DESARROLLO DE UN SISTEMA WEB PARA EL SEGUIMIENTO DEL TIEMPO DE EQUIPOS EN PROYECTOS DE SOFTWARE

Development of a web-based system for team time tracking in software projects

Oscar Agudelo-Varela, Msc[1]; Laura Parrado Villalobos[2]; Johan Leon Mora[3]
[1,2,3]*Universidad de los Llanos, Colombia, oscar.agudelo@unillanos.edu.co[1], laura.parrado.villalobos@unillanos.edu.co[2], johan.leon@unillanos.edu.co[3]*

Abstract– ***Companies must meet customer demands to remain competitive, which requires implementing technological solutions through IT. However, software development firms often struggle to meet delivery deadlines set by their clients. In such cases, they turn to outsourcing, a strategy that allows them to temporarily delegate specific tasks to external companies to ensure the timely delivery of the final product. Companies operating under this model must accurately track the hours invested in each project activity. This monitoring is not only crucial for optimizing processes and enhancing development quality but also essential for managing billing efficiently. Currently, many companies still perform this tracking manually on a monthly basis, leading to significant inefficiencies and increased workload on human resources. To address this challenge, a web-based system was developed to automate the tracking and management of time spent on various activities. This tool generates real-time reports, streamlining project management and facilitating data-driven decision-making.***

Keywords-Outsourcing, automation, software, time tracking.

Desarrollo de un Sistema Web para el Seguimiento del Tiempo de Equipos en Proyectos de Software

Oscar Agudelo-Varela, Msc[1]; Laura Parrado Villalobos[2]; Johan Leon Mora[3]
[1,2,3]*Universidad de los Llanos, Colombia, oscar.agudelo@unillanos.edu.co[1], laura.parrado.villalobos@unillanos.edu.co[2], johan.leon@unillanos.edu.co[3]*

Resumen– ***Las empresas deben satisfacer las necesidades de sus clientes para mantener su competitividad, lo que implica la implementación de soluciones tecnológicas a través de las TI. Sin embargo, las desarrolladoras de software suelen enfrentar dificultades para cumplir con los plazos de entrega establecidos con sus clientes. En estos casos, recurren a la subcontratación, un mecanismo que les permite delegar temporalmente ciertas actividades a empresas externas con el objetivo de garantizar la entrega del producto final.***

Las compañías contratadas bajo esta modalidad deben registrar con precisión las horas invertidas en cada actividad del proyecto. Este control no solo permite optimizar procesos y mejorar la calidad del desarrollo, también es fundamental para gestionar la facturación. Actualmente, muchas empresas realizan este seguimiento de manera manual y mensual, lo que genera un desgaste significativo en los recursos humanos.

Para abordar esta problemática, se desarrolló un sistema web que automatiza el seguimiento y control del tiempo invertido en las distintas actividades. Esta herramienta permite generar informes en tiempo real, optimizando la gestión de proyectos y facilitando la toma de decisiones.

Palabras clave- Subcontratación, automatización, software, seguimiento de tiempo.

I. Introducción

La subcontratación es una práctica usada por la industria a nivel mundial [1], debido a que muchas veces las organizaciones a razón de una mala planeación en los tiempos de ejecución o falta de personal idóneo, se ven comprometidos con los productos ofrecidos a los clientes, es allí cuando encuentran una solución en la subcontratación, siendo este un mecanismo en el cual una organización contrata a otra de forma temporal para desarrollar una cantidad definida de actividades [2], brindando así no solo una solución a los objetivos planteados, si no también obteniendo beneficios como lo son el ahorrar dinero reduciendo los costes de infraestructura, contratando de forma temporal el personal profesional idóneo y transfiriendo el riesgo a la organización subcontratada. Así mismo, la organización contratante accede a conocimientos y tecnologías conocidas por la organización contratada, y esta última obtiene conocimientos del negocio de la organización contratante de los cuales se puede respaldar, abriendo nuevas oportunidades con otros clientes, por lo cual es un mecanismo que de ser bien ejecutado, resulta en una ganancia en varios sentidos para ambas organizaciones [3].

