

Herramienta de software para consulta de metadatos de repositorios Git

Trabajo Terminal No. ____ - ____

*Alumno: Cuevas Olvera Ian Axel**

Director: Alcántara Méndez Alberto Jesús

**e-mail: icuevas1801@alumno.ipn.mx*

Resumen – El monitoreo de la productividad de los desarrolladores de software es importante para garantizar que se estén utilizando de manera eficiente los recursos y se estén alcanzando los objetivos deseados. Se presenta una herramienta de software que recopila los metadatos de los repositorios de Git locales y los envía a un repertorio en la nube para poder monitorear la productividad y eficacia de un desarrollador. Se discute la importancia de una herramienta para ayudar a las empresas a mejorar la gestión de sus proyectos de software y se describen el diseño, desarrollo y pruebas de la herramienta, así como su implementación y mantenimiento.

Palabras clave – Git, monitoreo, productividad, Tecnologías para la Web, metadatos

1. Introducción

En el mundo actual, el desarrollo de software juega un papel crucial en el éxito de las empresas y organizaciones. Con el aumento de la dependencia en la tecnología, las empresas deben estar en constante innovación y evolución para mantenerse competitivas en el mercado. Sin embargo, el desarrollo de software también puede ser un proceso complejo y costoso. Es por eso que es importante monitorear la productividad de los desarrolladores de software para garantizar que se estén utilizando de manera eficiente los recursos y se estén alcanzando los objetivos deseados. Esto permite a las empresas identificar problemas de rendimiento y buscar soluciones para mejorarlo, aumentando la eficiencia y reduciendo costos.

Una herramienta de monitoreo de productividad de desarrolladores de software puede ayudar a las empresas a evaluar el rendimiento de sus empleados, planificar y asignar tareas de manera efectiva, y establecer metas y objetivos claros. Además, el monitoreo de la productividad [1] de los desarrolladores de software puede ayudar a identificar a los desarrolladores con mayor rendimiento, lo cual permite a la empresa tomar decisiones informadas sobre el personal y asignar tareas de manera estratégica.

En este trabajo se discutirá la importancia del monitoreo de la productividad de los desarrolladores de software y cómo esta herramienta puede ayudar a las empresas a mejorar la gestión de sus proyectos de software. Se describirá el diseño, desarrollo y pruebas de la herramienta, así como su implementación y mantenimiento. El objetivo final es proporcionar una solución eficaz y eficiente para monitorear la productividad de los desarrolladores de software a través de la recopilación de datos de Git, lo que ayudará a las empresas y equipos de trabajo a mejorar su rendimiento y alcanzar sus objetivos de manera eficiente.

En la Tabla 1 se presentan distintas herramientas que fueron implementadas con el fin de monitorear la productividad o de recopilar datos de los repositorios de Git.

CARACTERÍSTICAS	RescueTime [2]	Toggl [3]	GitMiner [4]	GitMetrics [5]	Solución propuesta
PRECIO EN EL MERCADO	\$12 USD / mes \$78 USD / año	\$8 USD / mes (Personal) \$13.35 USD / mes (Empresarial)	Software gratuito	Software gratuito	Software gratuito
FUNCION PRINCIPAL	Rastrea automáticamente el tiempo invertido en diferentes aplicaciones y sitios web.	Genera informes detallados sobre el tiempo invertido en diferentes tareas y proyectos.	Permite a los usuarios buscar información en los repositorios de Git mediante técnicas de minería de datos.	Genera estadísticas detalladas sobre el repositorio, incluyendo número de commits, número de archivos modificados, número de palabras, etc.	Obtiene distintos metadatos de los repositorios de Git locales.
SOFTWARE CON INTEGRACION	Trello, Asana y Google Calendar.	Trello, Asana y Google Calendar.	NO	NO	NO
VISUALIZACION DE DATOS	Genera informes detallados sobre el tiempo invertido en	Permite buscar información específica en los repositorios, como	Permite buscar información específica en los repositorios,	Genera gráficos y visualizaciones para ayudar a los usuarios a	Despliega la información recopilada de

