



UNIVERSIDAD DE CHILE

FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS

DEPARTAMENTO DE INGENIERÍA INDUSTRIAL

**CREACIÓN, DISEÑO E IMPLEMENTACIÓN DE PANELES DE CONTROL PARA
EL ANÁLISIS, GESTIÓN Y MONITOREO DE LAS TAREAS DE NEOLAB EN SPACE**

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERA CIVIL INDUSTRIAL

DIANA BELÉN MARTÍNEZ CASTILLO

PROFESORA GUÍA:

ROCÍO RUIZ MORENO

MIEMBROS DE LA COMISIÓN:

ANDRÉS GORMAZ CANAVE

FELIPE VILDOSO CASTILLO

SANTIAGO DE CHILE

2024

CREACIÓN, DISEÑO E IMPLEMENTACIÓN DE PANELES DE CONTROL PARA EL ANÁLISIS, GESTIÓN Y MONITOREO DE LAS TAREAS DE NEOLAB EN SPACE

NeoLab, una empresa de desarrollo de software fundada en 2022, ofrece una amplia gama de servicios que abarcan todas las etapas del ciclo de vida de un proyecto web o de software. Esto incluye desde el diseño de la interfaz de usuario hasta la implementación y puesta en marcha de la solución tecnológica. Sin embargo, la empresa enfrenta desafíos significativos en la gestión eficiente de sus proyectos mediante su plataforma de gestión Space. Estos desafíos han resultado en retrasos en la entrega de proyectos. La falta de visibilidad clara sobre las tareas específicas y el tiempo dedicado a cada una ha dificultado que la gerencia identifique y gestione estos retrasos de manera oportuna.

Para solucionar este problema, se propone diseñar y desarrollar paneles de control para que la gerencia pueda identificar proyectos con retrasos y tomar medidas correctivas en menos de una semana, de modo de reducir el impacto económico que puedan implicar los retrasos. Para la realización de este proyecto, se seguirá una metodología de Business Intelligence. Para robustecer este proyecto, también se incluirá una etapa de gestión del cambio basada en el Modelo ADKAR.

Los resultados de este proyecto incluyen la creación de tres paneles de control con visualizaciones clave: seguimiento de tareas atrasadas por desarrollador, tiempo dedicado a tareas específicas por cada proyecto, entre otros. Estas visualizaciones permiten a la empresa identificar y abordar oportunamente tareas atrasadas, una capacidad que anteriormente no tenía debido a la falta de visibilidad sobre los retrasos. Esta capacidad de supervisión, previamente inalcanzable, ha permitido que NeoLab gestione eficazmente el 60.26% de los issues atrasados en menos de una semana, lo que marca una mejora notable en la eficiencia operativa.

El proyecto ha puesto de manifiesto varias fortalezas importantes. La implementación de los paneles de control ha permitido una redistribución más eficiente del personal, mejorando la asignación de recursos y optimizando la productividad. Además, proporciona a la gerencia herramientas cruciales para tomar decisiones estratégicas, como la evaluación de nuevos clientes y la planificación de futuros proyectos. La capacidad de identificar y abordar rápidamente los retrasos también facilita la adaptación a nuevos desafíos y oportunidades, mejorando la gestión continua de proyectos.

Agradecimientos

Quiero expresar mi más sincero agradecimiento a mi familia, que siempre ha estado a mi lado, brindándome apoyo y guiándome en mi desarrollo personal y académico. Sus enseñanzas y palabras de aliento, especialmente en los momentos difíciles, han sido vitales para mí. En particular, agradezco profundamente a mis padres, mi hermana y mi perrita, quienes han sido los pilares fundamentales en esta etapa de mi vida. Le agradezco a mi mamá, Elcira Castillo, por su amor y apoyo incondicional; a mi papá, Luis Martínez, por enseñarme a aspirar siempre a más y creer en mi capacidad para lograr grandes cosas; a mi hermana, Katherine Martínez, por su constante apoyo tanto en lo académico como en lo emocional; y a mi perrita, Nina, que me ha acompañado desde el inicio de mi carrera, trayendo alegría en los momentos que más lo necesitaba. No encuentro palabras suficientes para expresar mi admiración y agradecimiento hacia ellos. Sin su presencia y apoyo, no habría llegado hasta donde estoy hoy.

También quiero reconocer a la profesora Rocío Ruiz, quien ha sido fundamental en este proyecto. Su disponibilidad para resolver dudas, su disposición para discutir diferentes temas y sus valiosos consejos en cada etapa del proceso han sido de gran ayuda para mí.

Finalmente, quiero agradecer a mis amigas más cercanas, Laura Werner, Camila Contreras y Catalina Miranda, quienes me han brindado su apoyo a lo largo de toda la carrera y con quienes he compartido momentos muy bonitos.

Tabla de contenido

1.	Antecedentes generales	1
2.	Descripción del problema u oportunidad	2
3.	Descripción y justificación del proyecto	5
4.	Objetivos	7
4.1.	Objetivo general	7
4.2.	Objetivos específicos	7
5.	Alcances	8
6.	Marco conceptual	9
6.1.1.	Business Intelligence	9
6.1.2.	Etapas dentro del BI	9
6.1.3.	Ventajas y desventajas del BI	12
6.1.4.	Gestión del cambio: Modelo ADKAR	13
7.	Metodología	15
8.	Desarrollo	16
8.1.	Recolección de datos	16
8.2.	Almacenamiento	17
8.3.	Procesamiento y análisis de datos:	18
8.4.	Visualización de los datos	21
8.4.1.	Definición de plataforma	21
8.4.2.	Definición de paneles de control	22
8.4.3.	Panel de monitoreo de horas por proyecto:	23
8.4.4.	Panel de seguimiento para desarrolladores:	24
8.4.5.	Panel de monitoreo gerencial:	25
8.4.6.	Utilidad de cada uno de los paneles de control	27
8.4.7.	Implementación de la solución	28
8.5.	Gestión del cambio	30
9.	Discusión	32
10.	Conclusiones	34
11.	Bibliografía	36
12.	Anexos	37

Índice de tablas

Tabla 1. Tiempo en días de retraso promedio por trabajador. Fuente: Elaboración propia en base a los datos de NeoLab. 19

Tabla 2. Ejemplo de cantidad de horas por proyecto para el Empleado 1. Fuente: Elaboración propia en base a los datos de NeoLab. 24

