

Integración de herramienta para el seguimiento ágil mediante el uso de Machine Learning

**Yadira Jazmín Pérez Castillo¹, Sandra Dinora Orantes Jiménez¹,
Patricio Orlando Letelier Torres², María Elena Acevedo Mosqueda³**

¹Centro de Investigación en Computación, Instituto Politécnico Nacional, Av. Juan de Dios Bátiz, esq. Miguel Othón de Mendizábal, Col. Nueva Industrial Vallejo, Alcaldía Gustavo A. Madero, C.P. 07738, CDMX, México.

²Departamento de Sistemas Informáticos y Computación, Escuela Técnica Superior de Ingeniería Informática, Universitat Politècnica de València Camino de Vera, s/n, C.P. 46022, Valencia, España.

³Instituto Politécnico Nacional, Escuela Superior de Ingeniería Mecánica y Eléctrica (ESIME-IPN), CDMX, 07738, México.

yperezc2022@cic.ipn.mx, dinora@cic.ipn.mx, letelier@dsic.upv.es,
eacevedo@ipn.mx

Resumen. *El enfoque ágil ha impulsado la necesidad de herramientas avanzadas para la gestión de proyectos, especialmente en entornos multiproyecto. Este trabajo presenta la integración de un modelo de Machine Learning (ML) en Worki, una plataforma de seguimiento ágil, para analizar datos históricos de Sprints y evaluar el rendimiento. El modelo clasifica y evalúa el progreso y ritmo de trabajo, mostrando los resultados directamente en la plataforma. Esto permite una visión basada en datos, facilitando la detección temprana de anomalías y mejorando la eficiencia en la gestión de proyectos.*

Abstract. *The agile approach has driven the need for advanced tools in project management, especially in multi-project environments. This work presents the integration of a Machine Learning (ML) model into Worki, an agile tracking platform, to analyze historical Sprint data and evaluate performance. The model classifies and assesses progress and work pace, displaying the results directly within the platform. This enables a data-driven perspective, facilitating early anomaly detection and improving project management efficiency.*

1. Introducción

En la actualidad, el enfoque ágil ha adquirido una relevancia significativa en el desarrollo de software y la gestión de proyectos, desde sus inicios, consolidándose como un elemento clave para adaptarse a las necesidades dinámicas del mercado y de los equipos de trabajo. El Manifiesto Ágil (Sutherland, 2001), estableció los principios fundamentales que guían la práctica de desarrollo ágil, enfocándose en la colaboración, la flexibilidad y la entrega continua de valor. Este marco ha evolucionado para abordar los desafíos contemporáneos en proyectos ágiles.

Según el informe *The 17th State of Agile Report* de Digital.ai (2024) (Digital.ai, 2024), el 71% de las organizaciones encuestadas utilizan prácticas ágiles en su ciclo de vida de desarrollo de software (SDLC). Sin embargo, solo el 11% de los encuestados se sienten "muy satisfechos" con los resultados obtenidos, lo que indica que, a pesar de su amplia adopción, persisten desafíos en la implementación efectiva de estas metodologías (Digital.ai, 2024).

En este contexto, la integración de técnicas de "Machine Learning" (ML) en entornos ágiles se presenta como una oportunidad innovadora para optimizar procesos, mejorar la precisión en la planificación y facilitar la toma de decisiones en entornos ágiles, contribuyendo así a una gestión de proyectos eficiente y adaptativa.

Según (Dam, Tran, Grundy, Ghose, & Kamei, 2019), la inteligencia artificial (IA) tiene el potencial de transformar significativamente la práctica de la gestión de proyectos, ayudando a acelerar la productividad y aumentar las tasas de éxito de los proyectos. En este trabajo también proponen un marco donde las tecnologías de IA pueden ofrecer soporte para gestionar proyectos ágiles, automatizando tareas repetitivas y de alto volumen, habilitando análisis de proyectos para estimaciones y predicciones de riesgos, proporcionando recomendaciones accionables e incluso tomando decisiones. Esta perspectiva resalta la relevancia y los avances en la incorporación de técnicas avanzadas que integran ML para optimizar la Gestión Ágil de Proyectos (GAP), permitiendo a los gestores no solo supervisar el avance, sino también tomar decisiones oportunas.

Por lo anterior, en este trabajo se describe la integración de un modelo de ML en la herramienta Worki, perteneciente al enfoque TUNE-UP Process (TUNE-UP Process, 2017), que fue desarrollado para facilitar la transformación ágil de equipos de trabajo. En un contexto ágil, el término "Sprint" se refiere a un bloque de tiempo definido, generalmente de dos a cuatro semanas, en el que un equipo de desarrollo se enfoca en completar un conjunto específico de tareas previamente planificadas. Según la Guía Scrum de Schwaber y Sutherland, los "Sprint" son la base del desarrollo incremental, proporcionando una estructura iterativa para alcanzar objetivos claros dentro de ciclos breves (Schwaber & Sutherland, 2020).

La integración de algoritmos de ML en Worki busca potenciar este enfoque iterativo al analizar datos históricos de "Sprint" para evaluar el progreso y la velocidad del trabajo; permitiendo detectar anomalías en el progreso y tomar acciones correctivas oportunamente. Este trabajo analiza cómo la integración de técnicas de ML, sobre la herramienta Worki, contribuye a optimizar el seguimiento de proyectos. Se examinan los resultados derivados de la evaluación de los "Sprints" y se destacan las ventajas de emplear enfoques adaptativos para potenciar la supervisión del avance del Sprint. Estas funcionalidades subrayan el impacto transformador del ML en la GAP, especialmente en contextos multiproyecto.

