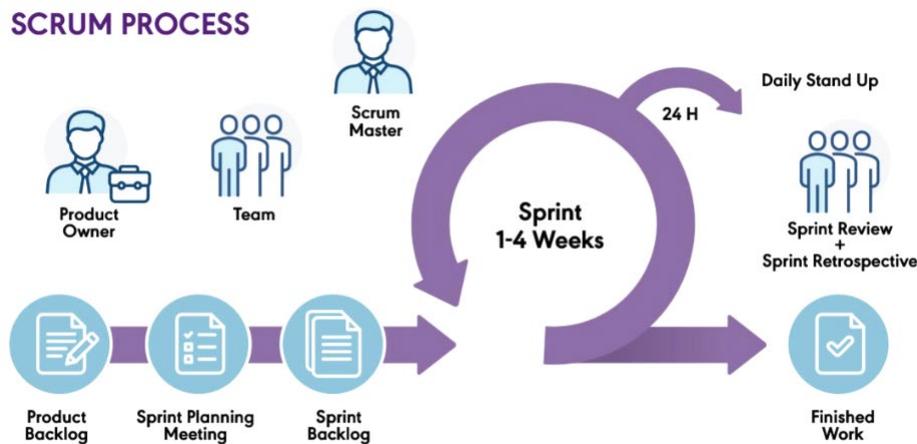
En primer lugar simular el desarrollo de un proyecto lo más parecido a uno real, se gestiona el mismo en 4 sprints de cuatro semanas. Así se podrán ver reflejadas las distintas métricas durante un periodo de secuencia según la metodología ágil lo más completo posible, como se muestra en la figura 40.

Se han creado un total 7 perfiles de desarrollo que llevarían a cabo las tareas generadas. Al generarse un perfil asociado a mi usuario, se considerará que ese usuario ejerce el rol de Scrum Master, no de técnico.

La jornada se considera intensiva, por lo que para el desarrollo de las historias de usuario los técnicos tienen para la estimación del trabajo cinco puntos de la escala Fibonacci, por cinco jornadas a la semana. Esto nos deja un total de puntos de ciento setenta y cinco a la semana para la estimación del trabajo realizado, teniendo en cuenta que igualmente se estimarán las subtareas asociadas teniendo un conteo total igual o inferior a la historia de usuario correspondiente (en este caso podrán utilizarse estimaciones que se salgan de la escala de Fibonacci, por ejemplo una historia de ocho puntos con dos subtareas de cuatro puntos cada una).

Figura 43

Proceso de desarrollo Scrum



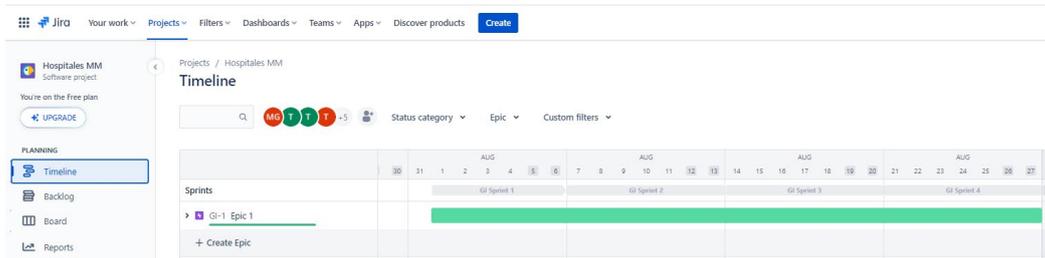
Nota. Pm-Partners. (2021). The Agile Journey: A Scrum overview. <https://www.pm-partners.com.au/the-agile-journey-a-scrum-overview/>

Para ello se tendrán en cuenta varias características que afectarán a los resultados en los distintos sprints.

- *Sprint 1:* Durante este sprint se considera que solo se encuentran trabajando dos técnicos, dado que cuatro de los restantes se encuentran de vacaciones y el quinto se encuentra de baja médica.  
Total puntos estimados: treinta puntos.
- *Sprint 2:* El técnico 7 sigue de baja médica, pero tiene previsión de incorporación el segundo día del sprint. Finalmente no se reincorpora.  
Total puntos estimados: ciento setenta y cinco puntos.
- *Sprint 3:* Durante este sprint se considera que todos los técnicos se encuentran en activo. Al tener retraso sobre el sprint anterior el equipo en la retrospectiva estima que debe poder abarcar la estimación completa incluyendo los atrasos que acarrear.  
Total puntos estimados: doscientos cuatro puntos.
- *Sprint 4:* Durante este sprint se considera que todos los técnicos se encuentran en activo. El equipo de desarrollo vuelve a tener retraso sobre el sprint anterior, por lo que esta vez consideran reducir la carga para poder solventar el retraso.  
Total puntos estimados: ciento cuarenta y cuatro puntos.

**Figura 44**

*Timeline del la Epic 1*



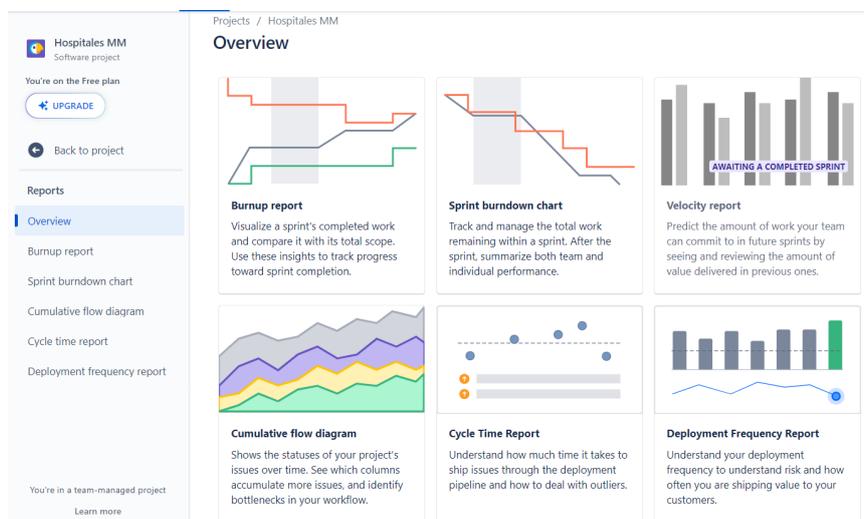
Se puede ver en la figura 44 el timeline desarrollado para los cuatro Sprints para el análisis.

## 7.2. Métricas de proyecto en Jira Cloud

Se generan todos los reportes por defecto disponibles en la herramienta. En algunos casos como se aprecia en la figura 45, algunas de las métricas solo están disponibles cuando se finaliza un Sprint, dos Sprints (como el caso de la velocidad, que es una métrica basada en la experiencia del primer sprint), o se visualizan en conjunto sin aparecer filtro por Sprint en la misma.

**Figura 45**

*Métricas generales de la opción “Reports”*



- Métricas Burnup por sprints:

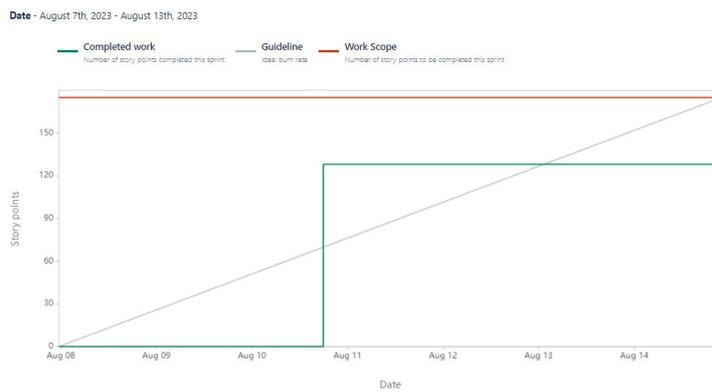
**Figura 46**

*Burnup Sprint 1*



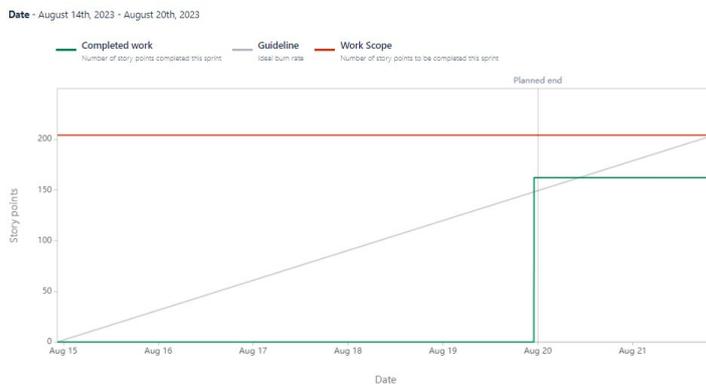
**Figura 47**

*Burnup Sprint 2*



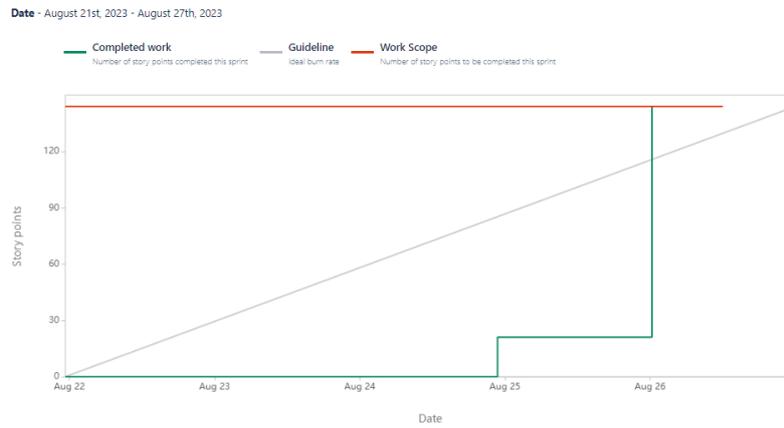
**Figura 48**

*Burnup Sprint 3*



## Figura 49

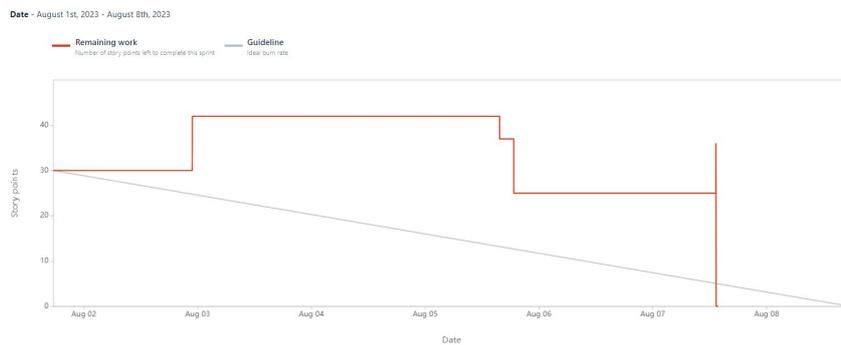
### Burnup Sprint 4



- Métricas Burndown por sprints:

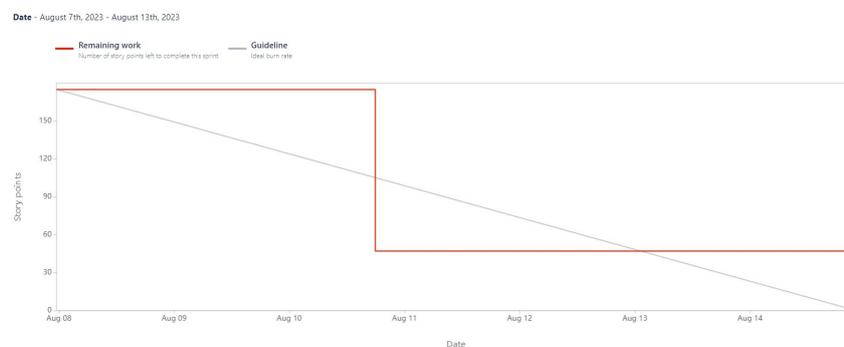
## Figura 50

### Burndown Sprint 1



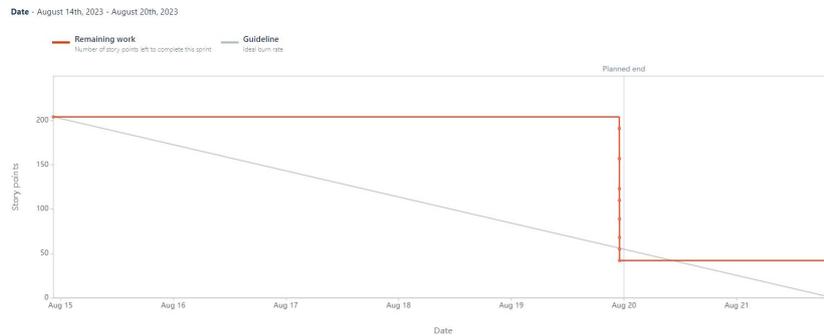
## Figura 51

### Burndown Sprint 2



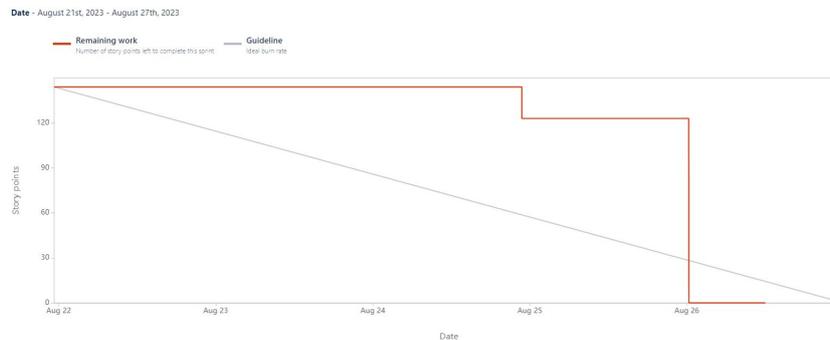
## Figura 52

### Burndown Sprint 3



## Figura 53

### Burndown Sprint 4



Como se puede ver en ambas métricas, en el primer Sprint se estimó por debajo de los cincuenta puntos para asegurar que los dos únicos técnicos pudieran tener la capacidad de realizar otras tareas de gestión o incidencias que pudieran surgir, durante el sprint se han recalculado algunas estimaciones. Pese a todo el equipo a completado las tareas asignadas durante el mismo satisfactoriamente.