Índice de ilustraciones

Figura 2.1. Visualización de un issue en Space. Fuente: Elaboración propia en base a la información de NeoLab en Space.	3
Figura 8.1. Flujo de trabajo en Apache NiFi. Fuente: Elaboración propia en base a los datos de la empresa.	17
Figura 8.2. Tabla de datos en MariaDB. Fuente: Elaboración propia en base a los datos de la empresa.	18
Figura 8.3. Atributos de la tabla estados. Fuente: Elaboración propia en base a los datos de la empresa	20
Figura 8.4. Visualización de horas por proyecto. Fuente: Elaboración propia en base a los datos en Metabase.	23
Figura 8.5. Visualización de panel de seguimiento para desarrolladores. Fuente: Elaboración propia en base a los datos en Metabase.	25
Figura 8.6. Visualización de panel de monitoreo gerencial. Fuente: Elaboración propia en base a los datos en Metabase.	26
Figura 8.7. Visualización de tareas sin Due Date. Fuente: Elaboración propia en base a los datos en Metabase.	27

1. Antecedentes generales

NeoLab, fundada en 2022, es una empresa especializada en el desarrollo de soluciones tecnológicas a medida para una amplia gama de industrias. Surgió como resultado de la división de NeoSoft, una compañía establecida en 1997 que se dedica exclusivamente a proporcionar servicios de desarrollo de software para el sector bancario y financiero. Mientras NeoSoft mantiene su enfoque en el ámbito financiero, NeoLab se creó para expandir el alcance de servicios tecnológicos a un espectro más amplio de clientes. Su cartera abarca desde instituciones bancarias hasta pequeñas y medianas empresas en sectores diversos, incluyendo tecnología, medio ambiente y muchos otros.

El modelo de negocio de NeoLab consiste en ofrecer una amplia gama de servicios que cubren todas las etapas del ciclo de vida de un proyecto web o de desarrollo de software; esto es, desde el diseño de la interfaz de usuario hasta la implementación y puesta en marcha de la página web o software. Esta versatilidad les permite participar tanto en el aspecto visual y la experiencia de usuario, como también en el desarrollo y la programación de las aplicaciones. Para ejemplificar sus capacidades, NeoLab ha sido la encargada del desarrollo de sitios web para empresas como Cashboom (véase en <https://cashboom.fun/>) y Clínica Veterinaria Los Andes (véase en <https://www.cvla.cl/>). Estos proyectos involucraron el diseño visual de las páginas web, así como la construcción de la plataforma tecnológica subyacente.

Cada tipo de proyecto puede variar en tiempo y complejidad. Por ejemplo, los proyectos de desarrollo web incluyen tanto el diseño visual como la construcción tecnológica, mientras que los proyectos de innovación pueden implicar prototipos y exploración de nuevas tecnologías. Proyectos de análisis de datos, por su parte, pueden requerir extensivo procesamiento de información.

La empresa se destaca como un socio estratégico para sus clientes, brindando un acompañamiento que abarca desde el levantamiento de requerimientos (con requerimiento se refiere a la solicitud que hace un cliente) hasta la implementación y puesta en marcha de las soluciones desarrolladas. Además, la empresa funciona como asesora del cliente, ayudando en el análisis y gestión de las páginas web desarrolladas.

NeoLab cuenta con un equipo reducido de alrededor de 15 personas, entre las que se encuentran Ingenieros Civiles Industriales e Ingenieros Civiles en Computación. Debido a su reciente creación y tamaño, NeoLab tiene una participación de mercado inferior al 1% en el sector del desarrollo de software. Este mercado se caracteriza por una alta competencia, con numerosas empresas pequeñas que compiten a nivel nacional, sin intenciones de expandirse internacionalmente. Al tratarse de una microempresa tecnológica, la facturación de NeoLab no supera los 400.000 dólares anuales (Achá & Bravo, 2009).

2. Descripción del problema u oportunidad

NeoLab emplea Space, una plataforma colaborativa en línea desarrollada por JetBrains, para gestionar el levantamiento de issues (también conocidos como requerimientos). Estos issues abarcan una amplia gama de tareas, desde la corrección de errores y la incorporación de nuevas funcionalidades hasta la optimización de la interfaz visual en sitios web. JetBrains es conocida por sus herramientas como PyCharm, una aplicación ampliamente utilizada para el desarrollo web en Python, y DataGrip, una herramienta multiplataforma para trabajar con diferentes sistemas de bases de datos. Ahora bien, es importante destacar que estos issues son tareas internas que no interactúan directamente con el cliente, sino que se ingresan dentro de esta plataforma para tener un control interno de las tareas que se realizan para un cliente.

Space ofrece una amplia gama de funcionalidades para la gestión de proyectos, entre las cuales destaca su capacidad para crear y asignar tareas, también conocidas como "issues", a los desarrolladores. Esta característica permite a NeoLab llevar un seguimiento inicial de los requisitos y el progreso de cada proyecto. Sin embargo, cabe destacar que no proporciona una visualización en tiempo real de la dedicación a estas tareas ni muestra cualquier tipo de error que pueda surgir durante su desarrollo.

Ahora bien, los issues representan tareas o solicitudes de mejora que se hacen a través de la plataforma Space de JetBrains. Un issue es una unidad básica para el seguimiento de problemas, tareas o mejoras en un proyecto. Puede representar un bug que necesita ser corregido, una nueva característica que se debe implementar en una página web o cualquier otra tarea que requiera atención.

NeoLab clasifica los issues en tres tipos:

1. Tareas administrativas: Incluyen actividades relacionadas con la gestión del proyecto y la coordinación, como envío de correo electrónicos o documentación de procesos.
2. Desarrollo de código: Estos issues están asociados con la programación y la implementación de nuevas funcionalidades o correcciones en el código. Incluyen todas las tareas relacionadas con el desarrollo y la depuración de software.
3. Diseño e implementación sin código: Este tipo de issue abarca actividades que involucran la creación de mockups, el diseño de interfaces y otras tareas que no requieren programación directa.

Ahora bien, un issue contempla los siguientes puntos, independiente de qué tipo de issue sea:

- Título: Un resumen conciso y descriptivo del problema o tarea.
- Descripción: Detalles adicionales sobre el issue (si se requiere), que podría contener pasos para reproducir un problema o algún otro contexto adicional.