El presente trabajo se organiza de la siguiente manera: en la sección 2, se presenta un panorama general sobre el uso de ML como apoyo a la GAP. La sección 3 ofrece una descripción detallada de la herramienta de seguimiento utilizada, Worki, además se describe como fue la generación del modelo a integrar. En la sección 4, se aborda el desarrollo de la integración del ML dentro de la herramienta Worki. La sección 5 se centra en la visualización de los resultados obtenidos.

En la sección 6 presenta una breve discusión sobre los hallazgos. Finalmente, las secciones 7 y 8 concluyen con las principales aportaciones del estudio y se proponen direcciones futuras de investigación.

2. Contexto

Hoy día, la GAP, especialmente en entornos multiproyecto, requiere herramientas que no solo permitan supervisar el progreso del trabajo, para así tomar decisiones oportunas.

Las herramientas de seguimiento tradicionales suelen limitarse a ofrecer una visión estática del estado actual del proyecto, sin proporcionar una capacidad integral para discernir entre los diversos proyectos en curso. Por esta razón, la integración de tecnologías avanzadas como ML se ha vuelto clave para abordar estos desafíos como la supervisión del progreso del trabajo, debido a que estas técnicas permiten analizar grandes volúmenes de datos y ofrecer predicciones y análisis más profundos.

Según (Mamatha & Suma, 2021) la aplicación de técnicas de ML en la gestión de proyectos de software ha demostrado ser efectiva para mejorar la toma de decisiones, mejorar la precisión en las estimaciones y facilitar una gestión más adaptativa. Estos avances tecnológicos permiten una visión más precisa y dinámica de los proyectos, lo que ayuda a tomar decisiones más informadas y ajustadas a las realidades cambiantes del entorno ágil (Mahdi, y otros, 2021).

De manera similar, (Mamatha & Suma, 2021) destacan el papel de ML en la mejora de la precisión de las estimaciones, la optimización de la planificación y la identificación temprana de riesgos, lo que a su vez facilita una gestión de proyectos más ágil y eficiente.

Además, en (Prasetyo, Peranginangin, Martinovic, Ichsan, & Wicaksono, 2025) destacan cómo la inteligencia artificial (IA) está ganando terreno en la gestión de proyectos de innovación abierta, subrayando la importancia de integrar tecnologías avanzadas para una gestión más eficiente y flexible.

Varios estudios han explorado la integración de ML en el contexto ágil, aunque cada uno se ha centrado en diferentes áreas de aplicación. Por ejemplo, en (Baharom, Rahman, Sabudin, & Nor, 2023) aplicaron ML en el desarrollo de herramientas de apoyo a la toma de decisiones, específicamente en un planificador inteligente, para apoyar en la planificación en proyectos ágiles, mejorando la asignación de recursos y la programación de tareas. En el área de estimación de esfuerzo y costos, en (Rodríguez Sánchez, Vázquez Santacruz, & Cervantes Maceda, 2023) utilizaron técnicas de árboles de decisión y punto de historia de usuario para predecir el esfuerzo y los costos asociados a proyectos ágiles de desarrollo de software. De manera similar, Meiliana et al. (2023) implementaron un enfoque basado en ML para estimar el esfuerzo en el desarrollo ágil, enfocándose en los elementos del backlog del producto. También, en (Tiwari, Phonsa, & Malik, 2024) la estimación en proyectos Scrum utilizando diversas técnicas de ML para mejorar la precisión de las predicciones en la planificación del esfuerzo y el tiempo.

Por otro lado, en (Gaona-Cuevas, Guerrero, & Vera-Rivera, 2024) aplicaron técnicas de ML para desarrollar un modelo de clasificación inteligente del Product Backlog, lo que permite una gestión más eficiente de las historias de usuario y una mejor priorización de tareas.

En el área de la determinación del tamaño ideal del equipo en prácticas ágiles, en (Olivares, Noel, Guzmán, Miranda, & Munoz, 2024) implementaron métodos basados en ML para optimizar la asignación de recursos y mejorar la productividad del equipo dentro del marco ágil.

Además, en (Shameem, Nadeem, & Zamani, 2023) propusieron un modelo probabilístico basado en algoritmos genéticos para predecir el éxito de los proyectos ágiles en el contexto del desarrollo global de software, mejorando la capacidad de planificación y anticipación de riesgos.

Estos trabajos destacan cómo el uso de ML se ha extendido a diversas áreas clave dentro de la GAP, como la estimación de esfuerzo, costos, riesgos y la optimización de equipos, mostrando su versatilidad y su potencial para transformar la toma de decisiones.