En en el segundo y tercer Sprint se ve como pese a ser más técnicos, en ambos casos quedaron story points sin completar, es decir tareas que se arrastraron al tercer y cuarto Sprint. Se puede ver mediante ambas líneas que no llegan al final de Sprint conjuntamente.

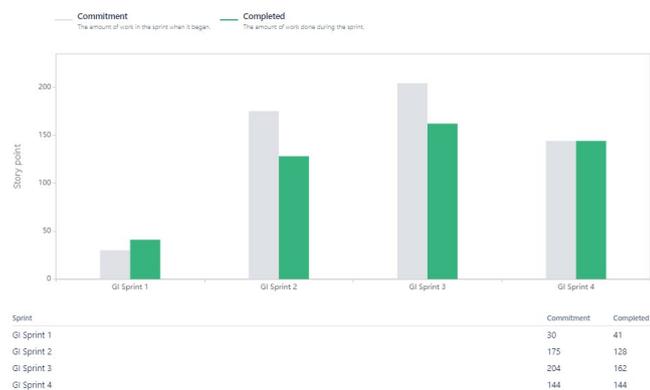
En el cuarto Sprint una estimación inferior llevó a que fuera posible terminar todas las tareas a tiempo. Es por esto que ambas líneas se unen al final del Sprint.

También se puede apreciar como se trata de tareas “largas” dado que en la mayoría del tiempo analizado, las tareas se alejan de la diagonal, lo que significa que se tarda bastante tiempo en cerrarlas. En caso de que fueran tareas con menos story points el avance del tiempo completado hubiera acompañado a la diagonal durante los Sprints en el Burnup.

- **Métrica Velocity report:**

**Figura 54**

*Velocity Report*



En este caso se puede ver con claridad que durante el primer Sprint se hizo una planificación inferior, dado que había menos técnicos disponibles, pero pese a ello se consiguió terminar más story points de los planificados.

Durante el segundo y tercer Sprint la velocidad del equipo se vio disminuída, probablemente por los problemas asociados a las ausencias. Es por esto que se aprecia que la cantidad de story points completados es inferior a los estimados.

Esta métrica puede tomarse como referencia a la hora de comprobar que en el cuarto Sprint ha de realizarse una disminución en la estimación y que de esta manera se pueda equilibrar y relanzar al equipo.

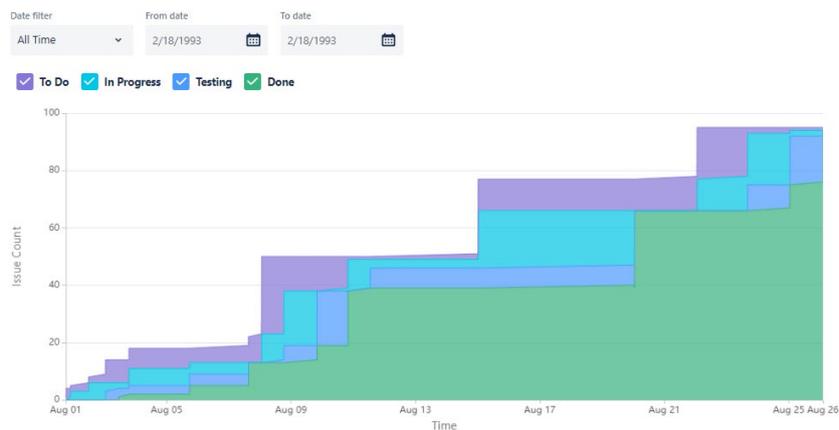
En el cuarto Sprint se puede ver como el equipo completó satisfactoriamente todas las tareas. La velocidad por tanto estaría adecuada entre 144 y 175 story points, aunque seguramente se acerque más a la segunda cifra, dado que las estimaciones no tienen en cuenta posibles bajas, al contrario de las ausencias planificadas. Se tiene que tener en cuenta para el máximo 175 story points y no los 162 story points que son el máximo completado, pese a las ausencias. Es por ello que una métrica como por ejemplo el WIP, puede no establecerse desde el inicio y basarse en la experiencia de los primeros sprints.

- **Cumulative flow diagram**

En el caso del diagrama de flujo acumulado, se puede hacer un seguimiento de los estados por los que han pasado las tareas de la épica. La distancia entre las columnas indica el tiempo que ha tardado una tarea en pasar de un estado a otro.

**Figura 55**

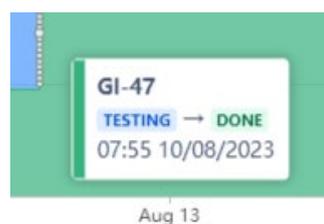
*Cumulative flow diagram*



Cada uno de los puntos de la gráfica permite además visualizar el cambio de estado de cada tarea en concreto, indicándose el ID de la tarea, así como el momento del cambio.

**Figura 56**

*Cumulative flow diagram. Detalle transición Issue del estado Testing a Done*



Se puede apreciar como del 10 al 15 de agosto un mayor número de tareas fueron llevadas al estado In Progress, coincidiendo con un final y un principio de Sprint. En el primer caso indica que en el segundo Sprint no se comenzaron muchas tareas desde el inicio, pero si a mediados y finales del Sprint. Y por el contrario en el tercer Sprint las tareas se comenzaron al inicio del mismo. Esto debe analizarse en conjunto con el tamaño de las tareas, dado que debe tenerse en cuenta que un número inferior de story points por tarea podría llevar a que los técnicos puedan abarcar más de una tarea al mismo tiempo.

Puede verse también como en algunos momentos del desarrollo, algunas tareas apenas han pasado por el estado de test. Tendría que comentarse con el equipo si esos cambios son debidos a que se cambia con retraso el estado en la herramienta, o a que se desea llegar con el mayor número de tareas finalizadas al final del Sprint y por ello se invierte menos tiempo en testing. De ser así podría suponer un problema a futuro dado que pueden aparecer Bugs debidos a falta de pruebas en las tareas.

Se puede comprobar mediante el filtrado por *Testing* del diagrama, como se puede ver en la figura 57.

### Figura 57

*Cumulative flow diagram. Filtro estado Testing*

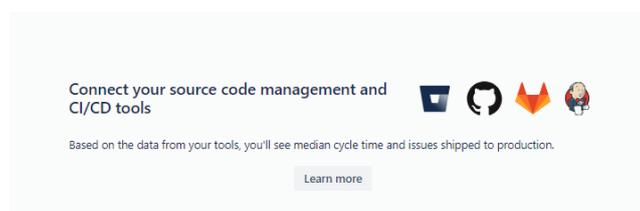


- **Cycle time report.**

Como se trata de una métrica condicionada a despliegues en el caso de Jira, para este estudio no se muestran datos para esta métrica.

### Figura 58

*Detalle de la no disponibilidad de la métrica Cycle time report*

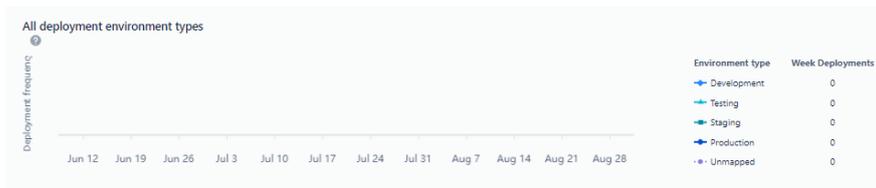


- **Deployment frequency report.**

Al igual que sucede con la métrica de Cycle time, dado que no se han realizado despliegues durante el estudio esta gráfica aparece sin datos asociados.

**Figura 59**

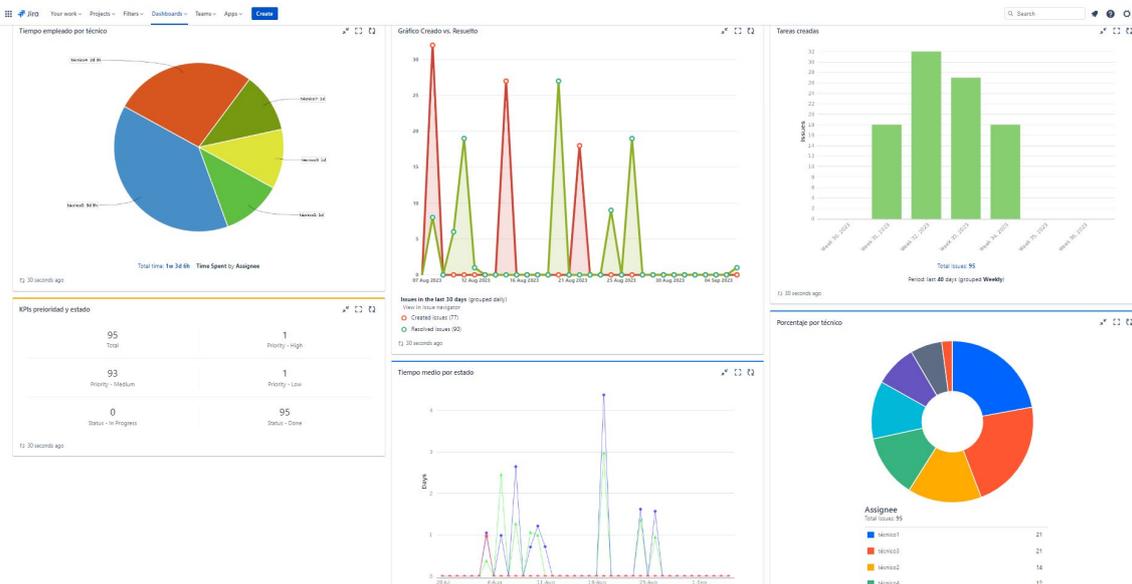
*Detalle de la no disponibilidad de la métrica Deployment frequency report*



Junto con las métricas primarias ofrecidas en el apartado *Reports* por Jira, se realiza un Dashboard personalizado con widgets que incluye la herramienta, para poder obtener más datos que permitan una evaluación más completa del proyecto.

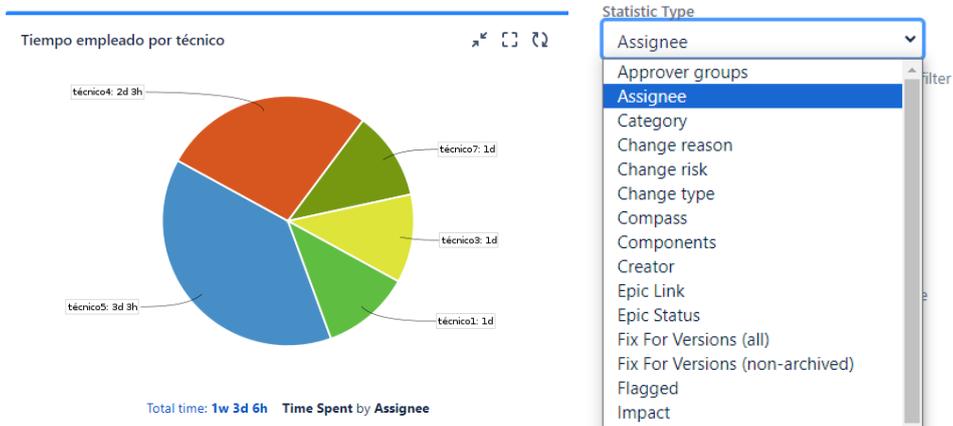
**Figura 60**

*Dashboard personalizado en Jira Cloud*



**Figura 61**

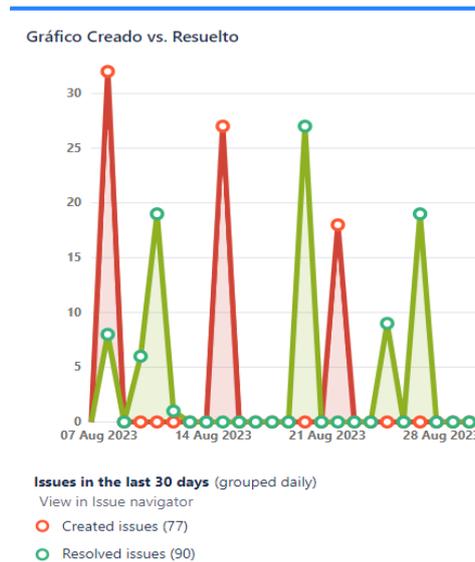
*Gráfica de tiempo empleado por técnico y su configuración. Dashboard personalizado en Jira Cloud*



En primer lugar se configura un gráfico Workload Pie Chart, seleccionando en este caso el tiempo invertido por técnico.

**Figura 62**

*Gráfica de Creado vs. Resuelto. Dashboard personalizado en Jira Cloud*



En esta gráfica se puede ver como progresivamente hay picos de tareas creadas, con su posterior pico de tareas resueltas, prácticamente coincidiendo con la secuencia de los sprints. Se puede observar como las tareas creadas han ido disminuyendo para reajustar la estimación del equipo.