- Estado: Indica en qué etapa se encuentra el issue.
- Asignación: Indica a quién se ha asignado el issue para su resolución.
- Due Date: Indica la fecha en la que el issue debe estar completado y desplegado para el cliente.
- Comentarios: Una sección para que los miembros del equipo y colaboradores dejen comentarios, archivos y/o actualizaciones relacionadas con el issue.

A continuación, se proporciona una vista general de cómo se visualiza un issue dentro de Space:

Nuevo issue

Título

Descripción

Comentarios

Presione para subir un nuevo archivo o para agregar un comentario

Asignación
Presione para asignar un responsable

Estado
Presione para actualizar el estado

Due date
Presione para actualizar

Figura 2.1. Visualización de un issue en Space. Fuente: Elaboración propia en base a la información de NeoLab en Space.

Cada issue sigue un ciclo de vida que abarca desde su creación hasta su implementación y entrega final al cliente. Este ciclo se compone de varias etapas (Ver Anexo A), las cuales se explican a continuación:

- Open: Indica que el issue ha sido creado, pero todavía no se está trabajando en él.
- In Progress: Especifica que un miembro del equipo asignado está actualmente trabajando en el issue.
- Done: Especifica que un miembro del equipo asignado ha terminado la tarea.
- Waiting for QA: Indica que el issue está a la espera de recibir QA.
- Quality Assurance (QA): Después de completar el trabajo, el issue pasa a esta etapa para someterse a revisión por parte de otros miembros del equipo.
- Accepted: El issue ha pasado exitosamente por la etapa anterior y ha sido aprobado para su implementación.

- In Validation: Indica que el issue ha sido entregado al cliente y está siendo validado.
- Finished: Indica que el issue se considera resuelto y cerrado. No se esperan más cambios o trabajos en relación con este issue.

Sin embargo, NeoLab enfrenta desafíos significativos en la gestión eficiente de estos issues. Frecuentemente, los problemas surgen debido a la falta de actualización en los estados de estos o retrasos excesivos en finalizar un issue, lo que afecta los tiempos de entrega al cliente.

Además, no existe un seguimiento adecuado de los issues en Space, lo que dificulta conocer la dedicación en tiempo real que le toma a un desarrollador completar una tarea. Esto provoca dificultad para realizar un seguimiento preciso de las horas dedicadas por los miembros del equipo. Al no poder hacer un seguimiento correcto, es imposible detectar anticipadamente los retrasos en los proyectos antes de que ocurran.

Los retrasos en las entregas pueden tener consecuencias críticas para la empresa, especialmente cuando los clientes pagan por una cantidad de horas mensualmente, las cuales pueden ir desde unas 100 a 180 horas mensuales. Si un proyecto utiliza más horas de las que ha pagado el cliente, la empresa se ve obligada a continuar trabajando sin recibir un pago adicional hasta que se complete el encargo solicitado por el cliente. Esta situación cobra mayor relevancia cuando los retrasos son significativos. En algunos casos, se han registrado retrasos de hasta 3 meses en la entrega de proyectos. Esto implica que la empresa ha tenido que invertir 3 meses adicionales de trabajo en un proyecto ya contratado, sin recibir ningún pago extra durante ese período. En este sentido, NeoLab tiene la oportunidad de reducir considerablemente los retrasos en la entrega de proyectos. Actualmente, estos retrasos pueden extenderse desde un mes hasta tres meses, lo que representa un impacto financiero significativo para la empresa. Según la gerencia, cada hora de retraso implica una pérdida promedio equivalente a un tercio de una Unidad de Fomento (UF).

Para ilustrar la gravedad de esta situación, considérese un proyecto en curso que ejemplifica claramente la problemática. Este proyecto fue contratado por 180 horas mensuales durante cinco meses, totalizando 900 horas. Debido a la falta de un sistema de seguimiento preciso, solo se detectó que se habían excedido las horas contratadas cuando ya se habían invertido 54 horas adicionales no remuneradas, lo que representa una pérdida de alrededor de 678.000 pesos chilenos. Si bien esta cantidad podría parecer menor, equivale aproximadamente al 2,29% de los ingresos mensuales de la empresa, lo cual es significativo si se considera que esta pérdida *es en un solo en un proyecto*. Dado que en NeoLab hay varios proyectos en curso, alrededor de quince, y ninguno cuenta con un cálculo o registro adecuado, las pérdidas acumuladas podrían ser considerablemente mayores. Ahora bien, volviendo al ejemplo, la situación se agrava por la falta de un registro sistemático. De los tres desarrolladores asignados al proyecto, solo uno mantenía un registro informal de sus horas trabajadas. Esto significa que no existe un control real sobre las horas dedicadas por los otros dos desarrolladores, lo que dificulta aún más la cuantificación exacta del sobre costo (es decir, podrían haber sido incluso más de 54 horas).

3. Descripción y justificación del proyecto

El proyecto se enfoca en el desarrollo de una suite de paneles de control que tendrán la función de monitorear el trabajo de los desarrolladores en los distintos proyectos. Está compuesto por tres paneles de control, los cuales se explican a continuación:

Panel de monitoreo de horas por proyecto: El componente principal de esta suite será un dashboard que proporcionará un seguimiento detallado de las horas dedicadas a cada proyecto. Este panel de control permitirá un monitoreo preciso del cumplimiento de los plazos acordados con los clientes. Por ejemplo, si un cliente ha contratado 150 horas de trabajo mensual, el dashboard mostrará claramente cuántas horas se han utilizado efectivamente en el proyecto cada mes y cuántas horas quedan disponibles. Para esto, será necesario que los trabajadores ingresen correctamente sus tareas en Space para que el tiempo que utilizan en esas tareas sea medido correctamente.

Panel de seguimiento para desarrolladores: Este panel estará orientado a los equipos de desarrollo y les permitirá acceder a información detallada sobre las tareas asignadas. Los desarrolladores podrán visualizar los issues pendientes y los plazos estimados.

Panel de monitoreo gerencial: Este panel estará diseñado para ofrecer a la gerencia una visión clara del desempeño de los equipos de desarrollo. Actuará como una herramienta complementaria al dashboard principal, proporcionando información clave sobre el progreso de los issues, la cantidad de días de retraso y el rendimiento individual de los desarrolladores.