	diferentes aplicaciones	nombres de usuario, correos electrónicos, comentarios y código.	como nombres de usuario, correos electrónicos, comentarios y código	comprender mejor las estadísticas	manera organizada. Genera gráficos para desplegar la información obtenida.
CODIGO ABIERTO	NO	NO	SI	SI	SI
SEGUIMIENTO DEL TIEMPO	Utiliza un rastreador de tiempo automático para que no tenga que iniciar y detener la aplicación cada vez que trabaja o toma un descanso.	Puede monitorear el tiempo dedicado a aplicaciones, sitios web, documentos y cualquier otra actividad informática.	NO	NO	Obtiene la información constantemente al ser un proceso que se ejecuta en segundo plano cada determinado tiempo
GESTION DE PROYECTOS	No monitorea tiempo dedicado a cada tarea	Descripción general de todos los proyectos	NO	Los informes producidos fusionan los registros de git de los diferentes informes y los tratan como un solo registro de git lógico.	NO
GESTION DE DISTRACCIONES	Bloquea aplicaciones en horas de trabajo productivas	Registra y detecta las distracciones	NO	NO	NO

Tabla 1 Resumen de productos similares

2. Objetivo

Desarrollar una herramienta de software que recopile los metadatos de los repositorios de Git locales y los envíe a la nube para visualizarlos en una interfaz web.

2.1 Objetivos específicos

Una vez que se ha identificado el objetivo general, debe dividirse en objetivos específicos para aclarar los puntos que deben hacerse para lograr ese objetivo. Los objetivos específicos de este trabajo son los siguientes:

1. Desarrollar un proceso que recopile los metadatos de los repositorios de Git.
2. Configurar un repertorio en la nube para almacenar metadatos.
3. Desarrollar un sistema de conexión entre la herramienta de software y la nube.
4. Desarrollar un API de consulta de datos.
5. Desarrollar una interfaz para visualizar la información obtenida.

3. Justificación

Monitorear la productividad de los desarrolladores de software es importante para garantizar que el equipo esté trabajando de manera eficiente y efectiva, y para asegurar que se están alcanzando los objetivos del proyecto. Uno de los principales beneficios del monitoreo de la productividad es que permite identificar cuellos de botella o problemas que estén retrasando el progreso del proyecto. Por ejemplo, si un desarrollador está teniendo dificultades para comprender una especificación o está lidiando con problemas técnicos, el monitoreo de la productividad puede detectarlo y tomar medidas para resolver el problema.

El monitoreo de la productividad también puede ayudar a aumentar la transparencia y la comunicación dentro del equipo de desarrollo. Al compartir información sobre la productividad, los miembros del equipo pueden identificar áreas en las que pueden colaborar o ayudarse mutuamente para aumentar la eficiencia. Además, el monitoreo de la productividad también es esencial para medir el rendimiento del equipo (en general) y de cada individuo (de forma particular) y puede ayudar a identificar los desarrolladores que son excepcionalmente eficientes o aquellos que necesitan mejorar.

Con frecuencia, los líderes de proyectos, directores y gerentes buscan supervisar la productividad de un equipo buscando una mejora para adaptar un plan de trabajo que pueda ser útil y alcanzable a los objetivos deseados. La autopercepción visual de un desarrollador puede ayudar a dimensionar la calidad de un trabajo propio, así como la administración del tiempo y recursos que destina a un proyecto. Un estudio empírico realizado a 81 desarrolladores de software en Microsoft revela que una medida importante de la productividad es la percepción personal [6], llamada productividad autoevaluada o autopercebida. Por lo que una herramienta de software que realiza un autodiagnóstico identificando el avance con relación al tiempo-productividad de un desarrollo permite obtener un punto de vista objetivo y mensurable.

Por otra parte, Git [7] es un sistema de control de versiones ampliamente utilizado en el desarrollo de software que permite registrar y rastrear los cambios en el código fuente a lo largo del tiempo. Sus ventajas principales son: colaboración entre los miembros del equipo, flexibilidad en la gestión de versiones y seguridad en la gestión del código. Es considerado como una de las herramientas más importantes en la industria del desarrollo de software.

Con este proyecto se pretende realizar una herramienta que permita recopilar distintos metadatos obtenidos de los repositorios de Git y depositarlos en la nube para que puedan ser consultados en una interfaz web, esto con el fin de poder monitorear la cantidad y calidad de trabajo de un desarrollador. El monitoreo de la productividad beneficia distintos ámbitos en el desarrollo del software, tales como:

- Toma de decisiones informadas y precisas.
- Identificación problemas de rendimiento y sus soluciones.
- Planificación y asignación de tareas.
- Establecimiento de metas y objetivos.
- Reducción de costos.