**Figura 63**

*Gráfica de tareas creadas. Dashboard personalizado en Jira Cloud*



Generado mediante el gráfico *Recently Created Chart*, este gráfico que permite configurar la periodicidad con la que se muestran los datos, en este caso semanal, así como el espacio de tiempo en días a visualizar. Permite ver cuantas tareas se han creado en los cuatro sprints, visualizando como se fue reduciendo el número al verse impactado el número de horas disponibles para estimar por falta de personal.

**Figura 64**

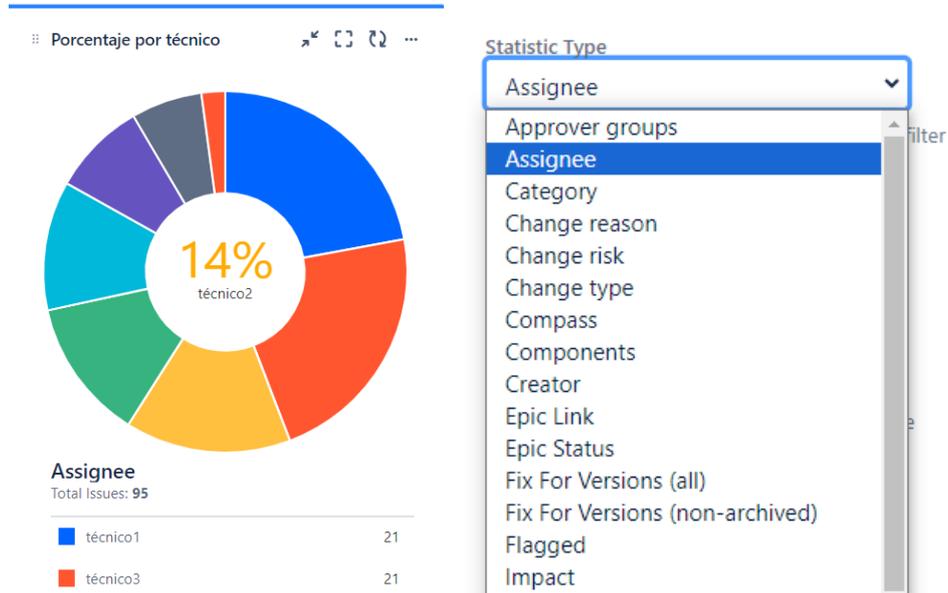
*Gráfica de tiempo medio por estado. Dashboard personalizado en Jira Cloud*



Creada mediante el gráfico *Average Time in Status* esta gráfica permite ver cual sería el tiempo medio que han pasado las tareas por ejemplo en desarrollo o en testing.

**Figura 65**

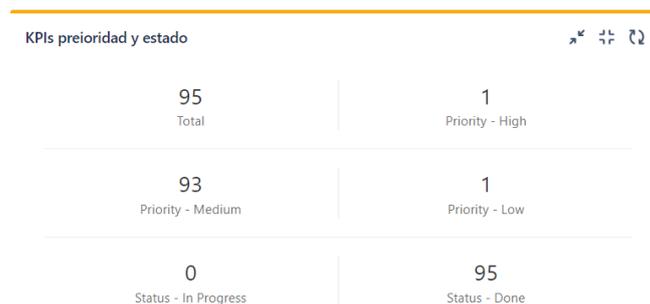
*Gráfica de porcentaje de tareas realizadas por técnico y su configuración. Dashboard personalizado en Jira Cloud*



Generado mediante un *Pie Chart* este gráfico permite ver la proporción de tareas llevadas a cabo por técnico.

**Figura 66**

*Gráfica de tiempo empleado por técnico. Dashboard personalizado en Jira Cloud*



Todas las métricas creadas dentro del dashboard pueden calendarizarse de manera individual permitiendo un refresco de 15 minutos o superior

## 8. Desarrollo de herramienta para elección de métricas

Para poder realizar una mejor toma de decisiones en un negocio es importante elegir las métricas adecuadas según el proyecto o los aspectos más relevantes a analizar. El siguiente paso sería la recopilación de datos de las métricas y su posterior modelado. Por último tendría que realizarse un análisis de datos que finalmente llevaría a crear una estrategia de toma de decisiones.

Para poder elegir las métricas de manera más adecuada según el proyecto o la empresa se va a desarrollar un prototipo de herramienta que ayudaría a elegir las métricas que forme parte de este proceso de mejora de toma de decisiones.

Para este estudio se va a generar la herramienta utilizando software de desarrollo Low Code. Las aplicaciones "Low Code" y "Non Code" son aplicaciones que pueden desarrollarse sin que sea necesario tener un perfil técnico ni grandes conocimientos de desarrollo. Se ha elegido este tipo de herramienta porque supone un gran avance en cuanto al desarrollo por su curva de aprendizaje muy corta y porque permite a los desarrolladores generar cambios y personalizaciones con poco esfuerzo.

Se ha generado un chatbot con preguntas clave que ayudarán a los perfiles de gestión a escoger la métrica más adecuada para su análisis según los criterios estudiados en este trabajo. En este caso la herramienta se desarrolla con Power Virtual Agents, aplicación parte de la suite de Power Platform de Microsoft.

### 8.1. Elección preguntas para el desarrollo de la herramienta

Tras elegir el tipo de herramienta, el paso fundamental es elegir las preguntas asociadas que permitirán tomar una mejor elección de métrica para cada caso.

Para mostrar que el desarrollo se podría aplicar a distintos campos, ampliando el árbol de decisión, se tendrán en cuenta de manera inicial el tipo de proyecto, pudiendo ser de entre estos tres tipos:

- Desarrollo de Software
- Marketing

- Manufactura

Se desarrollarán únicamente las en las asociadas al desarrollo de software que son las que aplican a este estudio.

#### 1. **Tipo de Proyecto.**

Pregunta: *¿Qué tipo de proyecto estás llevando a cabo? (Desarrollo de software, marketing, manufactura, etc.)*

#### 2. **Método Ágil.**

Pregunta: *¿Qué metodología ágil estás utilizando? (Scrum, Kanban, XP, etc.)*

- **Métricas Relacionadas:**

La metodología ágil que estás utilizando influye en las métricas que son más relevantes.

Por ejemplo, en Scrum, la Velocidad del equipo y el Tiempo promedio de ciclo son métricas que se alinean bien con la estructura de iteraciones y entregas regulares.

En Kanban, el enfoque podría ser más fluido, por lo que métricas como el Tiempo promedio de ciclo y la Eficiencia en el uso de recursos pueden ser esenciales.

#### 3. **Frecuencia de Entregas.**

Pregunta: *¿Con qué frecuencia realizas entregas o iteraciones? (Semanal, mensual, etc.)*

- **Métricas Relacionadas:**

La frecuencia de entregas influirá en las métricas que son más significativas. Si las entregas son frecuentes (por ejemplo, semanalmente), las métricas relacionadas con la velocidad y la calidad de entrega, como la Velocidad del equipo, Tiempo de entrega y el Tiempo promedio de ciclo, serán cruciales para evaluar la eficacia del proceso. Si las entregas son menos frecuentes, es posible que quieras centrarte en métricas de calidad y satisfacción del cliente a largo plazo.

#### 4. **Enfoque de Calidad.**

Pregunta: *¿Cómo de importante es la calidad del producto final? (Muy alta, normal, baja).*

- **Métricas Relacionadas:**

Métricas que evalúan la calidad del producto, como Número de errores por iteración, Porcentaje de tareas de tipo 'Non-Bug' y Change Failure Rate. Son métricas de especial importancia si la calidad es un punto muy relevante en la gestión del proyecto.

#### 5. **Criticidad.**

Pregunta: *¿Se trata de un producto muy crítico?*

- **Métricas Relacionadas:**

Métricas que evalúan la criticidad del producto, como Mean Time to Recovery, Tiempo promedio de corrección de tareas, Errores activos, Test Coverage, Frecuencia de despliegue, Tiempo medio de corrección de la tarea.

Son métricas de especial importancia si se trata de productos cuya criticidad es muy alta, como productos con un gran número de usuarios, o transaccionales como aplicaciones de banca.

## 6. **Experiencia del equipo.**

Pregunta: *¿Se trata de un equipo con experiencia?*

- **Métricas Relacionadas:**

Estas métricas pueden mostrar datos relevantes en caso de que se cuente con un equipo poco experimentado, lo que puede llevar a realizar nuevas contrataciones o ampliar el equipo con personal más senior para evitar cuellos de botella y con ello retrasos en las entregas. En este caso métricas que van asociadas a la velocidad del equipo y la distribución de trabajo pueden ser relevantes aquí. También el número de tareas bloqueadas, Tiempo medio de corrección de la tarea, Change Failure Rate y Densidad de defectos.

## 7. **Rendimiento.**

Pregunta: *¿Te preocupa el rendimiento de tu equipo?*

- **Métricas Relacionadas:**

Estas métricas permiten conocer el rendimiento del equipo para ver si éste está siendo eficaz. Tiempo de testing, Gráfico de velocidad vs velocidad planificada, Throughput, Tiempo medio de corrección de la tarea.

## 8. **Avance Sprint.**

Pregunta: *¿Quieres saber si llegas a tiempo al objetivo del Sprint?*

- **Métricas Relacionadas:**

Estas métricas y gráficos ayudan a conocer el estado actual del Sprint y ver si la entrega de todas las tareas es factible al final de este. Se incluyen: Gráficos Burn-down y Burn-up, Diagrama de flujo acumulado, Número de tareas sin asignar, Gráfico parking lot.

## 9. **Gestión alcance.**

Pregunta: *¿Necesitas saber si el trabajo pendiente está creciendo o se está progresando en el desarrollo?*

- **Métricas Relacionadas:**

Para poder llevar un control en caso de que el cliente quiera incluir cambios que puedan afectar en gran medida al avance del proyecto, o si hay tareas que pueden ser descartadas es necesario controlar si el alcance se está viendo afectado.

Entre las métricas asociadas al alcance se pueden incluir: Tareas obsoletas, Scope Creep, Acciones de retrospectiva completadas y Tareas Activas fuera del Sprint actual.

#### 10. Estimación.

Pregunta: *¿Quieres manejar la previsión de desarrollo de tu equipo?*

- **Métricas Relacionadas:**

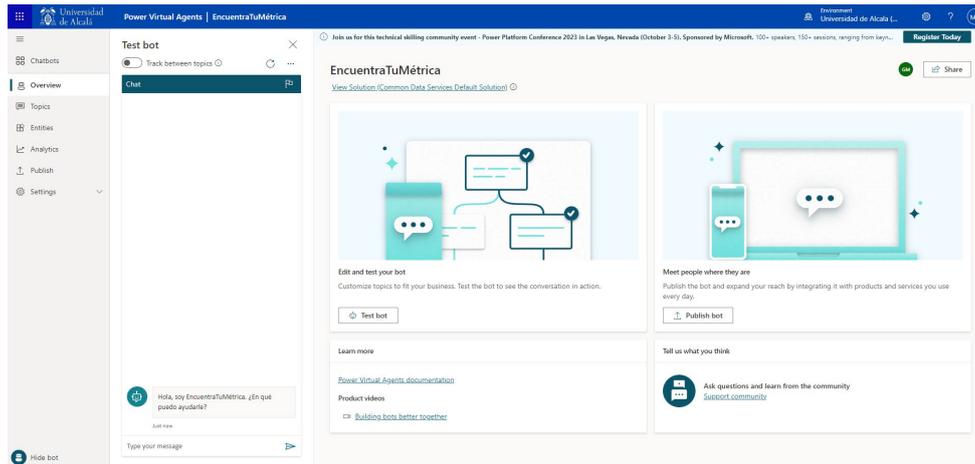
Estas métricas permiten tener una visión general sobre la estimación que está realizando el equipo, y la tendencia que tendrá a futuro el desarrollo del proyecto. Entre otras se pueden incluir: Mean Magnitude Relative Error, Puntos de historia, Suma total de esfuerzo estimado por proyecto por desarrollador, Velocity Trend, Gráfico de velocidad vs velocidad planificada.

## 8.2. Desarrollo

Para comenzar con el desarrollo con Power Virtual Agents, se crea un nuevo ChatBot que tendrá de nombre EncuentraTuMétrica.