Para fortalecer la efectividad de estos paneles, se desarrollará un sistema de alertas automatizado que enviará notificaciones por correo electrónico a la gerencia y los desarrolladores en caso de detectarse anomalías, como retrasos excesivos en la finalización de un issue u otras situaciones que requieran atención inmediata. De esta manera, se garantiza una gestión más eficiente y proactiva de los proyectos en curso.

La importancia de este proyecto reside en la necesidad de NeoLab de evaluar la rentabilidad de sus proyectos. Actualmente, la empresa depende de la información proporcionada verbalmente por los desarrolladores para medir la dedicación en horas, careciendo de un sistema que garantice la veracidad de estos datos. Esta falta de estructura en el seguimiento de los issues representa un desafío significativo, generando una gestión de proyectos deficiente y una falta de visibilidad sobre el avance del trabajo. La implementación de esta suite de paneles de control no solo permitirá a la gerencia conocer las horas reales dedicadas a cada proyecto, sino que también proporcionará a todos los involucrados las herramientas necesarias para identificar y abordar los obstáculos que ralentizan el proceso. Esto posibilitará la adopción de medidas correctivas de manera proactiva, evitando retrasos significativos en los proyectos.

Además, este proyecto contempla minimizar los costos mediante la utilización de herramientas de código abierto (open-source) gratuitas, evitando así gastos innecesarios para la empresa. Si bien existen soluciones de pago como Power BI y Tableau que podrían cumplir con los requerimientos, su implementación implicaría desembolsos que, dada la situación actual en la cual desean minimizar sus costos, no resultan viables. Por esta razón, el enfoque de utilizar herramientas gratuitas y open-source hace que este proyecto sea particularmente útil e importante, ya que brinda una solución sin incurrir en costos adicionales.

4. Objetivos

4.1. Objetivo general

Diseñar y desarrollar paneles de control que permitan a la gerencia de NeoLab detectar de forma anticipada aquellos issues que presentan retrasos en sus entregas, posibilitando la toma de acción en un plazo inferior a una semana.

4.2. Objetivos específicos

- Extraer, guardar y transformar los datos de la empresa de modo de consolidarlos en tablas en un servidor. Esta base de datos servirá como fuente principal para alimentar y diseñar los paneles de control.
- Diseñar la suite de paneles de control en conjunto con la gerencia y los desarrolladores, de modo de consolidar en los paneles toda la información relevante para la empresa.
- Desarrollar la suite de paneles de control automatizados con alertas por correo electrónico, que brinden visibilidad a los desarrolladores sobre sus tareas y permitan a la gerencia hacer un seguimiento continuo de los proyectos.
- Elaborar un plan de gestión del cambio para la implementación de los paneles de control dentro de la empresa, asegurando una transición fluida y una adopción adecuada por parte del personal.

5. Alcances

El objetivo principal de este proyecto es diseñar y desarrollar paneles de control para NeoLab que permitan identificar de forma anticipada los issues con retrasos en sus entregas. Estos paneles facilitarán la toma de decisiones dentro de un plazo inferior a una semana, enfocándose en identificar los proyectos que no cumplen con las horas mensuales pagadas por el cliente a causa de estos issues atrasados. En este sentido, es importante destacar que la solución no determinará directamente la rentabilidad de los proyectos, sino que la responsabilidad de tomar decisiones respecto a los resultados del proyecto recaerá en la empresa.

El proyecto se desarrollará en tres etapas específicas: diseño y desarrollo de los paneles de control, implementación técnica de la solución, y elaboración de recomendaciones para una adopción exitosa. Sin embargo, es crucial señalar que la ejecución del proceso de gestión del cambio queda fuera del alcance directo de este proyecto. La gerencia de la empresa será responsable de liderar y ejecutar la estrategia de adopción.

Existen algunas limitaciones y consideraciones importantes en el alcance del proyecto. La solución será aplicable exclusivamente a proyectos que utilicen Space, ya que depende de la información generada en esta plataforma. Además, debido a dificultades técnicas, la suite de paneles de control no proporcionará información en tiempo real, sino que se actualizará cuatro veces al día.

Un aspecto destacable es el potencial de expansión del proyecto. Dado que NeoSoft, empresa hermana de NeoLab, también utiliza Space y comparte características similares, existe una alta probabilidad de que la suite de paneles de control se pueda implementar en su entorno con modificaciones mínimas.

6. Marco conceptual

A continuación, se explican los distintos conceptos y definiciones que cobran relevancia para lograr llevar a cabo el objetivo principal que se plantea en esta memoria.

6.1.1. Business Intelligence

Se denomina Business Intelligence (BI) al conjunto de metodologías, prácticas y capacidades enfocadas al manejo de información que permite tomar mejores decisiones a las empresas (Silva, 2017). El objetivo principal del BI es permitir el acceso interactivo a los datos, facilitando la manipulación de estos y brindando a los gerentes y analistas la capacidad de realizar análisis apropiados. Al analizar datos históricos y actuales, la gerencia obtiene ideas valiosas que le permite tomar decisiones más informadas (Sharda et al., 2017).

6.1.2. Etapas dentro del BI

Las etapas que componen el Business Intelligence difieren en distintos autores, empresas de software y consultoras. Estas han ido evolucionando a medida que se consolidan nuevas tecnologías y tendencias estratégicas empresariales (Joyanes, 2019). Dado lo anterior, para esta memoria se considerarán las etapas de Business Intelligence con enfoque en el Big Data. Esta arquitectura está compuesta por cinco etapas (Ver anexo B): recolección de datos, almacenamiento, procesamiento de datos, visualización y administración (Duan, 2014).

1. *Recolección de datos*: En esta primera etapa, el sistema debe conectarse a las distintas fuentes de datos y extraer aquellos que posteriormente serán utilizados en las siguientes etapas (Joyanes, 2019). Según Joyanes, el método de extracción de datos puede dividirse en dos grupos, los cuales se explican a continuación:
 - a. *Batch o por lotes*: El método de extracción consiste en conectarse a la fuente de datos periódicamente (una vez al día, por ejemplo), buscando nueva información.
 - b. *Streaming o transmisión en tiempo real*: El método de extracción consiste en estar conectado en tiempo real a la base de datos, descargando información cada vez que esta se transmite.