Para la implementación de este proyecto terminal se pondrán en práctica los conocimientos adquiridos en diversas unidades de aprendizaje de la carrera de Ingeniería en Sistemas Computacionales, entre las cuales destacan: Algoritmia y Programación Estructurada, Análisis de Algoritmos, Redes de Computadoras, Tecnologías para la Web, entre otras, así como también se hará uso de los conocimientos adquiridos en la unidad de Ingeniería de Software para la organización del proyecto.

4. Productos o resultados esperados

Este trabajo tiene como fin contar con una herramienta que permita monitorear la productividad y la eficacia de un desarrollador de software, esto a través de la recopilación de metadatos obtenidos de los repositorios de Git locales que serán enviados a la nube, a continuación, en la Figura 1, se muestra la arquitectura propuesta:

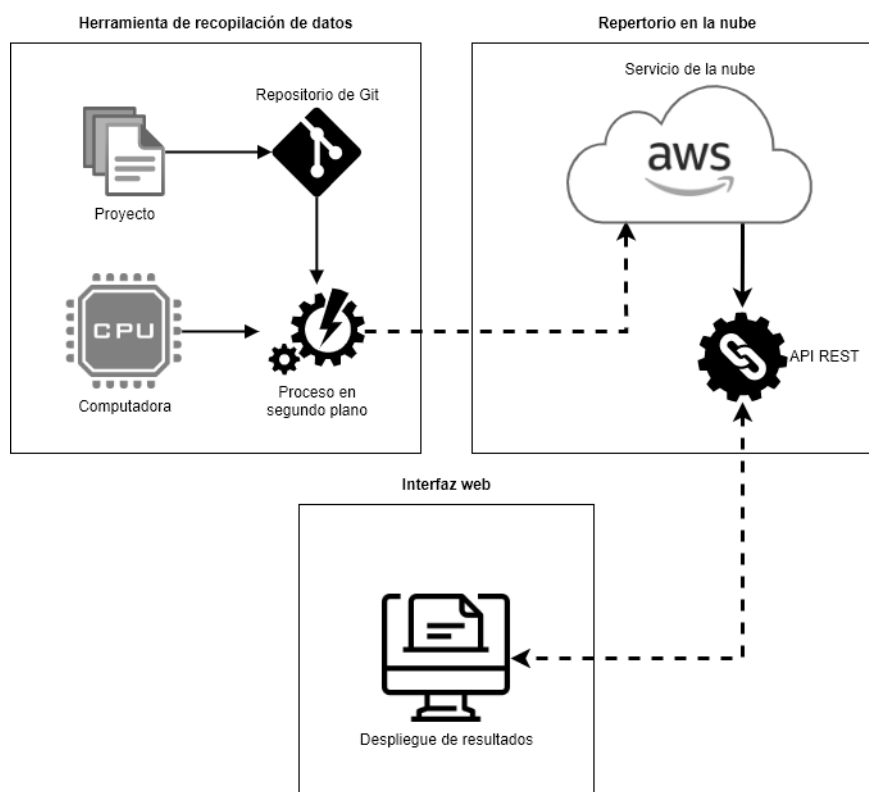


Figura 1 Arquitectura del sistema

Una vez explicada la forma en que se realizará el sistema, se pueden listar los productos que se esperan obtener:

1. Herramienta para el monitoreo de metadatos de los repositorios de Git.
2. Manual de usuario.
3. Manual técnico.
4. Código fuente

5. Metodología

Para desarrollar este Trabajo Terminal se hará uso de la metodología de cascada [8], la cual es un enfoque tradicional de desarrollo de software que se divide en varias fases secuenciales. Cada fase del proyecto debe ser completada antes de pasar a la siguiente.

- Fase de análisis: en esta fase se entrevista al cliente para recoger los requisitos del sistema, se realiza un estudio de viabilidad del proyecto y se crea un plan detallado para el desarrollo.
- Fase de diseño: en esta fase se crea el diseño detallado del sistema, se especifican los componentes y se hace una planificación del proyecto.
- Fase de construcción: en esta fase se escribe el código y se construyen las piezas individuales del sistema.
- Fase de pruebas: en esta fase se realizan las pruebas unitarias y de sistema para asegurar que el software funciona según los requisitos especificados.
- Fase de mantenimiento: una vez que el software se ha entregado al cliente, esta fase se encarga de proporcionar soporte y realizar cambios para mantener el sistema actualizado.