**Figura 67**

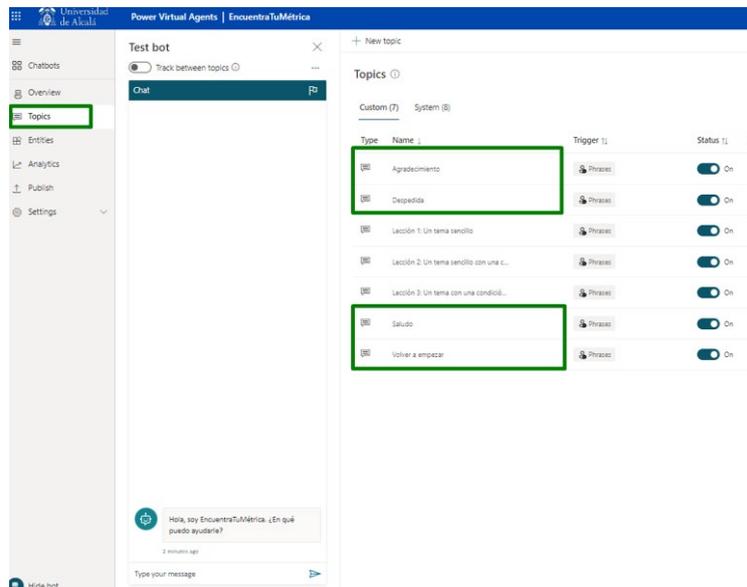
*Menú principal de Power Virtual Agents, chatbot EncuentraTuMétrica*



En primer lugar se configuran y personalizan los menús generales dentro del apartado topics personalizando los topic que vienen por defecto: Saludo, Despedida, Volver a empezar y agradecimiento

**Figura 68**

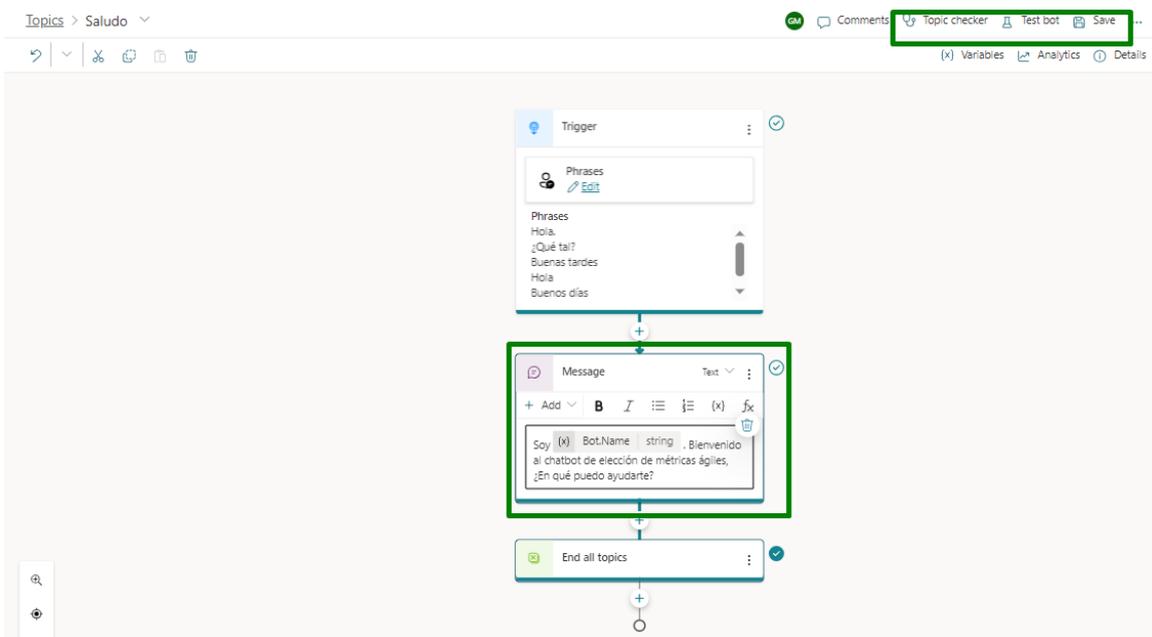
*Topics, chatbot EncuentraTuMétrica*



Para ello entrando en cada Topic, simplemente se modifica en el apartado Mensaje, el contenido. Junto con el botón de salvar cambios, se encuentra a su izquierda la opción “Test bot”, que permite depurar a demanda las modificaciones realizadas.

**Figura 69**

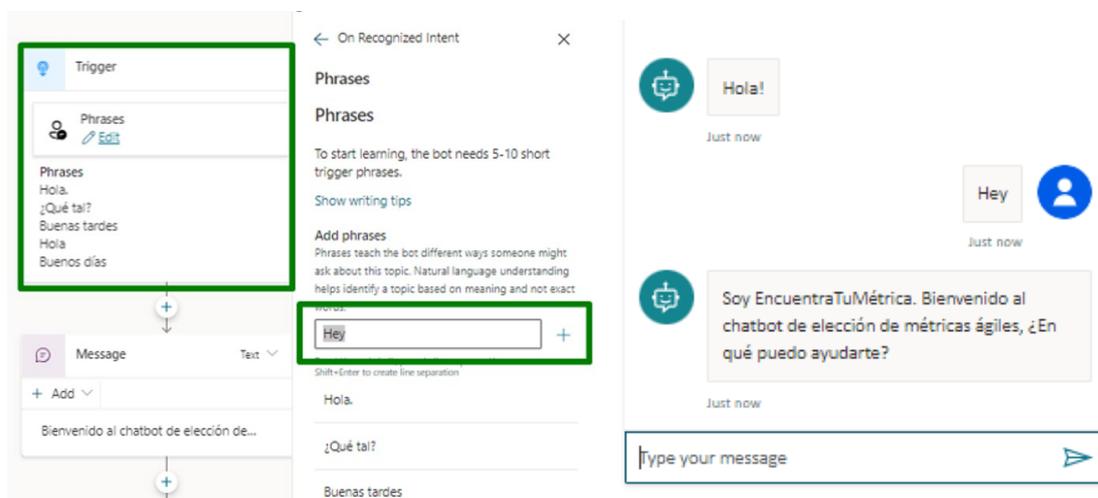
*Edición del topic Saludo, chatbot EncuentraTuMétrica*



Cada conversación cuenta con un Trigger opcional que son las palabras clave con las cuales reaccionará cada topic. En cada Trigger se pueden personalizar estas palabras clave.

**Figura 69**

*Edición del topic Saludo, chatbot EncuentraTuMétrica*



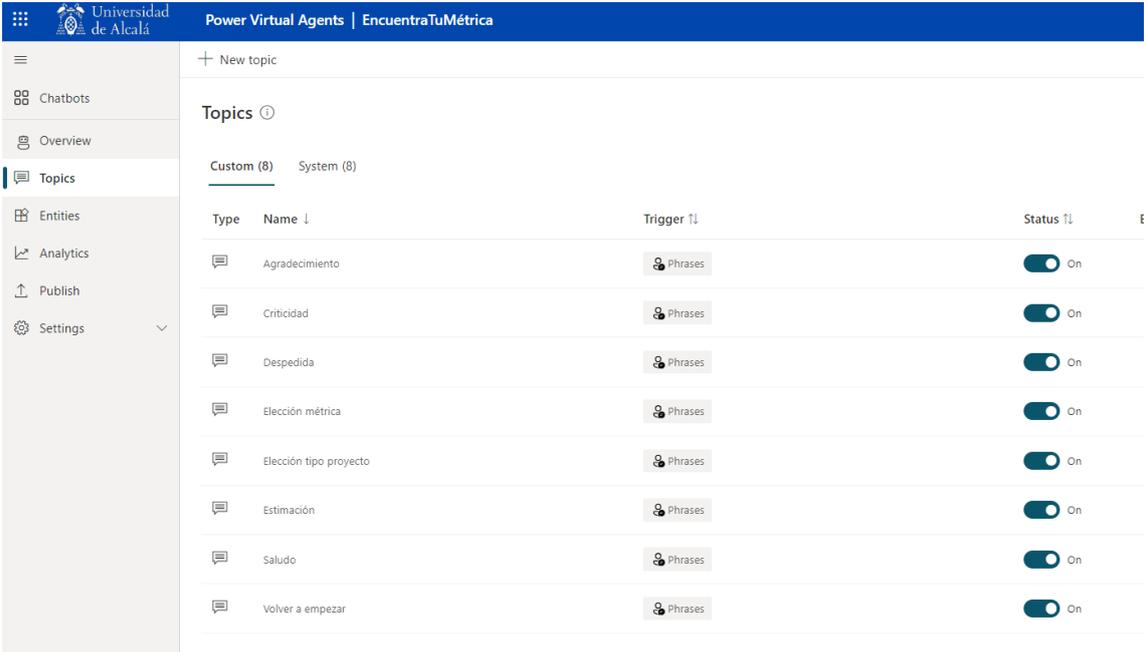
Se desarrollan los siguientes topics personalizados para este estudio:

- Criticidad
- Estimación
- Elección tipo proyecto
- Elección métrica

Tras la creación y personalización de todos los topics necesarios, quedan de la siguiente manera.

**Figura 71**

*Listado topics, chatbot EncuentraTuMétrica*



The screenshot shows the 'Topics' section of the Power Virtual Agents interface. The left sidebar contains navigation options: Chatbots, Overview, Topics (selected), Entities, Analytics, Publish, and Settings. The main area displays a list of custom topics. The table below represents the data shown in the screenshot.

Type	Name ↓	Trigger ↓	Status ↑
Custom	Agradecimiento	Phrases	On
Custom	Criticidad	Phrases	On
Custom	Despedida	Phrases	On
Custom	Elección métrica	Phrases	On
Custom	Elección tipo proyecto	Phrases	On
Custom	Estimación	Phrases	On
Custom	Saludo	Phrases	On
Custom	Volver a empezar	Phrases	On

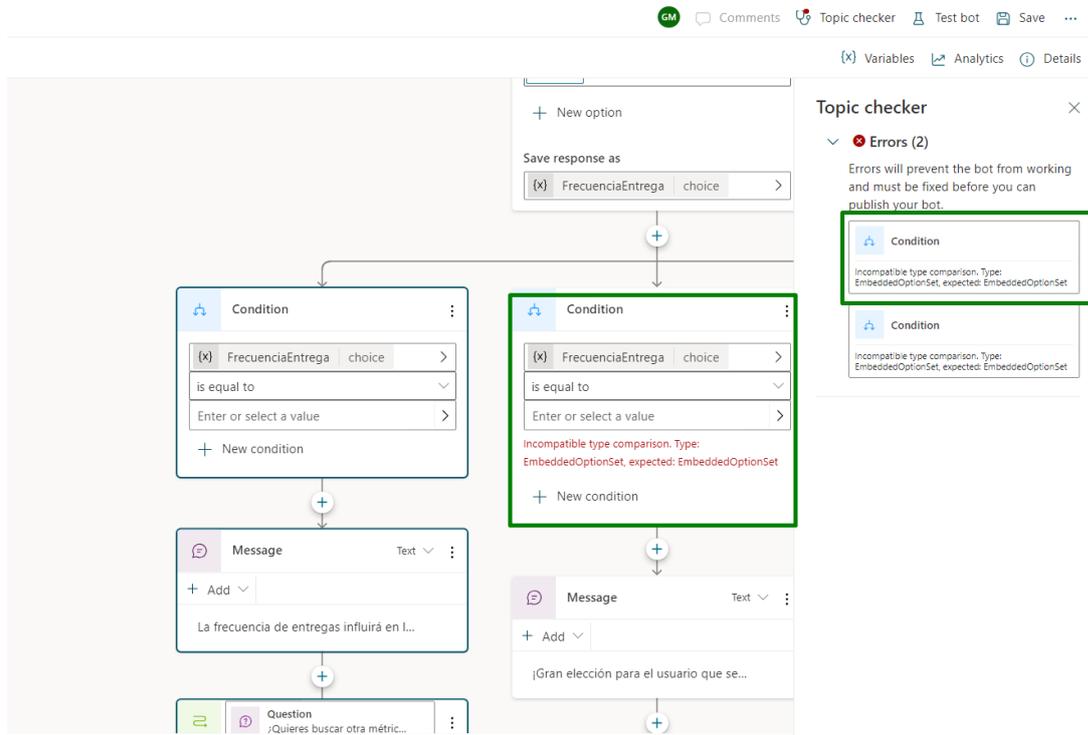
En lugar de realizar todo el flujo en un único topic, se opta por segregarlo en varios para facilitar su edición y lectura y también para comprobar la sencillez del proceso de poder realizar una conversación que pueda moverse de un topic a otro.

En cada paso se generan las variables necesarias para cada pregunta, que servirán como origen posterior en cada condición.

En caso de encontrarse errores durante el desarrollo Virtual agents no te permite salvar los cambios realizado. En el botón Topic checker aparecen los errores y seleccionándolos te lleva directamente al elemento que causa el error.