En el ámbito de la estimación de esfuerzo y costos, técnicas como los árboles de decisión y modelos probabilísticos se ha demostrado su capacidad para predecir con mayor precisión el esfuerzo requerido y los costos asociados, permitiendo una planificación más realista y ajustada a las necesidades del proyecto. En términos de gestión de riesgos, los algoritmos de ML permiten identificar y anticipar posibles problemas antes de que ocurran, mejorando la capacidad de respuesta ante cambios inesperados o problemas durante el desarrollo del proyecto. Además, en el área de la gestión de equipos, métodos de ML, como el análisis de datos históricos y la determinación del tamaño ideal del equipo han permitido mejorar la asignación de recursos, optimizando la productividad y asegurando que el equipo esté equilibrado para abordar las demandas del proyecto de manera más eficiente. Estos avances muestran cómo ML no solo apoya las decisiones, sino que también facilita una planificación más adaptativa, mejora las predicciones y permite un enfoque proactivo en, asegurando una mayor alineación con los objetivos estratégicos del proyecto.

Sin embargo, a pesar de los avances logrados en la integración de ML con el enfoque ágil, no se encontró trabajo en la línea de investigación enfocada al progreso del trabajo. Algunas herramientas de seguimiento como Jira (Atlassian, 2024) y Worki (Tuneupprocess, 2017), recopilan una amplia gama de datos durante el ciclo de vida de un proyecto, incluyendo tiempos de respuesta, asignación de tareas y rendimiento del equipo.

No obstante, la explotación de estos datos mediante técnicas de ML para proporcionar análisis predictivos y recomendaciones personalizadas aún está en sus etapas iniciales. Esta falta de integración, lejos de limitar la capacidad para tomar decisiones, representa una oportunidad para mejorar la calidad de las decisiones informadas y oportunas basadas en datos analíticos, especialmente en entornos multiproyecto.

En este contexto, el presente trabajo tiene como objetivo explorar cómo la integración de modelos de ML en la herramienta Worki puede mejorar el seguimiento de los Sprints, en los proyectos. Mediante el análisis de los datos proporcionados por Worki, este trabajo muestra cómo la integración de ML puede contribuir al seguimiento del progreso del trabajo.

3. Desarrollo previo

Worki es una herramienta desarrollada en Valencia, España, se nutre de tres vertientes: la docencia, la investigación y la aplicación industrial. A través de su metodología "TUNE-UP Process" (TUNE-UP Process, 2017), Worki integra prácticas de los métodos ágiles más

populares, como Scrum, Kanban, y Lean Development, permitiendo la gestión ágil de proyectos mediante workflows flexibles, configurables y un enfoque centrado en la gestión de requisitos a través de pruebas de aceptación. Además, facilita el seguimiento del tiempo, la gestión del backlog y la comunicación integrada en un entorno colaborativo.

Esta herramienta es un ejemplo de cómo se puede aplicar la metodología ágil en la práctica, contribuyendo a una gestión más eficiente y adaptativa de proyectos (Agilismoatwork, 2012).

Se eligió Worki como herramienta para el desarrollo de este estudio debido a su enfoque integral y su capacidad para gestionar proyectos ágiles y eficiente y particularmente por ofrecernos datos reales de 200 proyectos, considerando aproximadamente 551 Sprints para análisis. Estos datos fueron extraídos de Worki y empleados para generar el conjunto de datos destinado al entrenamiento de algoritmos de ML. Estos datos corresponden a equipos de trabajo de asignaturas de ingeniería de software, de la escuela de informática de la Universidad Politécnica de Valencia, España.

Dentro de Worki, específicamente en la sección de “dashboard”, se encuentran diversos gráficos diseñados para facilitar el seguimiento de los “Sprint”. Entre ellos, destacan la gráfica de Burndown y la gráfica de trabajo terminado y no terminado, tal como se ilustra en la Figura 1.

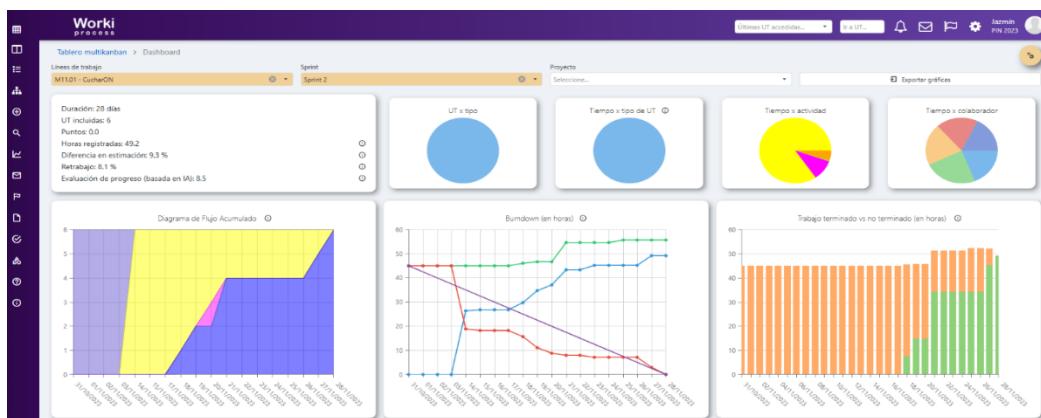


Figura 1 Dashboard de seguimiento de Sprints de Worki (Tuneupprocess, 2017)

En un trabajo previo realizado en (Pérez Castillo, Orantes Jiménez, & Letelier, 2024), se proporcionó un análisis detallado sobre el uso de estas gráficas y su importancia para evaluar el progreso y la velocidad de los Sprints. Además, se describió el proceso de extracción de datos llevado a cabo para la construcción del conjunto de datos y las consideraciones, para la generación de los rasgos y etiquetas, así como los experimentos realizados con algoritmos de ML seleccionados.