En la etapa de recolección de datos, los datos extraídos pueden someterse a transformaciones o modificaciones si es necesario, como el filtrado de información irrelevante.

NeoLab utiliza un sistema de recolección de datos por lotes (batch) para obtener información sobre los issues. Este método implica conectarse periódicamente a la fuente de datos y buscar nuevos registros. Específicamente, el sistema se conecta cuatro veces al día para extraer los datos más recientes sobre los issues. Para realizar esta tarea, NeoLab emplea un script de Python que se conecta a los datos de Space mediante una API. Las API son mecanismos que permiten a dos componentes de software comunicarse entre sí mediante ciertas definiciones y protocolos.

1.1. Sobre Python:

Python es un lenguaje de programación de alto nivel, interpretado y multipropósito (Fernández, 2013). Que sea interpretado significa que no es necesario compilar el código para su ejecución, ya que existe un intérprete que se encarga de leer el fichero fuente y ejecutarlo. Gracias a esta característica es posible ejecutar el mismo código en distintas plataformas y sistemas operativos, sin necesidad de cambiar el código fuente. Para su funcionamiento solamente bastaría con tener instalado el intérprete.

Habitualmente, se les llama script a los programas en Python (Fernández, 2013). Ahora bien, un script de Python es aquel archivo que contiene el código fuente de Python y que está destinado a iniciarse desde la línea de comandos. En otras palabras, un programa puede tener uno o más scripts.

2. *Almacenamiento:* En la etapa de almacenamiento, los datos recolectados son almacenados de manera estructurada en una base de datos, permitiendo su posterior acceso y manipulación (Joyanes, 2019). En el caso de NeoLab, los datos son almacenados en una base de datos relacional llamada MariaDB utilizando Apache NiFi.

2.1. Sobre MariaDB:

MariaDB es un sistema de gestión de bases de datos relacionales de código abierto, lo que significa que su código fuente es público y puede ser modificado y distribuido libremente bajo la licencia GNU. Se creó como un proyecto derivado o "fork" de MySQL por Michael Widenius, quien fue el creador original de MySQL (Kenler, 2015).

Al igual que MySQL, MariaDB permite crear, gestionar y modificar bases de datos relacionales, así como definir su estructura de datos (Kenler, 2015). Además, brinda la capacidad de realizar consultas sobre los datos almacenados mediante el uso del lenguaje estándar SQL (Structured Query Language). Estas consultas pueden ser escritas manualmente o generadas por programas, lo que posibilita la interacción con MariaDB desde aplicaciones de diversos tipos.

2.2. Sobre Apache NiFi:

Apache NiFi es un proyecto de software de Apache Software Foundation, diseñado para el procesamiento y distribución de datos. Este sistema se destaca por su fiabilidad y seguridad en la transferencia de datos entre diferentes sistemas. Gracias a su escalabilidad, es posible utilizar múltiples servidores NiFi en un clúster para manejar grandes volúmenes de datos (Racka, 2022). La función que desempeña Apache NiFi en el contexto de la empresa corresponde a transferir la información que se obtiene de Space (utilizando Python) a MariaDB.

3. *Procesamiento y análisis:* Una vez que se tienen los datos almacenados, el siguiente paso en un sistema Big Data es explotar la información para llegar a los datos deseados. Para esto es necesario procesar la información y analizarla. Aquí, los datos brutos son procesados para su limpieza y transformación, y luego son sometidos a diversos métodos analíticos para descubrir patrones, tendencias y relaciones significativas. Una vez completado el análisis, la información resultante es utilizada para identificar y seleccionar los datos relevantes que satisfagan las necesidades específicas del análisis, proporcionando así insights valiosos para la empresa (Joyanes, 2019).

4. *Visualización:* En esta etapa los datos originales se han convertido en conocimiento y sus resultados se presentan a la empresa para su estudio y toma de decisiones correspondiente. En el caso de la empresa, la herramienta que se utilizará para desplegar la información es Metabase.

4.1. Sobre Metabase:

Metabase es una herramienta de inteligencia de negocios (BI) de código abierto que se puede conectar a numerosas bases de datos populares. Permite realizar consultas sobre los datos y presenta las respuestas en formatos comprensibles, ya sea mediante gráficos de barras, tablas detalladas u otros tipos de visualización.

Ahora bien, para lograr una buena visualización de los datos, es fundamental involucrar estrechamente a la gerencia de NeoLab desde las primeras etapas del proceso. Esto garantizará que la experiencia de usuario de los paneles de control se ajuste a sus necesidades y preferencias, asegurando así su adopción y utilización correcta. Asimismo, es crucial tener en cuenta que el panel de control destinado a los desarrolladores debe ser funcional para ellos, no simplemente una herramienta diseñada para que la gerencia ejerza control sobre su trabajo. Esto es fundamental para garantizar una adecuada adopción por parte de los desarrolladores.

Administración: La administración es un componente crucial que garantiza el funcionamiento adecuado y eficiente de todo el proceso de gestión de datos, desde la recolección hasta la visualización, asegurando la integridad, seguridad y calidad de los datos.

6.1.3. Ventajas y desventajas del BI

La ventaja más grande del BI es que ayuda a descubrir cosas que antes no se sabían. Existen distintos tipos de beneficios que entrega el BI, pero estos pueden dividirse en tres (Cano, 2007):

1. Beneficios tangibles, por ejemplo: reducción de costes, generación de ingresos, reducción de tiempos para las distintas actividades del negocio.
2. Beneficios intangibles: el hecho de que se tenga disponible la información para la toma de decisiones hará que más usuarios utilicen dicha información para tomar decisiones y mejorar la posición competitiva.
3. Beneficios estratégicos: Todos aquellos que facilitan la formulación de la estrategia, es decir, a qué clientes, mercados o con qué productos dirigirse.

Bajo el marco de los beneficios del BI según Cano (2007), NeoLab puede obtener varios beneficios significativos al implementar un sistema de seguimiento de horas. En términos de beneficios tangibles, la empresa puede lograr un importante ahorro de dinero al detectar rápidamente los proyectos que se están retrasando, permitiendo tomar medidas antes de que estos retrasos se vuelvan costosos. Además, este sistema facilitará un mejor uso del personal, ya que la empresa podrá identificar quién está sobrecargado o subutilizado en cada proyecto, y ajustar los equipos según sea necesario. Otro beneficio tangible es la capacidad de evaluar el rendimiento, permitiendo ver más fácilmente qué tan productivo es cada empleado.