**Figura 72**

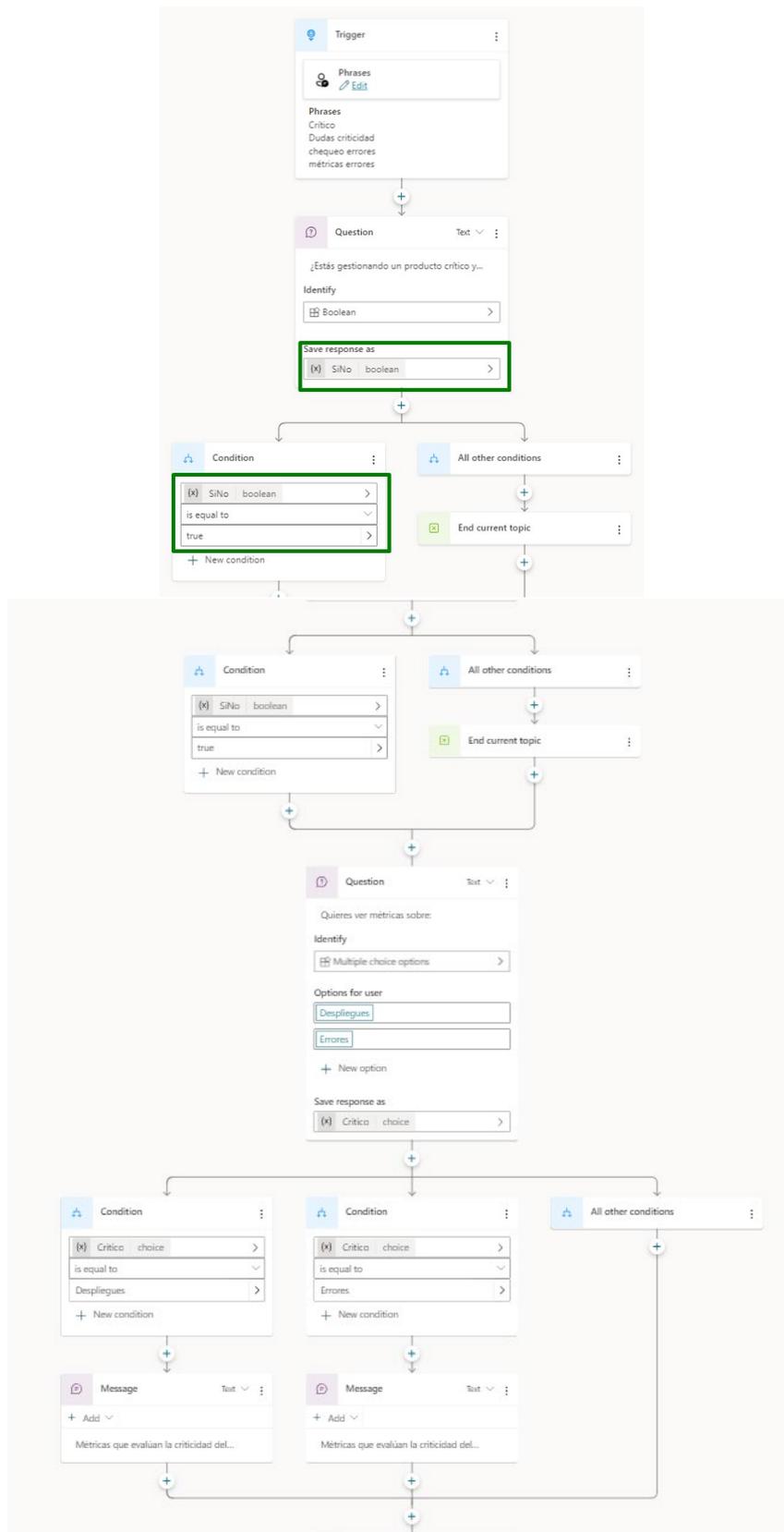
*Mensaje de error de compilación en el desarrollo de topic*



Siguiendo las preguntas expuestas en el anterior punto, se van creando a modo de diálogo para el chatbot en distintos pasos como se puede ver en las figuras 73 a 79.

Figura 73

Topic Criticidad , chatbot EncuentraTuMétrica



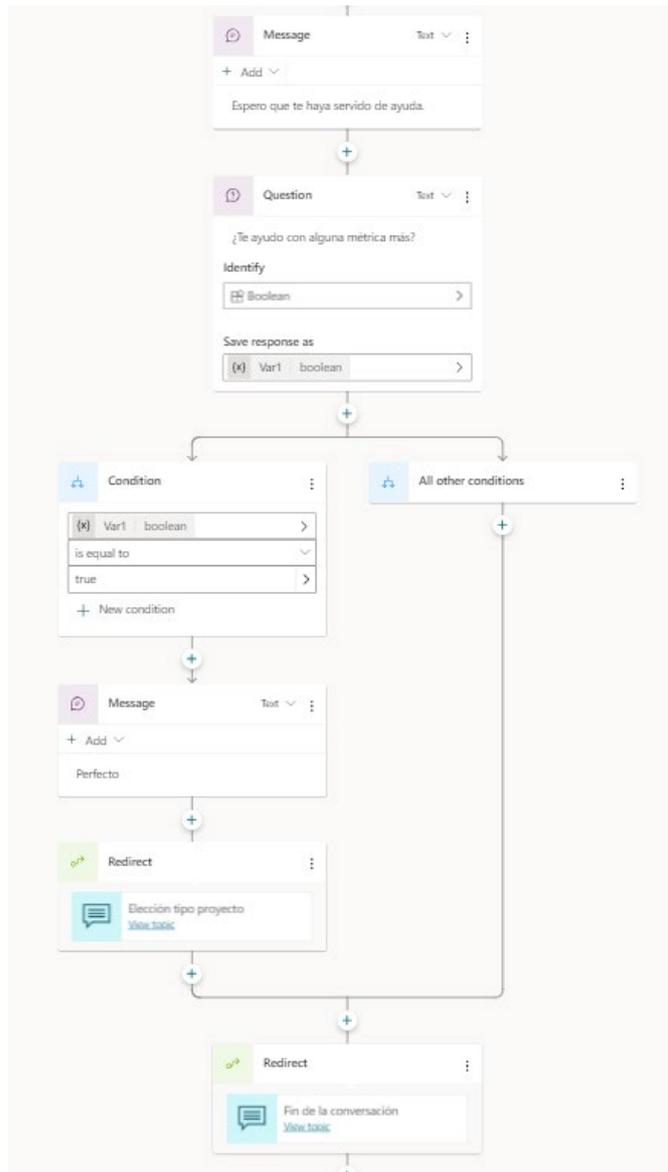


Figura 74

Topic Estimación , chatbot EncuentraTuMétrica

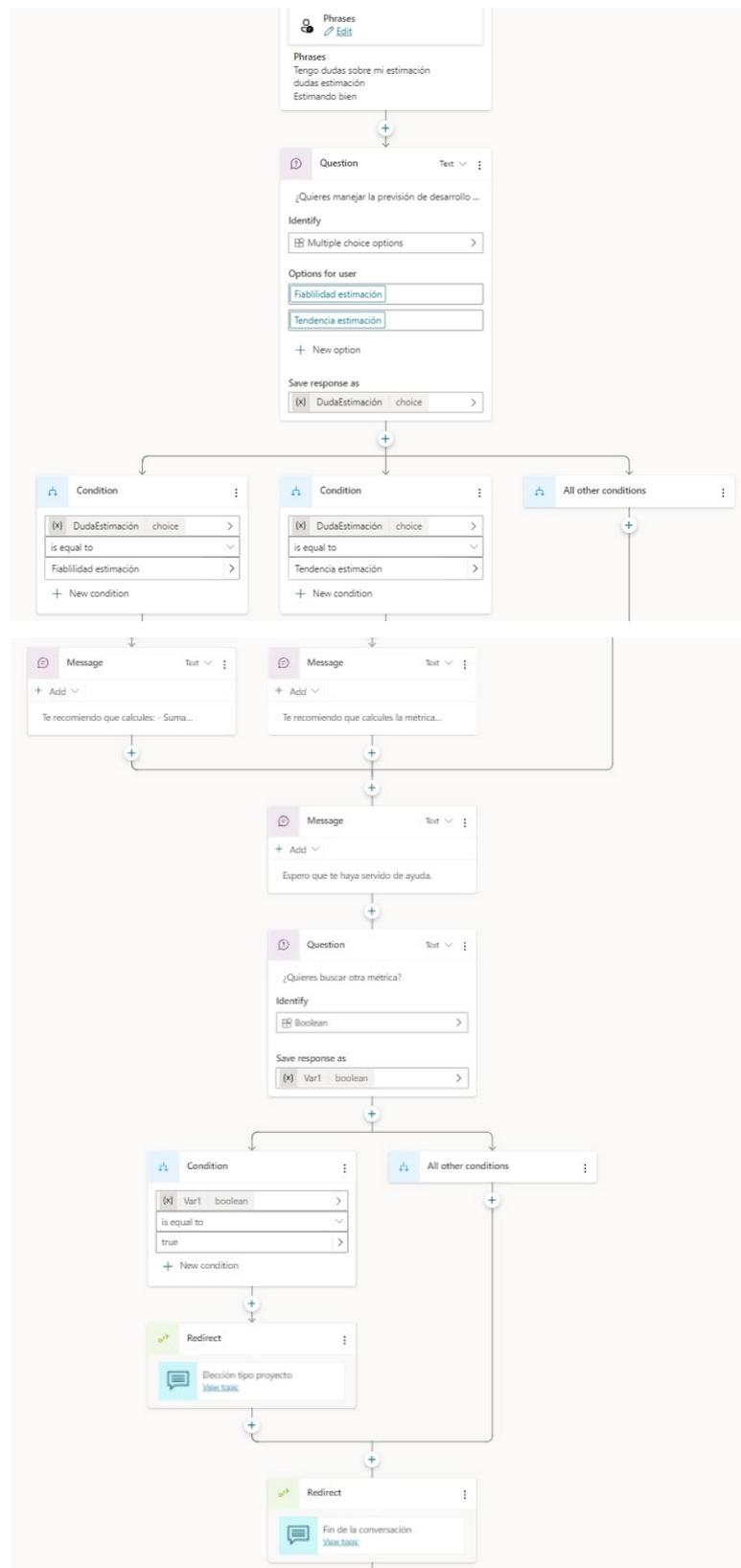


Figura 75

Topic Elección tipo proyecto , chatbot EncuentraTuMétrica

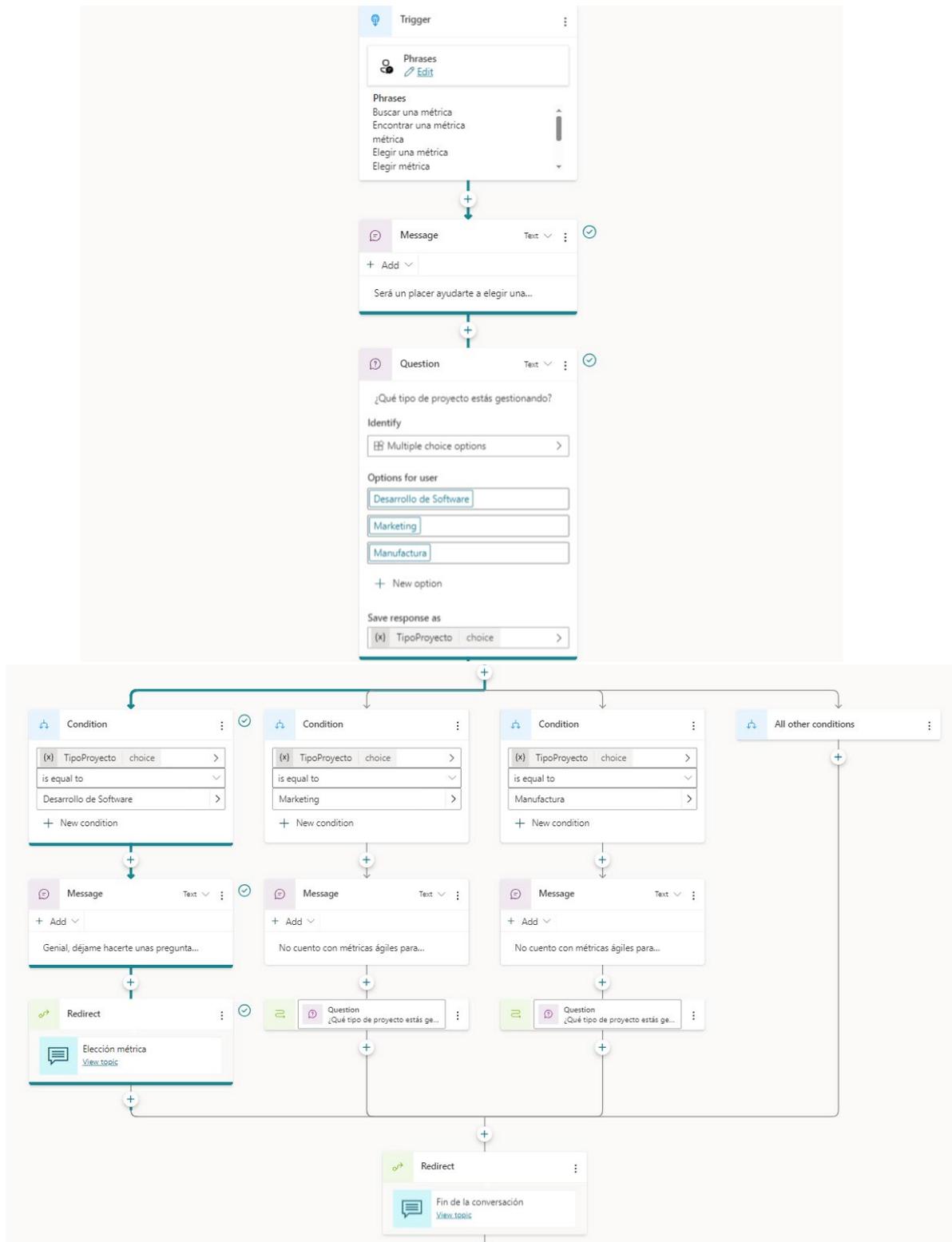
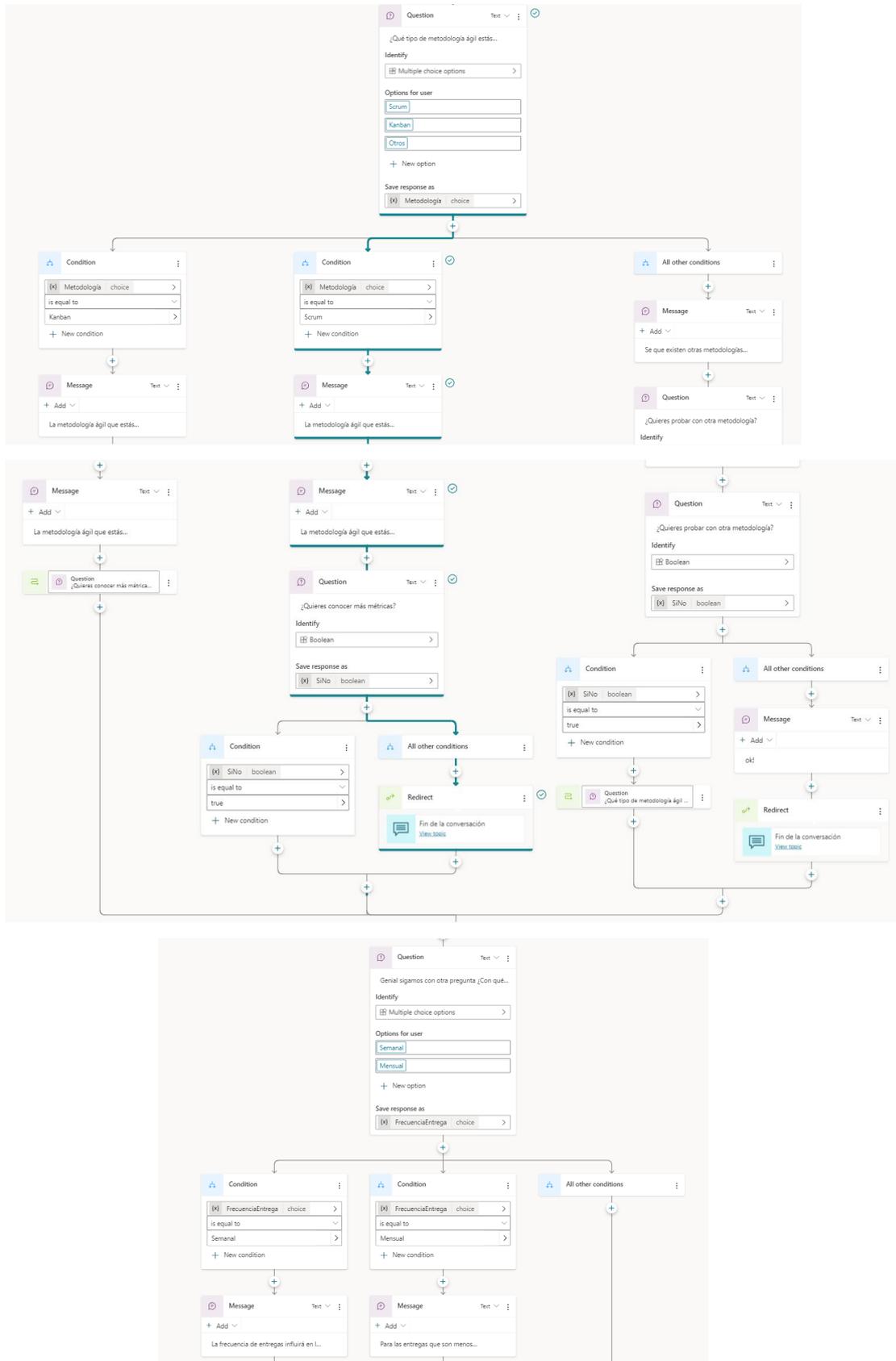
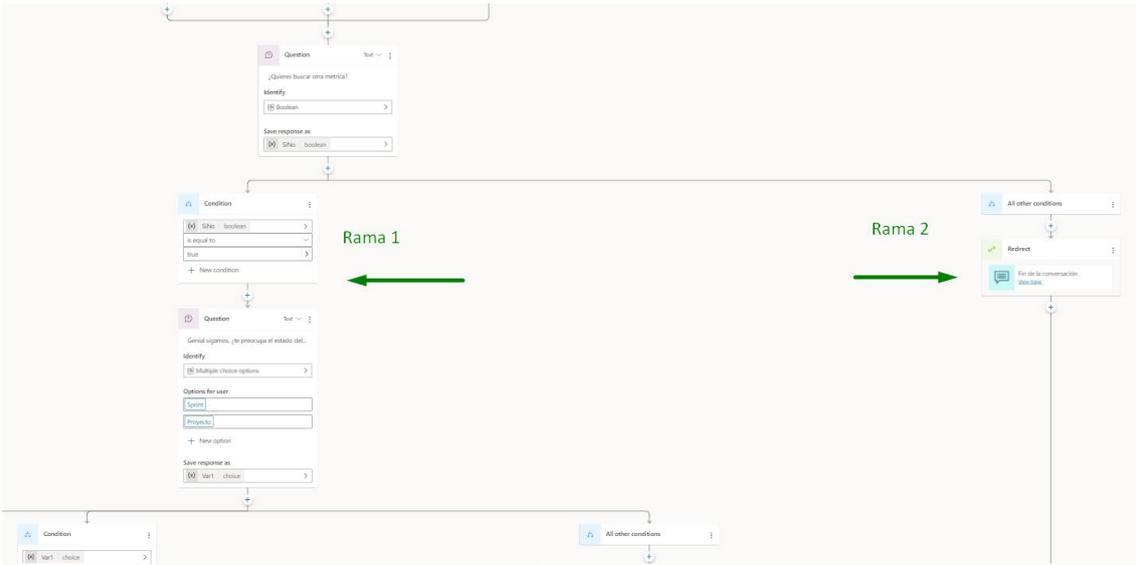