Dentro del trabajo previo realizado, se implementaron diversos experimentos utilizando algoritmos de ML. Entre los algoritmos probados en la experimentación, SVM (Support Vector Machine) destacó como el más apropiado debido a su capacidad para manejar datos complejos y ofrecer predicciones precisas. Los resultados experimentales demostraron que SVM proporciona un equilibrio óptimo entre precisión y eficiencia, lo que lo convierte en una elección sólida para evaluar el progreso y ritmo de “Sprint”.

La Tabla 1 resume las métricas clave obtenidas en los experimentos realizados con diferentes algoritmos de ML, evaluando su desempeño en términos de *Recall*, *F1-Score*, *Accuracy* y *Precision*. El algoritmo SVM sobresale en varias métricas, alcanzando un *Recall* de 0.88, un *Accuracy* de 0.88 y un *Precision* de 0.88, posicionándose como el más adecuado para el análisis de los Sprints.

Otros algoritmos como *KNN*, *MLP*, *RF* y *LR* también mostraron desempeños competitivos, con métricas similares, aunque ligeramente inferiores. En contraste, NB y AB presentaron resultados menos robustos, especialmente en *F1-Score* y *Recall*.

El proceso detallado de preprocesamiento de datos y experimentación con estos algoritmos se documenta en el trabajo previo. Este trabajo fundamenta la elección de SVM como el modelo óptimo entre precisión y eficiencia.

Tabla 1 Resultados experimentación

	Recall	F1	Accuracy	Precision
KNN	0.86	0.86	0.86	0.87
SVM	0.88	0.87	0.88	0.88
MLP	0.86	0.86	0.86	0.87
RF	0.87	0.86	0.87	0.87
NB	0.69	0.66	0.69	0.78
LR	0.87	0.87	0.87	0.88
AB	0.77	0.75	0.77	0.75

Por otro lado, las categorías de desempeño del “Sprint” también fueron definidas como: Malo, Regular, Bueno, Excelente. Estas categorías fueron determinadas en colaboración con un experto en enfoque ágil, quien, valido los criterios utilizados para su clasificación, además de que, se basan en métricas clave como el porcentaje de tareas completadas, la consistencia en el ritmo de trabajo diario y la cantidad de días sin progreso. A través de este proceso, se definieron umbrales específicos que permiten evaluar objetivamente hasta qué punto un sprint puede considerarse exitoso o necesita mejoras, garantizando una retroalimentación precisa.

4. Proceso de Integración

Con el objetivo de facilitar la integración del modelo de ML previamente desarrollado en Worki se diseñó y desarrolló una API (Interfaz de Programación de Aplicaciones), que permite interactuar con el modelo de forma eficiente y automatizada.

Esta API fue desarrollada en Python, utilizando el framework Flask¹, que permite gestionar las solicitudes HTTP. Además de Flask, se integraron diversas bibliotecas que permiten la manipulación, validación y análisis de los datos. También se incluyó Marshmallow² para la validación y serialización de los datos entrantes y salientes.

¹ <https://pypi.org/project/Flask>

² <https://pypi.org/project/marshmallow>

Para el procesamiento y análisis de grandes volúmenes de datos, como los gráficos generados por Worki, se utilizaron Pandas³ y NumPy⁴, que permiten manipular tablas y realizar cálculos numéricos eficientes. Además, se empleó Joblib⁵ para cargar y ejecutar el modelo de ML.

El principal propósito de la API es realizar la comunicación entre el modelo de ML, tomando los datos provenientes de Worki para pasarlo al modelo y así obtener la evaluación el desempeño de un Sprint.

A través de esta comunicación, los datos de las gráficas de BurnDown y otros indicadores de progreso se analizan y clasifican en cuatro categorías: Malo, Regular, Bueno, y Excelente, basadas en el rendimiento general del “Sprint”. Cada categoría incluye una retroalimentación detallada y recomendaciones prácticas para mejorar el desempeño de los equipos en futuros Sprints.

Una vez que la API fue desarrollada y probada localmente, se procedió con su integración con la plataforma de Worki (Tuneupprocess, 2017). Esta integración permite que Worki y la API trabajen de forma conjunta, agilizando el flujo de trabajo entre ambas plataformas y el modelo de ML. El flujo de interacción completo entre Worki, la API y el modelo de ML se representa de la siguiente manera: la API recibe datos directamente desde Worki, procesa la información utilizando el modelo de aprendizaje automático (ML) y devuelve los resultados junto con las recomendaciones, completando así un ciclo de retroalimentación eficiente. Este proceso de integración asegura que los datos se procesen y se devuelvan a Worki de forma inmediata, proporcionando información actualizada sobre el desempeño de los Sprints.

La interacción entre Worki, la API y el modelo de ML sigue un flujo estructurado, que se desglosa en tres fases principales:

1. **Consulta:** En esta etapa, la API valida los datos de entrada y realiza una solicitud a Worki para obtener la información necesaria sobre el estado del sprint solicitado. La validación de los datos es esencial para garantizar que la información recibida esté correctamente estructurada y sea útil, para el análisis.
2. **Análisis y Cálculo:** Tras obtener los datos, la API procede a analizar la información mediante el modelo de ML. Durante esta fase, se procesan datos de gráficas como las de Burndown y se calculan métricas clave que permiten evaluar aspectos como el ritmo de trabajo, los días sin progreso, la velocidad de entrega y la efectividad del equipo durante el “Sprint”.
3. **Retroalimentación:** Finalmente, la API clasifica el desempeño del sprint en las categorías predefinidas y genera un detalle de retroalimentación. Además, con el fin de proporcionar orientación sobre el estatus del “Sprint”.