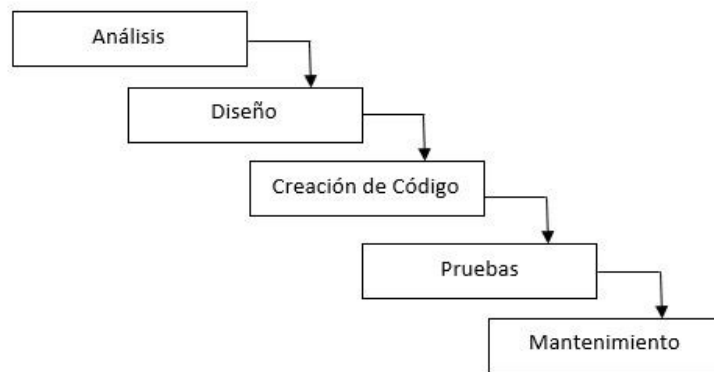


Figura 2 Diagrama de la metodología de cascada

La metodología de cascada (Figura 2) es una metodología en la que el proyecto avanza de manera secuencial, y cada fase es considerada como un acabado y no se regresa a ella, es una metodología recomendable para proyectos con requisitos conocidos y precisos. Aunque existe algunas limitaciones como puede ser el cambio de los requisitos en etapas tardías del proyecto.

6. Cronograma

Ver anexo 1

7. Referencias

- [1] V. Getov, Software Development Productivity: Challenges and Future Trends, Seoul, Korea (South): IEEE, 2010.
- [2] «RescueTime,» [En línea]. Available: <https://www.rescuetime.com/about>. [Último acceso: 10 1 2023].
- [3] Toggl, «Toggl,» [En línea]. Available: <https://toggl.com/mission/>. [Último acceso: 2023 1 10].
- [4] Github, «Git Miner,» [En línea]. Available: <https://github.com/UnkL4b/GitMiner>. [Último acceso: 10 1 2023].
- [5] Github, «Git Metrics,» [En línea]. Available: <https://github.com/Praqma/git-metrics>. [Último acceso: 10 1 2023].
- [6] Mind the Gap, «On the Relationship Between Automatically Measured an Self-Reported Productivity,» 14 12 2020. [En línea]. Available: <https://arxiv.org/pdf/2012.07428.pdf>. [Último acceso: 8 1 2023].
- [7] P. Upacharya, «Importance of Git,» 14 Enero 2017. [En línea]. Available: <https://www.linkedin.com/pulse/importance-git-pratik-upacharya/>.
- [8] P. C. J. G. L. H. Carlos Rivas, «Metodologías actuales de desarrollo de software,» *Revista Tecnología e Innovación*, vol. 2, n° 5, pp. 980-986, 2015.

8. Alumnos y directores

Cuevas Olvera Ian Axel. - Alumno de la carrera de Ing. en sistemas computacionales en ESCOM, especialidad sistemas, boleta: 2019630489, tel. 5548837265, email icuevas1801@alumno.ipn.mx

Firma: 

Alcántara Méndez Alberto Jesús. - Ingeniero en sistemas computacionales egresado de la ESCOM, maestría en ciencias en informática de UPIICSA, maestría en docencia científica y tecnológica del CIECAS IPN. Ha sido jefe de la UDI en la ESE IPN, presidente de la academia de electrónica analógica y presidente de la academia de fundamentos de sistemas electrónicos. Intereses: microelectrónica aplicada, redes neuronales y ciencia de datos. Email: ajalcantaram@ipn.mx

Firma: 

CARÁCTER: Confidencial
FUNDAMENTO LEGAL: Artículo 11 Fracc. V y Artículos 108, 113 y 117 de la Ley Federal de Transparencia y Acceso a la Información Pública.
PARTES CONFIDENCIALES: Número de boleta y teléfono.

Anexo 1

CRONOGRAMA

Nombre del Alumno: Cuevas Olvera Ian Axel

Título del TT: Herramienta de software para consulta de metadatos de repositorios Git

TT No.:

Actividad	FEB	MAR	ABR	MAY	JUN
Documentación					
Diseño del proceso que recopila los datos					
Diseño de los servicios API					
Diseño de la interfaz web					
Desarrollo del proceso que recopila los datos					
Configuración de un repertorio en la nube					
Implementación de la conexión al repertorio de datos					
Desarrollo de los servicios API					
Desarrollo de la interfaz web					
Pruebas					
Creación del manual de usuario					
Evaluación TTR					