Figura 76

Topic Elección métrica , chatbot EncuentraTuMétrica

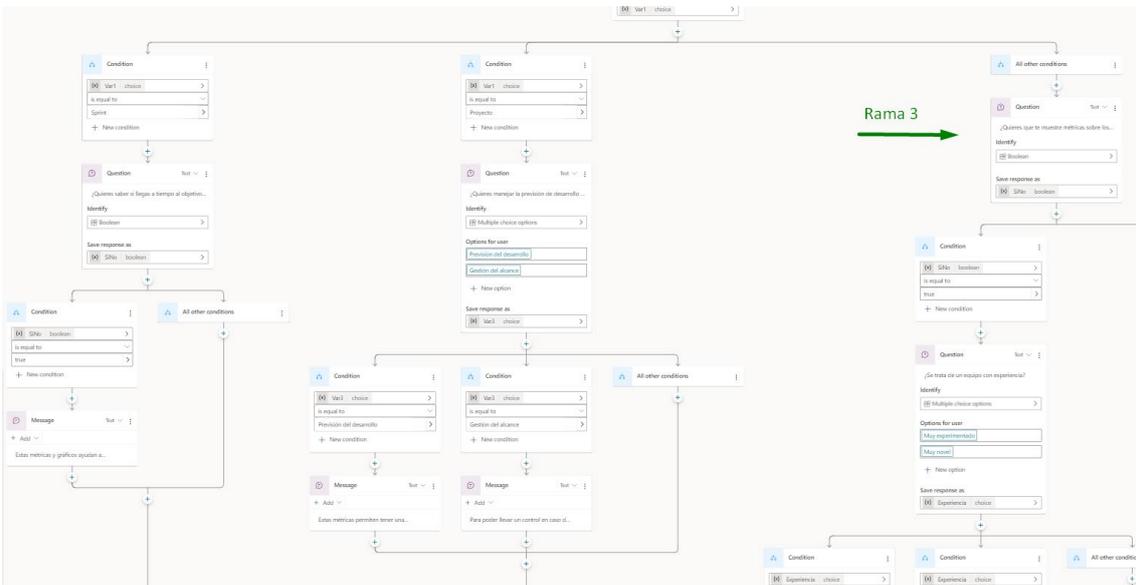




Nota. La rama 2 finaliza en el cierre.

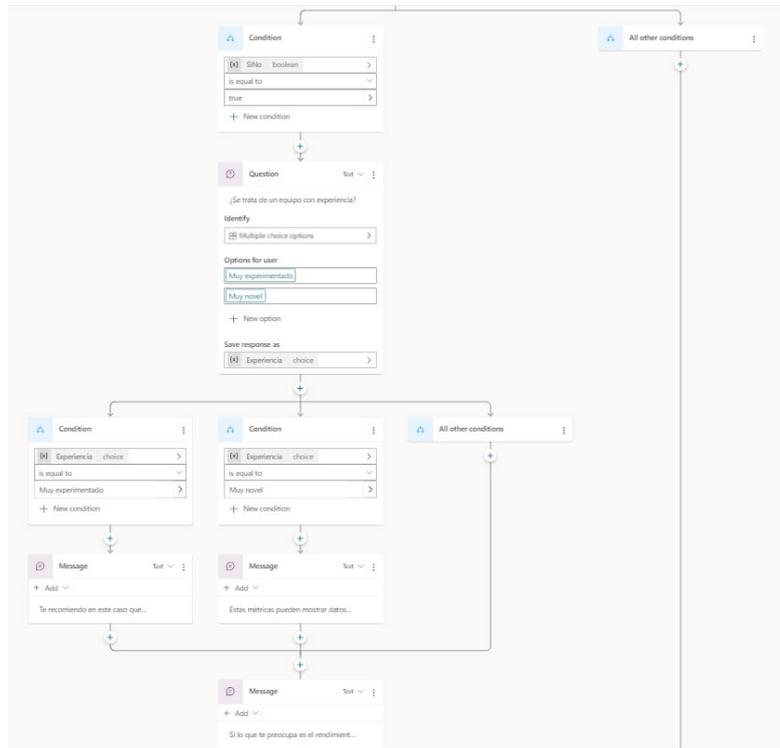
Figura 77

Topic Elección métrica – Rama 1 , chatbot EncuentraTuMétrica



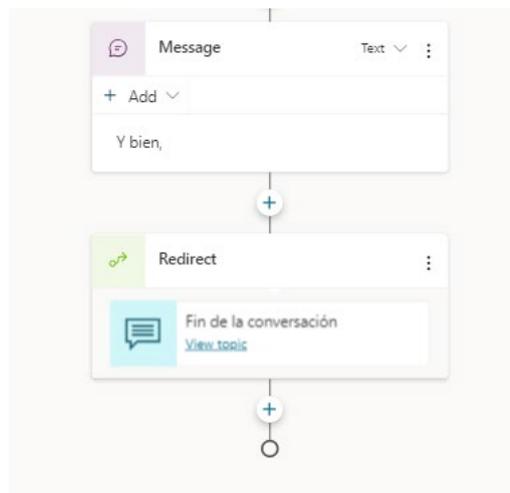
**Figura 78**

*Topic Elección métrica – Rama 3, chatbot EncuentraTuMétrica*



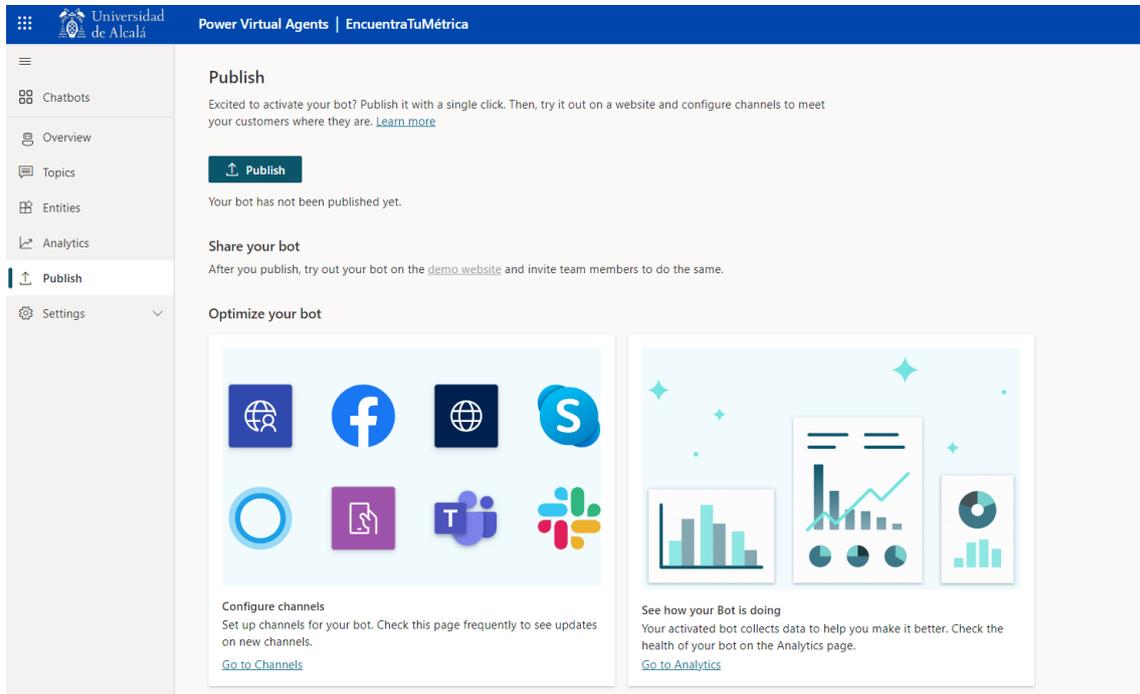
**Figura 79**

*Cierre Topic Elección métrica, chatbot EncuentraTuMétrica*



**Figura 80**

*Publicación chatbot EncuentraTuMétrica*



Tras la publicación se puede crear una web de demostración donde se podrá ejecutar el chatbot.