La integración del modelo de ML con la plataforma Worki requirió el diseño y desarrollo de una arquitectura robusta que actúa como intermediaria entre Worki y el modelo de ML.

³ <https://pypi.org/project/pandas/>

⁴ <https://numpy.org/>

⁵ <https://joblib.readthedocs.io/en/stable/>

Este proceso presentó algunos retos técnicos para garantizar que los resultados de la API se integrasen intuitivamente en el frontend de Worki, como ejemplo, validar una comunicación sin afectar tiempos de respuesta, tanto del modelo, como de frontend de Worki.

Gracias a esta integración, es posible realizar un análisis del desempeño de los Sprints en tiempo real. Los encargados del proyecto pueden conocer el estatus de cualquier “Sprint” de manera automática, lo que les permite tomar decisiones oportunas y realizar ajustes en tiempo real. Este análisis continuo contribuye a la mejora constante en entornos ágiles y a la toma de decisiones informadas basadas en datos. La implementación de esta API también abre la puerta a futuras implementaciones. Gracias a su arquitectura flexible, es posible incorporar nuevas funcionalidades, como la inclusión de métricas adicionales, la adaptación a otros tipos de gráficos. Este enfoque modular y escalable asegura que la API no solo sea útil en el contexto actual, sino que también pueda evolucionar con las necesidades futuras.

5. Resultados

Para validar la integración de la herramienta Worki con el modelo de ML desarrollado, se llevó a cabo una evaluación preliminar con un grupo de estudiantes de la asignatura *Proceso del Software* en la Escuela Técnica Superior de Ingeniería Informática de la Universitat Politècnica de València, durante el semestre B del año 2024. El grupo estuvo compuesto por 26 equipos de 4 estudiantes cada uno y la evaluación, fue aplicada específicamente en el “Sprint 3”.

Durante la evaluación preliminar realizada, se planteó una pregunta clave a los participantes con el objetivo de medir su percepción sobre la utilidad de la incorporación de la evaluación del “Sprint” con ML en la herramienta Worki. La pregunta fue:

“¿Considera que la herramienta facilita la evaluación del desempeño de los Sprints ágiles y proporciona retroalimentación útil para mejorar los futuros Sprints?”.

Los resultados obtenidos se encuentran resumidos en el gráfico de la Figura 2. La interpretación de estos resultados revela que una proporción considerable de los estudiantes (55%) estuvo de acuerdo o totalmente de acuerdo con que la herramienta facilita la evaluación del desempeño de los “Sprint”.

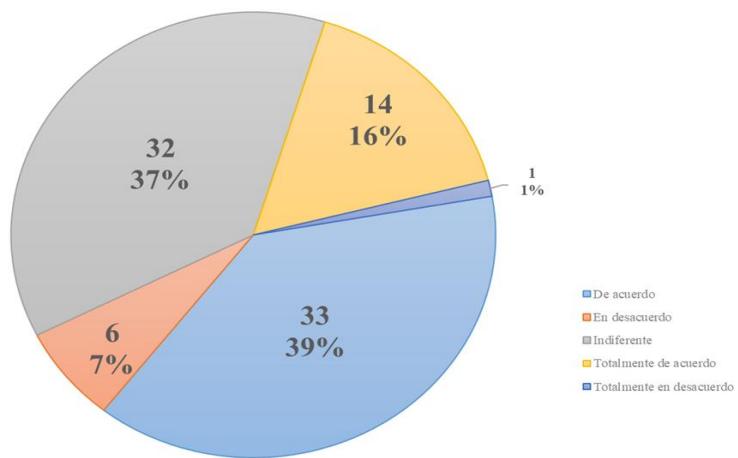


Figura 2 Respuestas al cuestionario

Lo anterior resalta la percepción positiva sobre su funcionalidad y potencial para apoyar el seguimiento ágil. Sin embargo, un 37% de los estudiantes se mostró indiferente, lo que podría estar relacionado con varios factores.

Por un lado, esta respuesta podría reflejar una falta de familiaridad con la herramienta o con los conceptos de evaluación de “Sprints” en general, lo que limitaría su capacidad para apreciar plenamente los beneficios ofrecidos.

Por otro lado, es posible que algunos estudiantes no hayan enfrentado escenarios lo suficientemente desafiantes durante los “Sprint” como para experimentar una necesidad tangible de apoyo en la evaluación.

También podría sugerir que la herramienta no logró conectar del todo con las expectativas o necesidades específicas de este grupo, ya sea por limitaciones en la interfaz, funcionalidad percibida o la relevancia de las métricas evaluadas.

En cuanto al 7% que expresó desacuerdo, este grupo podría haber encontrado dificultades específicas en el uso de la herramienta, como complejidad en la interpretación de los resultados, falta de personalización o posibles inconsistencias en los datos analizados.