Los procesos internos de los proyectos de software como lo son el definir, medir y controlar las variables, métricas e indicadores, se han vuelto importantes para la toma de decisiones en cuanto al mejoramiento de los procesos empresariales, proporcionando una base cuantitativa sobre la cual soportar la toma de decisiones [4], obteniendo así reducción de costos, esfuerzo y aumento de la calidad [5], pero el medir el tiempo de desarrollo de las diversas actividades del proyecto no solo es importante para la gestión interna, sino que también es importante para las empresas desarrolladoras de software que prestan servicios bajo la modalidad de subcontratación, ya que el método de pago utilizado por las empresas contratantes es el pago por horas trabajadas, de modo que para recibir la remuneración por el trabajo hecho, hay que respaldarlo no solo con el producto final, sino con un documento en donde se especifique las actividades y el tiempo que se tomó para desarrollarlas.

Existen diversas herramientas utilizadas para el seguimiento del tiempo, entre las más usadas resaltan WakaTime, Timing, Clockify y Monday P.M, las cuales no están enfocadas en su totalidad al desarrollo de software y algunas veces el costo de utilizar estas herramientas puede llegar a ser alto a pesar de no brindar una solución total a las necesidades de los desarrolladores de software. A continuación en la Tabla I se presenta una comparativa de las herramientas mencionadas.

TABLA I
Comparativa de características

Herramienta / Características	**WakaTime**	**Timing**	**Clockify**	**Monday P.M.**
Determinar aplicaciones abiertas, títulos y URL de los navegadores abiertos	No lo permite	Lo permite	No lo permite	No lo permite
Registro de tiempo del uso de cada aplicación	Lo permite solo a nivel de editor de texto	-	-	Lo permite a nivel de

Herramienta / Características	WakaTime	Timing	Clockify	Monday P.M.
activa				actividad (manualmente)
Obtener nombre de branches, commit, tiempo y archivos editados en el repositorio (datos del uso de Git)	Lo permite	No lo permite	No lo permite	No lo permite
Mostrar resumen de tiempo	Lo permite	Lo permite	Lo permite	Lo permite
Generar reportes	Lo permite	Lo permite	Lo permite	Lo permite
Costo mensual	$12 USD / Usuario	$12 USD / Usuario	Gratis	$16 USD / Usuario

Por lo que se buscó llevar el seguimiento y control del tiempo invertido en actividades realizadas por el recurso humano en los diferentes proyectos de las empresas mediante un software que optimice los procesos como el seguimiento de tiempo en las actividades durante el desarrollo de un proyecto, la producción, gestión y generación de informes que permitan identificar aspectos a mejorar en los procesos de una empresa, aumentar la calidad de los productos finales y satisfacer las necesidades de los clientes a la vez que se evidencia detalladamente el trabajo realizado por la empresa para realizar la diligencia del proceso de cobro.

II. Descripción del trabajo

Para el desarrollo del sistema web se utilizó Django como framework de desarrollo en la parte del backend, en frontend, se empleó Vue 3, y para el almacenamiento de datos, se utilizó el sistema de gestión de bases de datos PostgreSQL versión 11.

En la Fig 1 se ilustra el Modelo Entidad Relación (MER) y los nombres de los paquetes.