En cuanto a los beneficios intangibles, pero igualmente importantes, NeoLab podrá obtener una mejor comprensión de sus proyectos, comparando fácilmente las horas planificadas con las realmente utilizadas. También contará con información centralizada, lo que significa que todos los datos estarán en un solo lugar, facilitando la toma de decisiones basadas en información actualizada.

A largo plazo, los beneficios estratégicos para NeoLab son cruciales para su crecimiento y posicionamiento en el mercado. La implementación de este sistema de BI permitirá una mejora significativa en la planificación de proyectos futuros, basándose en datos precisos de proyectos anteriores. Esto es particularmente relevante, ya que permitirá a la empresa entender mejor la duración real de los proyectos y, como resultado, hacer ofertas más precisas y llegar a mejores acuerdos con sus clientes.

Entre las desventajas significativas del BI se encuentra el considerable impacto en recursos que su implementación conlleva. La implementación de sistemas de BI demanda una

inversión sustancial en tiempo, capital y talento humano. Este proceso puede desviar la atención y energía de empleados clave de sus responsabilidades habituales, afectando la continuidad de proyectos en curso y la eficiencia de las operaciones habituales (Tunowski, 2015). En este sentido, un riesgo particular del BI radica en la posibilidad de que la alta dirección se enfoque excesivamente en la herramienta, descuidando otras áreas críticas de gestión. En el contexto específico de NeoLab, es crucial mantener un equilibrio adecuado, asegurando que la implementación y uso del BI no comprometa el tiempo y la atención que la gerencia debe dedicar a sus funciones regulares.

Según Tunowski (2015), la etapa de recolección de datos consume aproximadamente el 80% del tiempo total del proyecto, dejando solo el 20% restante para el análisis de estos datos. Esta desproporción en la asignación de tiempo puede potencialmente comprometer la profundidad y calidad del análisis posterior. En el caso específico de NeoLab, es crucial gestionar cuidadosamente esta distribución de recursos. Se debe asegurar que, a pesar del tiempo extensivo requerido para la recolección y preparación de datos, se reserve un período adecuado para el análisis exhaustivo de la información obtenida.

6.1.4. Gestión del cambio: Modelo ADKAR

La implementación de estos paneles de control requiere cambios conductuales tanto en la gerencia como los desarrolladores, ya que requiere que se adapten nuevas herramientas. En consecuencia, es de suma relevancia utilizar herramientas de gestión del cambio. La gestión del cambio se trata de lograr un diseño óptimo del camino que se seguirá para cumplir un objetivo, contemplando desde el punto de partida hasta el objetivo a lograr (Lauer, 2021).

En este caso, es fundamental que los miembros de la empresa utilicen correctamente las herramientas a nivel individual (por ejemplo, ingresando sus tareas adecuadamente en Space y actualizando constantemente su estado). Para lograr esto, se implementará el modelo ADKAR, que se enfoca en el cambio a nivel individual (Hiatt, 2006). ADKAR es un acrónimo que representa cinco elementos esenciales:

1. *Awareness* (Conciencia): Es crucial que los individuos comprendan por qué es necesario el cambio y cuáles son los riesgos de no llevarlo a cabo. Sin la debida importancia a esta etapa, es probable que no se tome en serio el proceso y se retomen los antiguos hábitos.
2. *Desire* (Deseo): La mera conciencia y el reconocimiento de la necesidad de cambio no son suficientes para lograr una transformación. Se necesita algo que motive a las personas y las impulse a actuar.
3. *Knowledge* (Conocimiento): Los individuos deben saber cómo realizar el cambio. Esto implica conocer las herramientas, recibir el entrenamiento adecuado y tener acceso a la información necesaria.
4. *Ability* (Habilidad): Este elemento se refiere al desarrollo de las habilidades necesarias para que los individuos puedan adaptar el cambio a sus funciones o actividades diarias.

5. *Reinforcement* (Reforzamiento): Un cambio exitoso requiere un refuerzo constante. Esto incluye establecer mecanismos para mantener y consolidar el cambio a largo plazo, celebrar los éxitos, reconocer los esfuerzos y proporcionar apoyo continuo.

Una de las principales ventajas del modelo ADKAR es su enfoque en la aceptación del cambio por parte de los empleados y los miembros del equipo del proyecto. Este proceso, que comienza y termina con los individuos como protagonistas del cambio, destaca la importancia de considerar sus perspectivas y necesidades (Galli, 2018). Sin embargo, una desventaja de este modelo es que, al centrarse en el aspecto humano del cambio, resulta más adecuado para equipos pequeños, en lugar de organizaciones a gran escala con procesos complejos (Galli, 2018). No obstante, en el caso de NeoLab, una empresa pequeña con 15 empleados, el enfoque en el individuo no presenta mayores complicaciones debido al reducido número de personas, facilitando así la implementación de este modelo.

7. Metodología

A continuación, se procede a detallar la metodología a seguir para la implementación del sistema de paneles de control que servirá a la gerencia y los desarrolladores de NeoLab.

1. **Recolección de los datos:** En primer lugar, se deberá determinar qué datos se necesitan para este propósito y solicitar los distintos accesos que puedan ser necesarios. En segundo lugar, se deberán extraer estos datos.
2. **Almacenamiento:** Como su nombre lo dice, los datos deberán ser almacenados para su posterior uso. En el caso de NeoLab los datos serán almacenados en MariaDB.
3. **Procesamiento y análisis de datos:** Una vez obtenida la data, se realizará un análisis con el fin de determinar el estado actual de los datos. De manera paralela, se harán todas las transformaciones necesarias para una mejor visualización de los datos. Es importante destacar el cuidado que se debe tener en el manejo de estos datos, ya que estos darán cuenta del desempeño de los desarrolladores dentro la empresa, por lo que cualquier error en el manejo de los datos podría causar problemas para el desarrollador, que podrían ir desde amonestaciones a despido de personal.
4. **Visualización de los datos:** En esta etapa, se diseñarán e implementarán los paneles de control utilizando Metabase, una herramienta de código abierto y gratuita. Se ha seleccionado Metabase debido a que NeoLab ya cuenta con experiencia previa en su uso, lo que facilitará su adopción y minimizará la curva de aprendizaje.
5. **Gestión del cambio:** La implementación de los nuevos paneles de control implicará cambios en los procesos y las prácticas actuales de NeoLab. Por lo tanto, es fundamental desarrollar e implementar un plan de gestión del cambio para garantizar una transición fluida y una adopción adecuada por parte del personal. En específico, se utilizará el modelo ADKAR, el cual tiene un enfoque a nivel individual.
6. **Administración:** El éxito de la implementación de la herramienta se evaluará mediante métricas de desempeño cuidadosamente definidas. Al finalizar el proyecto, estas métricas permitirán determinar si los paneles de control resuelven eficazmente el problema actual o si es necesario reconsiderar las alternativas empleadas hasta el momento. Sin embargo, esta etapa le corresponderá a la gerencia realizarla.