Por otro lado, en la Figura 3, se puede observar un ejemplo de la interfaz de Worki, donde se muestran los detalles de la evaluación, incluyendo la categorización (calificación) del desempeño de los equipos, mostrando así el resultado de la integración del modelo de ML y la herramienta Worki.

Cabe mencionar que, para establecer la evaluación del “Sprint” en el frontend de Worki, primeramente, se consideraron cuatro niveles, previamente establecidos con el experto: Malo, Regular, Bueno, y Excelente. Estas métricas fueron transformadas a una escala numérica, donde cada categoría corresponde a un rango específico: Malo a 4.0, Regular a 7.0, Bueno a 8.5, y Excelente a 9.5.

Este cambio permitió simplificar la interpretación de los resultados, facilitando la clasificación del desempeño de los Sprint y proporcionando retroalimentación categorizada de manera más comprensible.

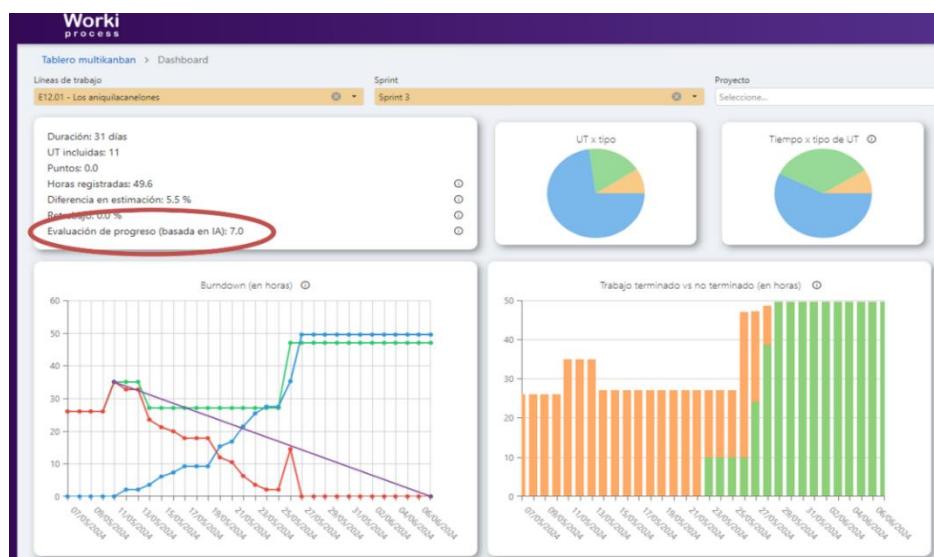


Figura 3 Resultado de integración

Finalmente, los resultados obtenidos de esta integración entre modelo y la herramienta confirman la capacidad del modelo ML para generar retroalimentación de forma precisa y eficiente, permitiendo a los equipos y el gestor identificar áreas de mejora. También se observó que, al recibir retroalimentación continua y categorizada, es posible tomar decisiones más informadas y ajustar su enfoque de trabajo para mejorar su rendimiento en los siguientes ciclos de “Sprint”.

La evaluación preliminar permitió validar que la integración entre el modelo ML y Worki es una herramienta funcional y eficaz. La fluidez en el procesamiento y entrega de los resultados en tiempo real demostró efectividad para proporcionar retroalimentación inmediata.

En conclusión, los resultados obtenidos evidencian el valor de la solución implementada para la evaluación de “Sprints” ágiles. La integración de ML con Worki no solo facilita el análisis de los datos, sino que también ofrece una retroalimentación útil y personalizada, permitiendo a los equipos fortalecer uno de los valores clave del enfoque ágil que es la mejora continua.

6. Discusión

Los resultados obtenidos en la evaluación preliminar de la integración de ML con la herramienta Worki reflejan una aceptación general de esta unión como una solución viable para la evaluación del desempeño de “Sprint”. Sin embargo, la diversidad en las respuestas también destaca la necesidad de analizar ciertos aspectos de la implementación y funcionalidad.

Un punto clave identificado es la percepción de utilidad expresada por el 55% de los estudiantes que estuvieron de acuerdo y totalmente de acuerdo, con que la herramienta facilita la evaluación y proporciona retroalimentación útil.

Este resultado sugiere que, el modelo de ML implementado tiene un impacto positivo en la identificación y análisis de un “Sprint”. Sin embargo, el porcentaje de estudiantes indiferentes (37%) y aquellos en desacuerdo (8%) subraya posibles limitaciones, como la complejidad percibida de los resultados o la falta de familiaridad con la herramienta.

Otro aspecto importante de la integración es que, aunque se logró una comunicación eficiente entre ambas partes, el entorno académico en el que se probó la herramienta podría haber influido en las percepciones de los estudiantes. En un contexto profesional, donde los usuarios tienen experiencia en entornos ágiles y herramientas de análisis avanzadas, la aceptación y utilidad podrían variar significativamente.

Además, el diseño de las categorías de desempeño (traducidas en calificaciones numéricas: 4.0, 7.0, 8.5 y 9.5) plantea un enfoque interesante pero también subjetivo para evaluar los “Sprint”. Si bien estas categorías simplifican la interpretación de los resultados, podrían requerir ajustes para reflejar de manera más precisa las complejidades del trabajo en equipo en proyectos ágiles reales.

Por otro lado, los hallazgos en los resultados invitan a una reflexión más profunda para identificar las áreas de mejora que permitan abordar las expectativas de los usuarios y potenciar la adopción de la herramienta Worki, ampliando su impacto en diferentes contextos educativos y profesionales.