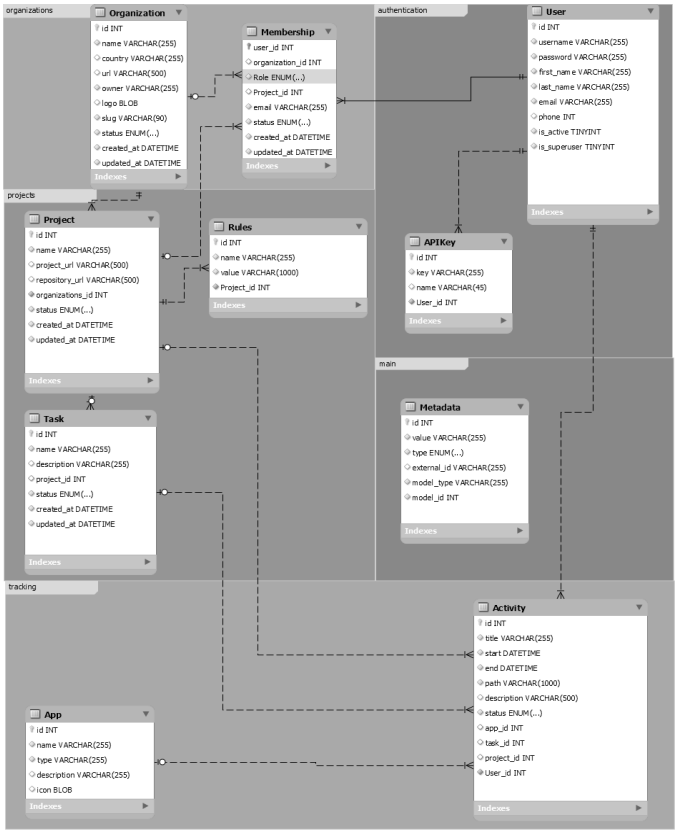

Fig. 1 Modelo Entidad Relación

En primera instancia, se trabajó en el backend, se crearon cinco aplicaciones con el Framework, junto con los modelos correspondientes para cada una.

Una vez definidos los modelos y sus interrelaciones, se desarrollaron los endpoints necesarios en cada aplicación, permitiendo operaciones mediante los métodos HTTP: POST para almacenar datos, GET para obtener información, PUT para modificar registros y DELETE para eliminarlos.

Para agregar un nivel adicional de seguridad y manejo de sesiones en las peticiones del usuario al servidor, se utilizó la biblioteca OAuth de Django. OAuth es un estándar que delega la autenticación del usuario a un servicio gestor de cuentas, que en este caso es el mismo servidor.

El proceso sigue un flujo en el que el cliente proporciona un Authorization Grant. Para este desarrollo, se eligió el método de usuario y contraseña. Una vez que el servidor recibe el Authorization Grant, devuelve un Authorization Code, que el cliente intercambia automáticamente por un Access Token. Este token se almacena y se envía en cada petición al servidor; de lo contrario, la solicitud no será atendida.

Además, dado que el sistema debía contar con cuatro roles distintos (owner, administrador, colaborador y cliente), y cada uno tendría diferentes niveles de permisos sobre la

organización, se decidió implementar un sistema de permisos tanto a nivel de modelo como a nivel de objeto según el rol. Aunque Django cuenta con una librería para la gestión de permisos a nivel de objeto, su implementación puede resultar compleja. Por ello, se optó por desarrollar un sistema de permisos más flexible y adaptado a las necesidades del proyecto, asegurando una implementación más sencilla y centralizada en un solo archivo para facilitar el mantenimiento del código.

III. Resultados

Una vez finalizado la primera versión del sistema web para el seguimiento del tiempo de equipos en proyectos de software, se obtuvo lo siguiente:

1. Un documento con 21 requerimientos funcionales y 5 no funcionales, el cual sirvió para limitar el alcance del proyecto, especificar los requisitos mínimos con los que debía contar la primera versión y establecer el tiempo de desarrollo de las actividades, permitiendo estimar un total de tiempo de desarrollo del proyecto.
2. Un documento con la arquitectura de software utilizada para conocer las tecnologías que se utilizarían en este proyecto y la interacción entre ellas.
3. Un servidor con 5 apps que contienen 9 tablas en total, siendo estas las tablas de usuario, APIkey, membresía, organización, regla, proyecto, tarea, actividad y aplicación, las cuales contienen campos que junto con la empresa Altix se consideraron relevantes para la gestión de los datos en la base de datos a través de la API REST. Por otra parte, cuenta con un sistema de permisos a nivel de objeto el cual limita el acceso, modificación o eliminación de contenido dependiendo del rango que posea el usuario.
4. Un componente en Vue 3 llamado Dashboard que muestra métricas como la cantidad total de proyectos, tiempo total de actividades "trackeadas", tiempo total de actividades asignadas y el número total de actividades "trackeadas", también representa por medio de gráficas de barras y de pastel los indicadores claves de rendimiento, como lo son el total de horas por día de actividades asignadas y sin asignar a alguna tarea o proyecto, y el total de horas por proyecto, pudiendo filtrar estas gráficas por mes o semana actual, en la Figura 2 se observa el Dashboard.