**Figura 81**

*Conversación con chatbot EncuentraTuMétrica en web demo*

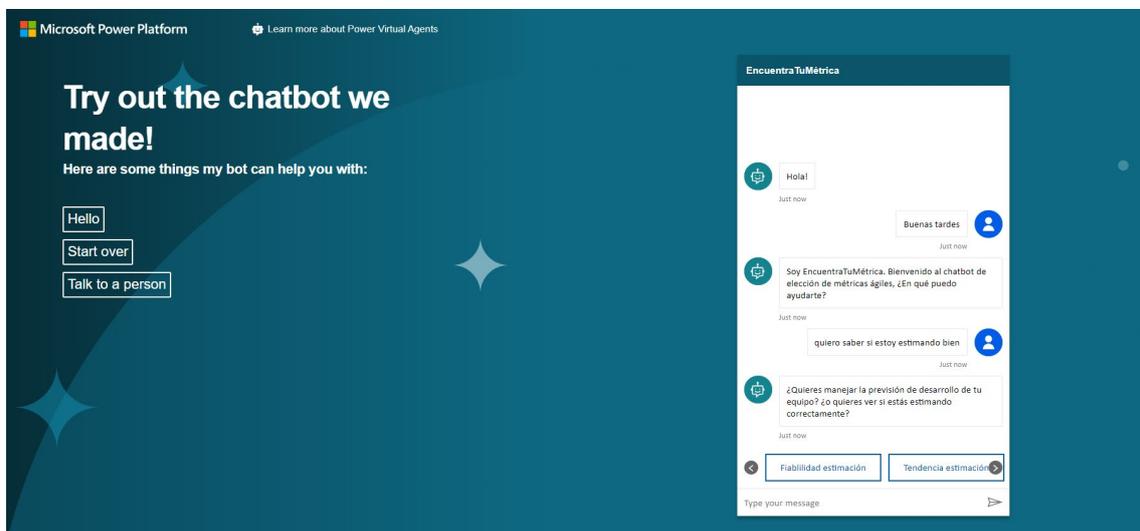


Figura 82

Conversación con chatbot EncuentraTuMétrica en web demo

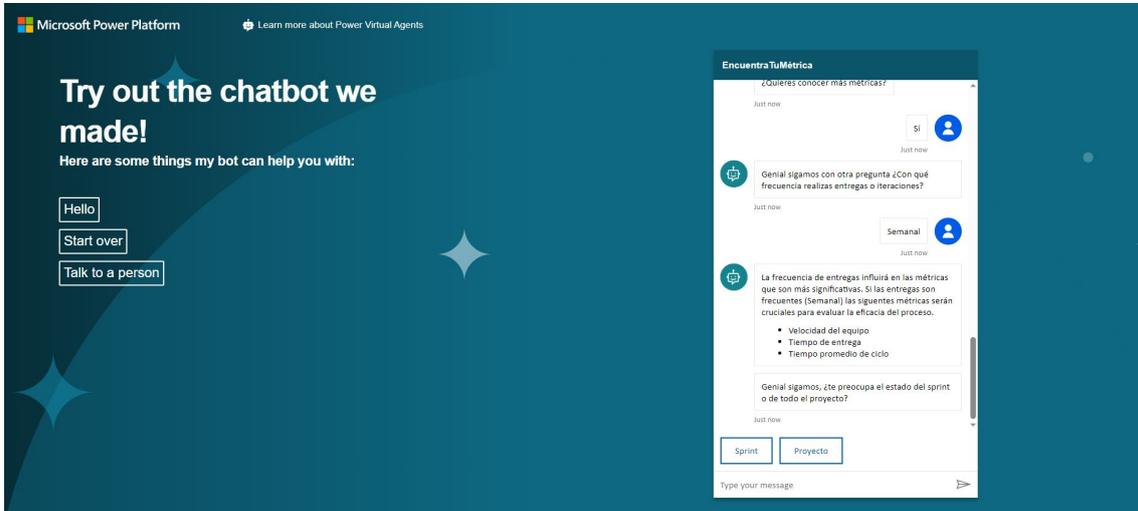


Figura 83

Conversación con chatbot EncuentraTuMétrica en web demo

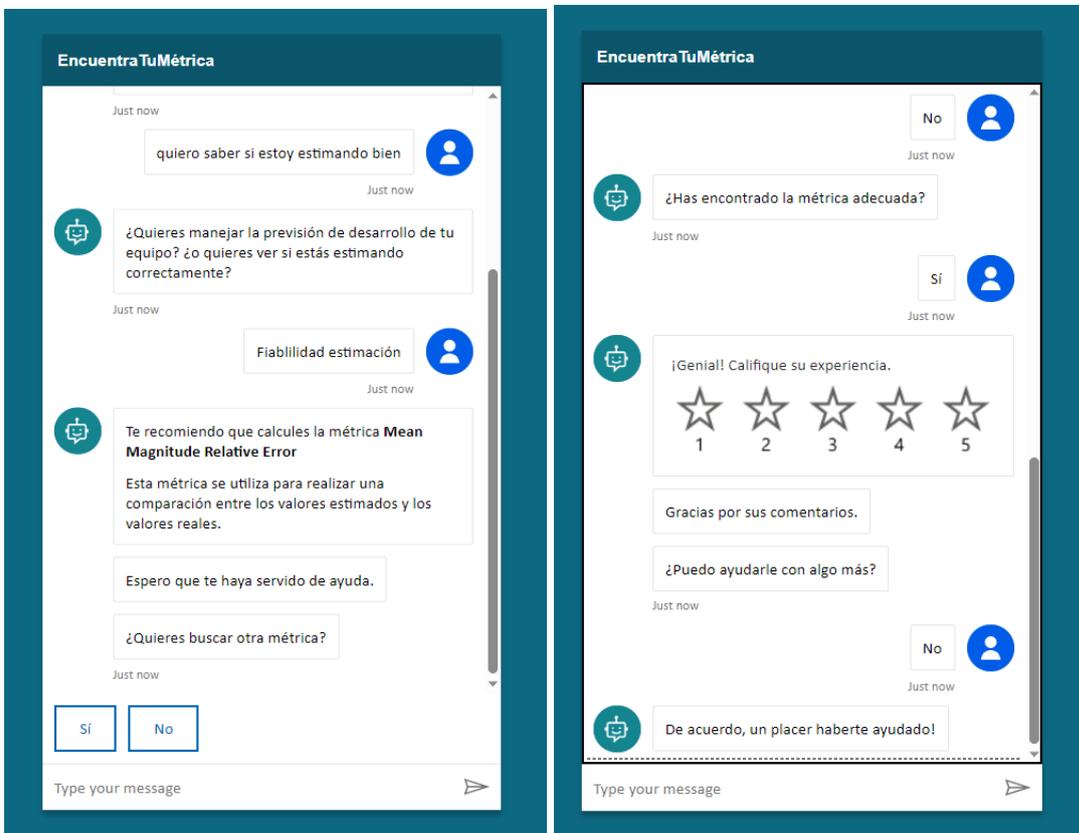


Figura 84

Conversación con chatbot EncuentraTuMétrica en web demo

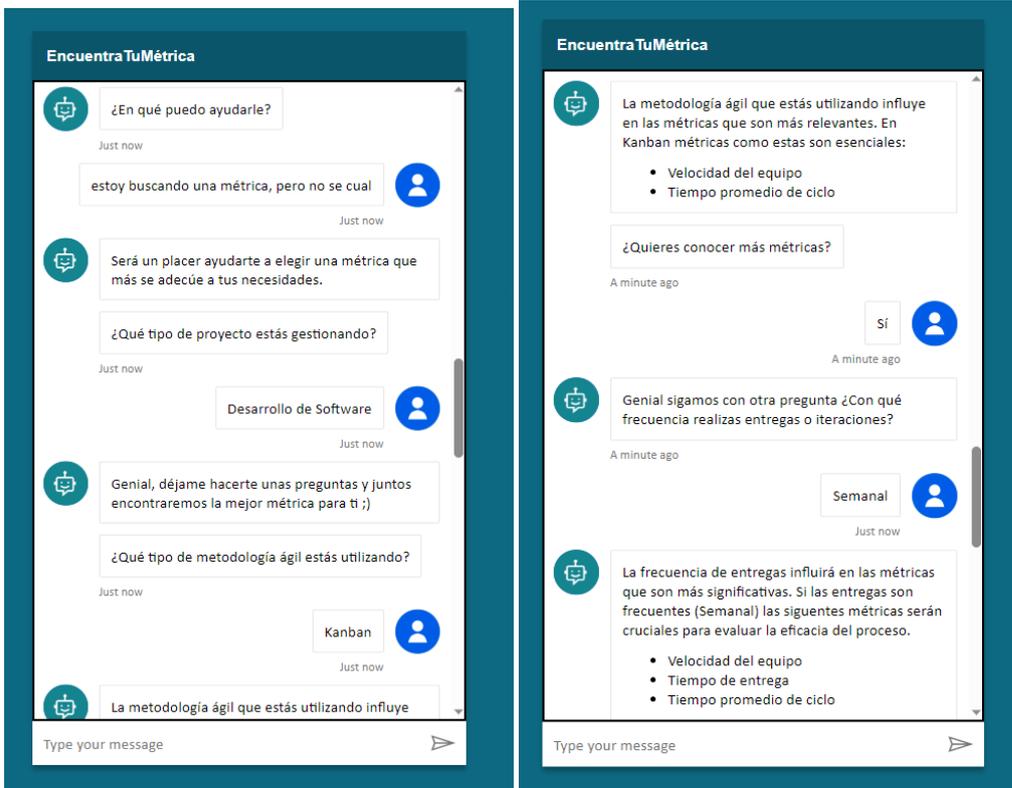
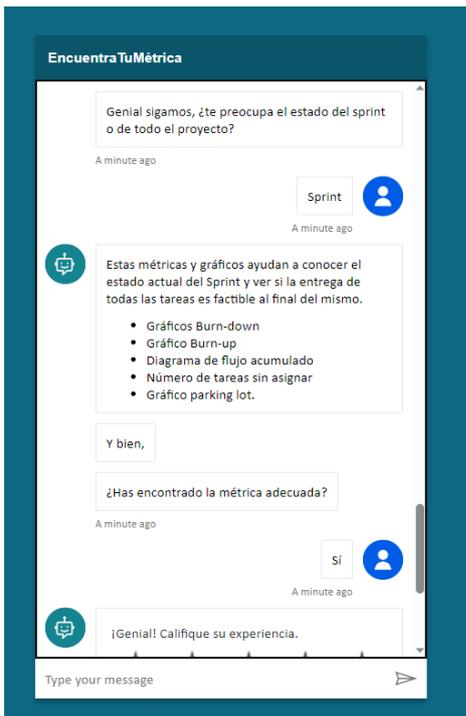


Figura 85

Conversación con chatbot EncuentraTuMétrica en web demo



## 9. Resultados

Como resultado puedo apreciar que existen en la actualidad una variedad de métricas que puede aportar mucha información a los equipos y a los jefes de proyecto sobre los avances de un proyecto, sus dificultades o su capacidad de desarrollo.

En algunos casos los conceptos se han tratado de la misma manera métricas numéricas, gráficas, KPIs, etc. Para este estudio se han considerado todos ellos métricas dado que el objetivo es evaluar todos los posibles aportes de valor de datos obtenidos del uso de estas metodologías. En algún caso una gráfica puede aportar tanto una imagen visual de un progreso o cuello de botella, como una o varias conclusiones numéricas.

Se han encontrado diversas clasificaciones en cuanto a las nuevas métricas estudiadas, por lo que la propuesta de división es únicamente un compendio de varias de ellas, intentando englobar las más actuales y utilizadas en temáticas que tienen en común.

También es importante destacar de de igual manera que las métricas usadas en la gestión tradicional de proyectos pueden adaptarse a la gestión mediante metodologías ágiles, también es posible adaptar gran parte de las métricas generadas para las metodologías ágiles hacia los proyectos tradicionales. Esto es debido a que estas nuevas metodologías tienen como parte de sus principios la flexibilidad, y hace posible que los desarrollos no sean gestionados de una forma tan rígida como se hacía en el pasado.

## 10. Conclusiones

La realización de este trabajo final de máster me lleva a concluir que como parte del cambio de paradigma que supone pasar de realizar la gestión de proyectos de un proceso más tradicional a realizarse mediante metodologías ágiles, está el hecho de poner el foco en el trabajo que se realiza y no solo en los números. Esto supone que además de tomar las medidas de productividad como objetivos a alcanzar, también se utilizan como una forma de aprender para los miembros del equipo, realizar reuniones que lleven a establecer conversaciones y elevar la calidad de las discusiones.

Esto supone que cada vez que un equipo finaliza una tarea o una historia, sea bueno mirarla objetivamente y que se profundice en los detalles para entender lo que realmente ha sucedido en la misma. Estas reuniones y discusiones, junto que las métricas que se proponen en esta metodología, son las que ayudan a los equipos a mejorar su flujo de trabajo y su productividad con el tiempo.

Definitivamente, las métricas desempeñan un papel crucial en las metodologías ágiles, permitiendo a los equipos y las organizaciones evaluar su rendimiento y tomar decisiones informadas. La selección y aplicación adecuada de las métricas en metodologías ágiles es fundamental para lograr el éxito en los proyectos y mejorar la calidad del producto entregado.

Los estudios relativos a las métricas y metodologías ágiles han ganado gran interés en los últimos cinco años. En el contexto del desarrollo de software se han realizado numerosas investigaciones, tanto basadas en datos como en valoraciones subjetivas.

Toda esta variedad de métricas se enfocan en general en reforzar el uso de metodologías ágiles, promoviendo la velocidad en la que se entrega producto al cliente, pero también en revisar de manera constante la calidad que se provee.

En el caso del análisis de herramientas de gestión de proyectos mediante metodologías ágiles, se han encontrado una amplia cantidad de herramientas que permiten distintos tipos de métricas, pero con la distinción principal de el tipo y número de métricas según el enfoque de la herramienta: en el caso de herramientas más centradas en la gestión del marketing o los clientes se encuentran más métricas asociadas a estos valores; en la mayoría de los casos las herramientas están centradas en la gestión del proyecto y del equipo desarrollador, así como de los errores reportados a tratar, por lo que las métricas están enfocadas a estos puntos.

En un entorno en el que cada dato aporta valor, estas herramientas permiten conectar con otras más poderosas en el análisis de datos, que permitan realizar todo tipo de métricas a demanda.