Los resultados reflejan una percepción general positiva, aunque también sugiere áreas de oportunidad para mejorar la incorporación de ML en worki. El propósito de la evaluación fue medir de manera objetiva y práctica el rendimiento de los equipos en la gestión de “Sprints” ágiles.

Finalmente, la retroalimentación detallada y las recomendaciones prácticas generadas por el modelo ML ofrecen un valor añadido. No obstante, sería relevante explorar cómo los usuarios aplican esta retroalimentación y si realmente contribuye a mejorar la gestión de los “Sprint” en iteraciones posteriores. Este aspecto podría abordarse en futuras investigaciones que incluyan un análisis del impacto de la herramienta en escenarios más complejos.

En general, la evaluación preliminar resalta el potencial de la integración de Worki con ML, pero también subraya la importancia de iterar sobre su diseño e implementación para maximizar su utilidad en diversos contextos.

7. Conclusiones

Este trabajo ha explorado el potencial de integrar técnicas de ML en un entorno ágil, específicamente mediante el desarrollo de una API que se conecta con la plataforma de seguimiento de proyectos ágiles, Worki, con la finalidad de analizar el progreso de los Sprint. La implementación de esta integración ha permitido procesar datos relevantes, de gráficos como Burndown, obteniendo métricas clave, facilitando así la evaluación de un Sprint.

La evaluación preliminar realizada con estudiantes en un contexto académico ofreció información sobre la aceptación y percepción del modelo de ML en la herramienta. Los resultados mostraron que, aunque una mayoría se consideró útil la integración, persisten desafíos relacionados con la interpretación de los resultados. Esto indica la necesidad de ajustes tanto en la presentación de los resultados como en la interfaz de usuario para mejorar la experiencia de los equipos que adopten esta solución.

Una contribución importante de este trabajo es la capacidad de categorizar el desempeño de los Sprint en niveles y de generar recomendaciones prácticas para la mejora continua. Estas funcionalidades no solo simplifican la evaluación, sino que también proporcionan un enfoque basado en datos para apoyar la toma de decisiones, una característica fundamental en entornos ágiles.

Sin embargo, también se destaca la importancia de ampliar las pruebas de la herramienta en entornos reales, donde los equipos trabajan con mayor presión y complejidad. Esto permitirá evaluar con mayor precisión la efectividad del modelo ML y ajustar las métricas y recomendaciones a las necesidades específicas de proyectos del mundo profesional. Los resultados obtenidos destacan el potencial del modelo de ML para obtener la evaluación de un “Sprint” y proporcionar retroalimentación objetiva basada en datos. No obstante, se identifican diversas oportunidades para perfeccionar y expandir la solución, abordando escenarios más complejos y adaptándola a las dinámicas de equipos y proyectos más exigentes.

Este trabajo constituye un avance significativo en la integración de ML para optimizar el seguimiento ágil de proyectos en entornos multiproyecto, favoreciendo las retrospectivas

del sprint curso. Worki demuestra ser una herramienta eficaz para el seguimiento ágil de proyectos, ofreciendo funcionalidades clave para evaluar el progreso y ritmo de los Sprint.

Los resultados experimentales respaldan la elección de SVM como el modelo más adecuado para la integración con Worki, gracias a su equilibrio entre Precision y Accuracy en el análisis de datos. Esto refuerza el potencial de integrar ML en herramientas ágiles para optimizar la gestión y la toma de decisiones basadas en datos, estableciendo una base prometedora para futuras mejoras y aplicaciones.

8. Trabajos Futuros

A partir de los resultados obtenidos y las limitaciones identificadas, se proponen diversas líneas de trabajo para futuras investigaciones que permitan fortalecer y ampliar el uso de ML como apoyo al seguimiento ágil:

1. **Validación de la evaluación del “Sprint” en tiempos de desarrollo específicos:** Es necesario explorar la viabilidad de evaluar sprints dentro de marcos de tiempo específicos, considerando variaciones en las duraciones de los ciclos de trabajo. Esto permitirá determinar si las métricas y clasificaciones generadas son consistentes y relevantes en diferentes contextos de planificación ágil.
2. **Incorporación de gráficas de seguimiento adicionales:** Si bien la herramienta utiliza gráficos, como la gráfica de BurnDown, para evaluar el progreso, una línea de trabajo futura podría incluir el análisis de otras métricas y gráficas relevantes, como los diagramas de flujo acumulado. Esto complementaría la evaluación del “Sprint” al proporcionar una visión más integral sobre los patrones de trabajo y la productividad del equipo.
3. **Algoritmos avanzados para la evaluación del “Sprint”:** El uso de técnicas como análisis de clustering o redes neuronales podría ser explorado para agrupar, por ejemplo, calificaciones similares y detectar puntos de discordancia o anomalías en el desempeño de los equipos de trabajo. Estas técnicas permitirían identificar dinámicas de equipo menos evidentes y ofrecer retroalimentación más específica y detallada con la finalidad de complementar la evaluación del “Sprint”.
4. **Validación en entornos empresariales reales:** Ampliar las pruebas de la herramienta a entornos de desarrollo de software reales será clave para validar su aplicabilidad. La evaluación en empresas de diferentes sectores, con equipos diversos y en proyectos de alta complejidad, permitirá ajustar el modelo y los algoritmos para que respondan de manera óptima a escenarios reales.
5. **Estudio del impacto de la propuesta:** Evaluar cómo los gestores de trabajo perciben la utilidad y facilidad de uso de la herramienta, así como su impacto en la toma de decisiones, sería clave para garantizar su aceptación. Investigaciones adicionales podrían diseñarse para identificar barreras de adopción y posibles mejoras en la interfaz y la retroalimentación generada.