8. Desarrollo

8.1. Recolección de datos

La primera etapa dentro del marco conceptual corresponde a la recolección de datos, la cual la empresa realiza mediante un script de Python. Inicialmente, la empresa cuenta con un script llamado "Space.py", que realiza la función de extraer información de Space y guardarla en formato CSV. Este script proporciona una instantánea del estado actual de los issues en el momento de su ejecución.

A través de este script, se puede obtener una variedad de datos sobre cada issue:

1. Proyecto: Se refiere al proyecto al que pertenece el issue.
2. Responsable: Es la persona asignada para abordar y resolver el issue.
3. Participante: Son las personas que han tenido algún tipo de involucramiento en el issue, ya sea proporcionando información, contribuyendo a la solución o simplemente siguiendo su progreso.
4. Nombre: Es el título del issue.
5. Estado: Indica la situación actual del issue.
6. Resuelto: Esta información indica si el issue ha pasado satisfactoriamente por la etapa de control de calidad (QA), es decir, si ha sido aceptado como completado de manera satisfactoria.
7. CreationTime: Es la fecha y hora en que el issue fue creado.
8. LastModified: Indica la fecha en la cual un issue ha experimentado alguna modificación desde su creación.
9. DueDate: Es la fecha límite para la finalización del issue, es decir, la fecha en la que se espera que un issue esté marcado como "Finished".

Además de estos elementos iniciales, se identificó información adicional valiosa que sería beneficioso incorporar para permitir un seguimiento y análisis más completo de los issues. Después de consultar con la gerencia, y asegurarse que estuviesen de acuerdo, se decidió incluir los siguientes elementos:

1. IdSpace: Se añadió un identificador único para cada issue, ya que los detalles anteriores podrían no ser suficientes para distinguir entre issues con el mismo nombre en el mismo proyecto. Esto evita confusiones y facilita el seguimiento individual de cada issue.
2. Backlog: Se incluyó información sobre el "backlog", que son las tareas pendientes para futuras entregas y que actualmente no se están desarrollando.
3. DueDateDone: Además de la fecha de vencimiento, se agregó información sobre cuándo se espera que un issue esté completado y marcado como "Done".

4. Text: Este apartado exhibe un mensaje asociado a cada issue, los cuales abarcan una variedad de tipos como cambios de estado, asignación de responsabilidades, comentarios adicionales, entre otros (Ver Anexo C).
5. Time: Indica la hora en la cual ha ocurrido el mensaje dentro del issue.

8.2. Almacenamiento

Para la segunda etapa del marco conceptual, que consiste en el almacenamiento de datos, se utiliza Apache NiFi, una plataforma especializada en la extracción, transformación y carga de datos. En NiFi, se crea un flujo de trabajo que se detalla a continuación:

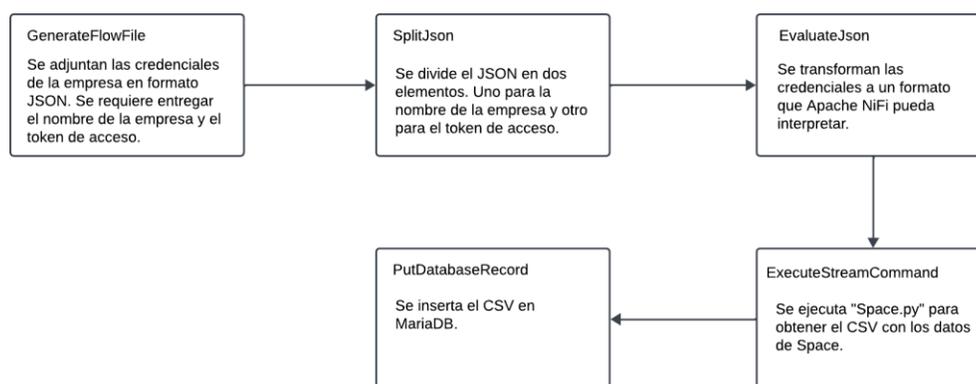


Figura 8.1. Flujo de trabajo en Apache NiFi. Fuente: Elaboración propia en base a los datos de la empresa.

Como se muestra en la Figura 8.1, el flujo de trabajo comienza con el componente "GenerateFlowFile". Este componente se encarga de obtener las credenciales necesarias para acceder a la información proporcionada por "Space.py". Luego, los datos pasan a través de los componentes "SplitJson" y "EvaluateJson", que permiten que NiFi interprete las credenciales proporcionadas por el usuario. A continuación, se ejecuta el componente "ExecuteStreamCommand", que ejecuta el script "Space.py" para obtener el archivo CSV actualizado. Una vez obtenido este archivo, se utiliza el componente "PutDatabaseRecord" para insertar los datos en MariaDB.

Finalmente, se configura una tarea programada (CRON) para ejecutarse diariamente a las 2 a.m., 9 a.m., 2 p.m. y 7 p.m., con el propósito de capturar el progreso diario de los "issues". Esto implica que el flujo de trabajo descrito en la Figura 8.1 se llevará a cabo a diario en los horarios mencionados. Estos horarios fueron seleccionados principalmente por las siguientes razones:

1. Algunos empleados de NeoLab trabajan durante la madrugada, lo que hace necesario actualizar los datos a las 2 a.m. para reflejar su actividad.
2. La jornada laboral comienza a las 9 a.m., por lo que es crucial que los datos estén actualizados al inicio del día.

3. A las 3 p.m. hay una reunión en la que los empleados de NeoLab presentan sus tareas a la gerencia. El CRON se ejecuta a las 2 p.m. para que tengan tiempo de revisar sus tareas y preparar el informe verbal para la gerencia.