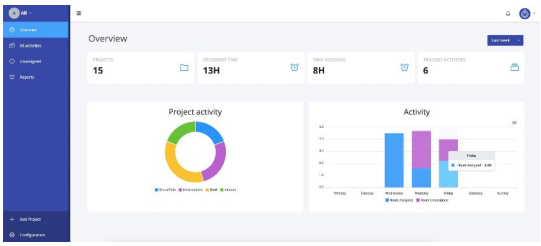

Fig. 2 Dashboard con sus respectivas métricas

5. Un modulo que permite seleccionar las tareas asociadas a un cliente, proyecto o ambos, especificar la divisa y el valor monetario por hora con la que se calcula el valor a devengar por actividad y opcionalmente filtrar por mes, el modulo cuenta con una lista de columnas definidas como lo son el nombre de las tareas, la descripción, el tiempo que requirió cumplir la tarea y el valor de esta tarea, finalmente permite exportar informes en formato pdf o excel, la Figura 3 ilustra la pantalla de generación de reportes.

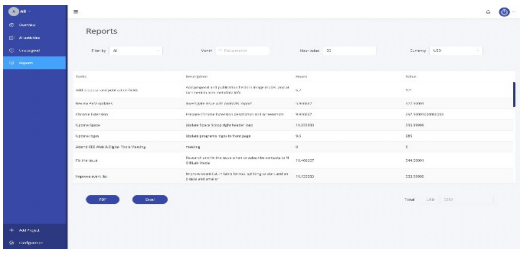

Fig. 3 Modulo de generación de reportes.

IV. Conclusiones

Los requerimientos funcionales y no funcionales permiten limitar el alcance del proyecto al especificar las funcionalidades mínimas con las que debía contar la primera versión del sistema web.

Los requerimientos funcionales y no funcionales, permiten elaborar el cronograma de las actividades, delegar responsables y establecer un tiempo de desarrollo, pudiendo así estimar un total de tiempo de desarrollo del proyecto.

El elaborar un artefacto de la arquitectura del software permite seleccionar las tecnologías idóneas para el desarrollo de un proyecto, identificando desde un principio las ventajas y restricciones de cada una, definiendo así el flujo de interacción entre las tecnologías

La API REST permitió separar la lógica de negocio en el servidor y el lado del cliente, dando la flexibilidad de consumir la información por diferentes tipos de aplicaciones.

La creación de los diferentes endpoints en la API REST aseguraron la comunicación entre el backend y el frontend, llevando a cabo las diferentes consultas, inserciones, modificaciones y eliminaciones en la base de datos.

La aplicación de filtros en los endpoints facilitan el manejo y búsqueda de información, lo que permite brindar una mejor usabilidad de los datos.

Una API REST posee la ventaja de ser escalable y flexible, permitiendo agregar nuevos endpoints en versiones futuras, y en caso de algún fallo en endpoints ya existentes, no afectar a los demás.

Las métricas e indicadores claves de rendimiento(KPI) definidos en el Dashboard consiguieron brindar una visión general del trabajo efectuado en la organización. Permitiendo así establecer límites y metas para el cumplimiento de tiempos en los diferentes proyectos.

REFERENCES

[1] D. Arias-Aranda, O. F. Bustinza, V. Barrales-Molina, "Operations flexibility and outsourcing benefits: an empirical study in service firms," Serv. Ind. J., vol. 31, no. 11, pp. 1849–1870, 2011.

[2] M. Belcourt, "Outsourcing — The benefits and the risks," Hum. Resour. Manag. Rev., vol. 16, no. 2, pp. 269–279, 2006.

[3] C. Simo˜es and G. Santos, "IT workforce outsourcing benefits, challenges and success factors in the customer and supplier perspectives," in 19th Brazilian Symposium on Software Quality, 2020.

[4] E. B. Belachew, "Analysis of software quality using software metrics," Int. j. comput. sci. appl., vol. 8, no. 4/5, pp. 11–20, 2018.

[5] Vásquez, Diana Calvache, César Collazos, César Pino, Francisco. (2010). Modelo liviano de medidas para evaluar la mejora de procesos de desarrollo de software MLM-PDS. Ingeniería y Ciencia.