6. Compatibilidad con otras herramientas de gestión: Aunque el trabajo se centró en la integración con Worki, futuros desarrollos podrían extenderse a investigar la usabilidad de técnicas de ML con herramientas como Jira, Trello o Asana.

En conjunto, estas propuestas buscan fortalecer la herramienta desde un enfoque técnico, funcional y práctico, permitiendo una evaluación más completa de los “Sprint” y su adaptación a distintos entornos de trabajo.

Referencias

- Agilismoatwork. (2012). Herramientas para gestión ágil de proyectos de desarrollo de software. Obtenido de <https://agilismoatwork.blogspot.com/2012/01/herramientas-para-gestion-agil-de.html>
- Atlassian. (2024). Jira. Recuperado el 20 de octubre de 2024, de <https://www.atlassian.com/es/software/jira>
- Baharom, M., Rahman, M., Sabudin, A., & Nor, M. (2023). Decision support tools: Machine learning application in smart planner. Lecture Notes in Mechanical Engineering, 753-760. https://doi.org/10.1007/978-981-19-1939-8_58
- Pérez Castillo, Y. J., Orantes Jiménez, S. D., & Letelier Torres, P. O. (2024). Sprint management in agile approach: Progress and velocity evaluation applying machine learning. *Information*, 15(11), 726. <https://doi.org/10.3390/info15110726>
- Dam, H., Tran, T., Grundy, J., Ghose, A., & Kamei, Y. (2019). Towards effective AI-powered agile project management. IEEE/ACM 41st International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER), 41-44. <https://doi.org/10.1109/ICSE-NIER.2019.00019>
- Digital.ai. (2024). The 17th state of agile report. *Analyst Reports - Digital.ai*. Recuperado de <https://digital.ai/resource-center/analyst-reports/state-of-agile-report/>
- Gaona-Cuevas, M., Guerrero, V., & Vera-Rivera, F. (2024). The Smart Product Backlog: A Classification Model of User Stories. *IEEE Access*, 12, 150008-150019. <https://doi.org/10.1109/ACCESS.2024.3478833>
- Mahdi, M., Zabil, M., Ahmad, A., Ismail, R., Yusoff, Y., Cheng, L., Naidu, H. (2021). Software Project Management Using Machine Learning Technique—A Review. *Applied Sciences*, 11(11), 5183. <https://doi.org/10.3390/app11115183>
- Mamatha, R., & Suma, K. (2021, October). Role of machine learning in software project management. *Journal of Physics: Conference Series*, 2040(1), 012038. <https://doi.org/10.1088/1742-6596/2040/1/012038>
- Meiliana, Daniella, G., Wijaya, N., Putra, N., & Efata, R. (2023, January). Agile software development effort estimation based on product backlog items. *Procedia Computer Science*, 227, 186-193. <https://doi.org/10.1016/j.procs.2023.10.516>
- Olivares, R., Noel, R., Guzmán, S., Miranda, D., & Munoz, R. (2024, May). Intelligent learning-based methods for determining the ideal team size in agile practices. *Bioengineering*, 9(5), 292. <https://doi.org/10.3390/biomimetics9050292>

- Prasetyo, M., Peranginangin, R., Martinovic, N., Ichsan, M., & Wicaksono, H. (2025, March). Artificial intelligence in open innovation project management: A systematic literature review on technologies, applications, and integration requirements. *Lecture Notes in Networks and Systems*, 1005, 200-213. <https://doi.org/10.1016/j.joitmc.2024.100445>
- Rodríguez Sánchez, E., Vázquez Santacruz, E., & Cervantes Maceda, H. (2023, March). Effort and cost estimation using decision tree techniques and story points in agile software development. *Mathematics*, 11(6), 1477. <https://doi.org/10.3390/math11061477>
- Schwaber, K., & Sutherland, J. (2020). La guía definitiva de Scrum: Las reglas del juego. Recuperado de <https://scrumguides.org/>
- Shameem, M., Nadeem, M., & Zamani, A. (2023, March). Genetic algorithm-based probabilistic model for agile project success in global software development. *Journal of Software: Evolution and Process*. <https://doi.org/10.1002/smr.2349>
- Sutherland, J. (2001). Manifesto for agile software development. Recuperado el 24 de noviembre de 2024, de <http://agilemanifesto.org/>
- Tiwari, S., Phonsa, G., & Malik, N. (2024, January). Estimation approaches of machine learning in Scrum projects. *Lecture Notes in Networks and Systems*, 731, 103-111. https://doi.org/10.1007/978-981-99-4071-4_9
- TUNE-UP Process. (2017). Recuperado el 01 de agosto de 2023, de <http://www.tuneupprocess.com/>
- Tuneupprocess. (2017). Worki. Recuperado el 18 de octubre de 2024, de <https://cliente.tuneupprocess.com/web/#/login>