El desarrollo realizado para la elección de la métrica más adecuada al usuario ha supuesto un análisis en detalle de cuales son las necesidades que un gestor de proyectos puede tener a la hora de evaluar el estado o los problemas que pueda tener su producto.

Este proceso ha destacado como con una herramienta que permita una conversación con cierta fluidez se puede conseguir de manera dinámica ofrecer las métricas adecuadas.

También, al ser realizada con una aplicación “Low code” se ha enfatizado el auge de las aplicaciones “self-service” que permiten con pocos conocimientos desarrollar a cualquier usuario flujos que automaticen y faciliten tareas recurrentes o tediosas.

### 10.1. Limitaciones del estudio

Debido al gran volumen de publicaciones existentes respecto al análisis de metodologías ágiles y las métricas que son de aplicación a éstas, no es posible asegurar que los estudios más relevantes hayan sido seleccionados para el trabajo. Sin embargo se han analizado numerosos artículos seleccionados en IEEE Xplore, Science Direct, SCOPUS, ACM Digital Library y WOS de amplio reconocimiento por la comunidad científica.

Para poder realizar un estudio enfocado a la actualidad, los artículos seleccionados han sido todos posteriores a 2017, por lo que es posible que alguna métrica o artículo de relevancia haya sido descartado por este motivo pudiendo contener información de relevancia para el estudio.

En el caso del desarrollo del proyecto en Jira la limitación es el haber desarrollado personalmente la creación de las tareas, y sus tiempos, asignaciones, etc. por lo que esta información podría estar sesgada y no ofrecer una imagen que podría ser más real si se hubiera aplicado a un desarrollo real.

## 10.2. Trabajos futuros

Un próximo paso podría ser la inclusión de nuevas líneas de preguntas al árbol de decisión del chatbot, que incluyan nuevas métricas. También podría realizarse el árbol de manera personalizada para un proyecto o empresa concreto conociendo cuales son las preguntas que tienen más importancia para poder arrojar resultados adaptados a casos concretos de desarrollo.

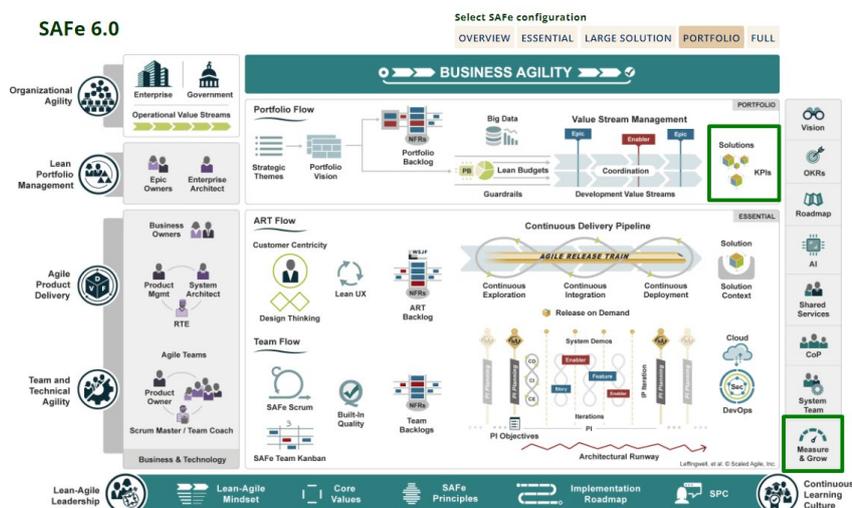
También podría realizarse a futuro una evaluación en Jira más completa aplicada a casos reales. Esto permitiría tener mayores conclusiones sobre las métricas aplicadas y las conclusiones y consecuencias que tiene el conocimiento de éstas.

Por otra parte, el mundo de las metodologías ágiles se encuentra en constante evolución. Actualmente se está implementando en un gran número de empresas la metodología SAFe, en la que, como se puede apreciar en la imagen inferior, los KPIs y las métricas forman parte de los pasos finales y en ellos recae la importancia de la evaluación de los datos obtenidos para poder realizar una mejora continua en sucesivas ejecuciones del proceso.

Una línea futura de investigación pueden ser cuales son las nuevas métricas que aparecen al implementar SAFe como metodología ágil en un caso práctico.

**Figura 86**

*Estructura organizativa de SAFe 6.0*



*Nota.* Leffingwell, D. (2023). Say Hello to SAFe 6.0! [Say Hello to SAFe 6.0! - Scaled Agile Framework](#)

## 11. Referencias bibliográficas

- *16 State of Agile Informe | Informes de analistas | Digital.ai.* (s. f.-b).  
Digital.ai. <https://digital.ai/es/resource-center/analyst-reports/state-of-agile-report/>
- *2021 State of DevOps Report | Google Cloud.* (s. f.). Google Cloud.  
<https://cloud.google.com/devops/state-of-devops/>
- *28 Important project Management KPIs (& How to Track Them) | ClearPoint Strategy.*  
(s. f.). <https://www.clearpointstrategy.com/blog/important-project-management-kpis>
- Albaladejo, X. (2020, 22 marzo). *Métricas ágiles y cuadro de mandos integral para Scrum.* Proyectos Ágiles. <https://proyectosagiles.org/2008/12/07/metricas-agiles-cuadro-mandos-balanceado-scrum/>
- Asana. (s. f.). *Cómo explorar Asana | Guía del producto • Guía del producto de Asana.* Guía del producto de Asana. <https://asana.com/es/guide/help/fundamentals/navigating-asana>
- Aston, B. (2023). 10 Herramientas para la gestión ágil de proyectos. *The Digital Project Manager.* <https://thedigitalprojectmanager.com/es/tools/herramientas-agiles-proyectos/>
- *Atlassian Marketplace.* (s. f.). <https://marketplace.atlassian.com/categories/analytics-reports>
- *Agile Reporting - Agile Dashboard | Axosoft.* (s. f.). Axosoft.com.  
<https://www.axosoft.com/tour/reporting>
- Ching Beh, H., Yah Jusoh, Y., Abdullah, R., Hassan, S. (2022). *Dimensions in Measuring Performance of Agile Software Development Projects: A Literature Review* [Comunicación en congreso]. 2022 Applied Informatics International Conference (AiIC), Serdang, Malaysia. IEEE Conference Publication | IEEE Xplore.  
<https://ieeexplore.ieee.org/document/9914025>
- Chcomley. (s. f.). *Documentación de Azure Boards.* Microsoft Learn. <https://learn.microsoft.com/es-es/azure/devops/boards/?view=azure-devops>
- Coelho, E., y Basu, A. (2012). Effort estimation in agile software development using story points. *International Journal of Applied Information Systems (IJ AIS).*
- Cohn, M. (2005). *Agile estimating and planning.* Pearson Education.
- Cohn, M. (2009). *Succeeding with Agile: Software development using Scrum.* [http://openlibrary.org/books/OL24044791M/Succeeding\\_with\\_agile](http://openlibrary.org/books/OL24044791M/Succeeding_with_agile)

- Choraś et al. (2020). *Measuring and improving agile processes in a Small-Size software development company*. Journals & Magazine | IEEE Xplore.  
<https://ieeexplore.ieee.org/document/9076670>
- Davis, C. W. H. (2015). *Agile Metrics in Action: Measuring and enhancing the performance of agile teams*.
- *Deep dive into azure boards*.  
(2019). <https://microsoftonlineguide.blogspot.com/2018/09/deep-dive-into-azure-boards.html>
- *Development Analytics - SquaredUp*. (s. f.).  
<https://squaredup.com/solutions/development-analytics-devops-analytics/>
- Gordon, J. (2021). *AzureFunBytes - A brief intro to azure boards*. *DEV Community*.  
<https://dev.to/azure/azurefunbytes-a-brief-intro-to-azure-boards-m84>
- Hazzan, O., & Dubinsky, Y. (2014). *The Agile Manifesto*. En *SpringerBriefs in computer science* (pp. 9-14). [https://doi.org/10.1007/978-3-319-10157-6\\_3](https://doi.org/10.1007/978-3-319-10157-6_3)
- Hilska, O. (2022b, marzo 9). *Practical Guide to DORA Metrics*. *Swarmia*. <https://www.swarmia.com/blog/dora-metrics/>
- *Indicators and reporting – IceScrum*. (2021, 14 diciembre).  
IceScrum. <https://www.icescrum.com/documentation/indicators-and-reporting/>
- IFPUG. (2023, 18 mayo). *We prefer Facts to Stories (Managing Agile activities using standardised measures)*. <https://www.ifpug.org/wp-content/uploads/2018/05/Managing-Agile-activities-using-standardised-measures.pdf>
- Joiner, B. (2021, 20 octubre). *Un resumen sobre las funciones controladas por datos de Trello*. <https://blog.trello.com/es/funciones-controladas-por-datos-de-trello#:~:text=Usa%20la%20Vista%20de%20panel,la%20mayor%20cantidad%20de%20tarjetas%3F>
- Junquera, A. (2019). *8 Herramientas ágiles para la gestión de proyectos*. *Blog Grupo Digital*. <https://www.grupodigital.eu/blog/8-herramientas-agiles-para-la-gestion-de-proyectos/>
- Fernández-Vivancos, G. M. (2019). *Diseño de indicadores para la gestión de proyectos*. Repositorio Institucional – Universidad de Valladolid.  
<https://doi.org/10.35376/10324/22086>

- *Las 10 herramientas Agile más valiosas para el 2021 | Conexión ESAN.*  
(s. f.). <https://www.esan.edu.pe/conexion-esan/las-10-herramientas-agile-mas-valiosas-para-el-2021>
- Leffingwell, D. (2023, 15 marzo). *Say Hello to SAFE 6.0! - Scaled Agile Framework.* Scaled Agile Framework. <https://scaledagileframework.com/blog/say-hello-to-safe-6-0/>
- Malathi, S. (2012, 6 noviembre). *Estimation of effort in software cost analysis for heterogenous dataset using fuzzy analogy.* arXiv.org. <https://arxiv.org/abs/1211.1136>
- Martín, E. (2023, 4 julio). *¿Qué diferencias hay entre Jira Work Management y Trello?* Deiser Blog. <https://blog.deiser.com/es/diferencias-entre-jira-work-management-vs-trello>
- monday.com. (s. f.). *Monday Dev Product Development Management Features | monday.com.* <https://monday.com/dev/features>
- Pmadmin. (2023, 15 febrero). *What is Scrum? | The Agile Journey with PM-Partners.* PM Partners. <https://www.pm-partners.com.au/the-agile-journey-a-scrum-overview/>
- Project Management Institute Inc. (2021). *Guía de los Fundamentos para la Dirección de Proyectos, (Guía del PMBOK®) – Séptima Edición.*
- R. Méndez, E. (2018). *Estimación de esfuerzo en proyectos de desarrollo de Software con metodologías ágiles.* [TFM]. Repositorio Institucional – Universidad Politécnica de Valencia. [https://riunet.upv.es/bitstream/handle/10251/109927/PSC9691917\\_TFM\\_153255262\\_0188720975771735607734.pdf?sequence=1&isAllowed=y](https://riunet.upv.es/bitstream/handle/10251/109927/PSC9691917_TFM_153255262_0188720975771735607734.pdf?sequence=1&isAllowed=y)
- Radigan, D. D. (s. f.). *Gestión ágil de proyectos – Métricas. Cinco métricas ágiles que no odiarás | Atlassian.* Atlassian. <https://www.atlassian.com/es/agile/project-management/metrics>
- Screenful - #1 Productivity Analytics platform. (s. f.). <https://screenful.com/>
- Sention. (2022). *5 Herramientas para la gestión ágil de proyectos de software en 2022.* Sention. <https://sention.io/blog/mejores-herramientas-gestion-agil/>
- Sutherland, J. (2017, 31 mayo). *Story points: Why are they better than hours? - Scrum Inc.* Scrum Inc. <https://www.scruminc.com/story-points-why-are-they-better-than/>
- Techagilist. (2023). *Velocity Chart in JIRA – Stabilize and improve velocity.* Tech Agilist. <https://www.techagilist.com/agile/jira/velocity-chart-in-jira-stabilize-and-improve-velocity/>

- Universidad Europea. (2023). Las mejores herramientas Agile para la gestión de proyectos. *Universidad Europea*. <https://universidadeuropea.com/blog/herramientas-agile/>
- Victor. (2023). Jira vs Trello vs Asana – ¿Cuál es el mejor gestor de tareas? - Walkiria Apps. *Víctor Alonso*. <https://walkiriaapps.com/tecnologia/jira-vs-trello-vs-asana/>
- Vige, W. (2022, 3 diciembre). Puntos de Historia: Guía para estimar las historias de usuarios en Agile [2022]. Asana. *Asana*. <https://asana.com/es/resources/story-points>
- Villán, V. R. (2022c, agosto 22). *13 Herramientas para la gestión de proyectos Agile*. Thinking for Innovation. <https://www.iebschool.com/blog/herramientas-gestion-agil-proyectos-agile-agile-scrum/>
- Włodarski et al. (2020, 17 septiembre). *Comparative case study of Plan-Driven and Agile approaches in student computing projects*. IEEE Conference Publication | IEEE Xplore. <https://ieeexplore.ieee.org/document/9238196>
- Yec. (2023, 4 abril). Mean Time to Recovery—A DORA metric explained. *Forbes*. <https://www.forbes.com/sites/theyec/2023/04/04/mean-time-to-recovery-a-dora-metric-explained/?sh=1f0c312c60b4>
- Yusov, K. (2020, 16 agosto). Our experience of using most common agile metrics. *Jelvix*. <https://jelvix.com/blog/9-fundamental-agile-metrics>