4. Aunque la jornada laboral termina a las 6:30 p.m., puede extenderse un poco más. Por ello, el CRON se ejecuta a las 7 p.m. para capturar el estado de los "issues" al final del día.

Estos horarios fueron aprobados por la gerencia, quien también decidió que el CRON se ejecute solo cuatro veces al día.

En MariaDB, se diseña una tabla destinada al almacenamiento de los datos transferidos a través de Apache NiFi, cuya estructura se ilustra en la Figura 8.2:

issues	
<i>id_issue</i>	<i>int</i>
<i>idSpace</i>	<i>varchar(50)</i>
<i>nombre</i>	<i>varchar(50)</i>
<i>backlog</i>	<i>varchar(50)</i>
<i>responsable</i>	<i>mediumtext</i>
<i>participante</i>	<i>mediumtext</i>
<i>proyecto</i>	<i>mediumtext</i>
<i>estado</i>	<i>varchar(50)</i>
<i>resuelto</i>	<i>varchar(50)</i>
<i>creationTime</i>	<i>datetime</i>
<i>lastModified</i>	<i>datetime</i>
<i>dueDateDone</i>	<i>float</i>
<i>dueDate</i>	<i>float</i>
<i>text</i>	<i>longtext</i>
<i>time</i>	<i>datetime</i>
<i>fecha_insercion</i>	<i>datetime</i>

Figura 8.2. Tabla de datos en MariaDB. Fuente: Elaboración propia en base a los datos de la empresa.

Los atributos que se muestran en la Figura 8.2 son los mismos elementos obtenidos a través de Space.py. Sin embargo, se agregan dos elementos adicionales para fines de gestión y registro. En primer lugar, se crea una clave primaria denominada *id_issue*, la cual permite llevar un control único e identificar de manera inequívoca cada issue en el momento específico en que se registra. Además, se incorpora el campo *fecha_insercion*, cuyo propósito es almacenar la fecha y hora exactas en las que los datos del issue se insertan dentro de la tabla.

8.3. Procesamiento y análisis de datos:

Antes de diseñar la solución, es fundamental comprender la información disponible en Space. Por lo tanto, se analizaron las tablas obtenidas a través de Space.py y almacenadas en MariaDB. Del análisis de esta información, se identificaron varios aspectos relevantes:

- Se detectaron 77 issues en estado "Open" o "In Progress" creados después del 1 de enero de 2024 que aún no han sido completados. Esta observación respalda la existencia de retrasos significativos en la finalización de los issues.

- Se mide el tiempo de retraso promedio por trabajador, del cual se evidencia lo siguiente:

Trabajador	Retraso en días
Trabajador A	3
Trabajador B	4
Trabajador C	2
Trabajador D	4
Trabajador E	9.4
Trabajador F	4.9
Trabajador G	15.57
Trabajador H	34
Trabajador I	32
Trabajador J	33.3

Tabla 1. Tiempo en días de retraso promedio por trabajador. Fuente: Elaboración propia en base a los datos de NeoLab.

El análisis reveló un retraso promedio de 14 días en la finalización de los issues por parte de los trabajadores. Algunos casos extremos muestran retrasos de hasta un mes. Estos datos indican dos problemas principales:

1. Retrasos significativos en el cumplimiento de tareas.
2. Distribución desigual del trabajo entre los empleados.

La segunda conclusión se desprende del hecho de que hay trabajadores con retrasos mucho mayores que otros. Esto sugiere que algunas personas están sobrecargadas de tareas, lo que les impide completarlas a tiempo, mientras que otras podrían tener menos trabajo asignado.

Ahora bien, la disposición de los datos en la tabla "issues" presenta dificultades para identificar cuándo ocurre un cambio de estado en un issue. Aunque la columna "text" registra estos cambios, lo hace de manera desordenada y en un formato longtext que dificulta su procesamiento. Para abordar esta problemática y mejorar la organización de los datos, así como prevenir posibles pérdidas o daños en la tabla "issues", se ha decidido trasladar la gestión de los cambios de estado a una tabla adicional llamada "estados". En la nueva tabla "estados", se registra de forma ordenada y estructurada tanto el estado inicial como el estado final de cada issue, junto con su identificador único, nombre, proyecto asociado, responsable y la fecha en la que se realizó el cambio. Además, se han agregado dos nuevas columnas: "tiempo_ciclo" y "tiempo_arranque". Estas columnas representan el tiempo (en horas) que un issue tarda en pasar de "In Progress" a "Done" y el tiempo que tarda en pasar de "Open" a "In Progress", respectivamente.

Este enfoque ofrece varias ventajas: simplifica las consultas SQL relacionadas con los cambios de estado, mejora la legibilidad y la comprensión de los datos, y garantiza la integridad de la información en la tabla principal "issues".

estados	
id_estado	int
idSpace	varchar(50)
nombre_estado	varchar(50)
backlog_estado	varchar(50)
responsable_estado	mediumtext
proyecto_estado	mediumtext
estado_inicial	varchar(50)
estado_final	varchar(50)
creationTime	datetime
dueDateDone	float
dueDate	float
fecha_cambio	datetime

Figura 8.3. Atributos de la tabla estados. Fuente: Elaboración propia en base a los datos de la empresa

Como se aprecia en la Figura 8.3, esta tabla contempla información adicional relevante, como el tiempo de arranque y el tiempo de ciclo, expresados en horas. Debido a que uno de los aspectos más críticos es el cumplimiento de las horas dedicadas, resulta esencial contar con la dedicación promedio de cada trabajador a sus issues asignados. Además, es fundamental registrar los tiempos en los que un issue transita por diferentes estados relevantes durante su ciclo de vida. Por consiguiente, se propone el siguiente cálculo para los tiempos de arranque y ciclo:

1. *Ciclo de tiempo individual*: Es necesario saber el tiempo promedio en la que un desarrollador se demora en pasar sus issues de estado In Progress a estado Done, por consiguiente, se calcula como sigue:

$$\bar{x} = \frac{\sum_{i=1}^n Fecha(Done)_i - Fecha(In Progress)_i}{n}$$

En donde la sumatoria indica la suma de los tiempos de ciclo individuales de todos los issues, y n es el número total de issues por cada persona.

Para calcular el tiempo de ciclo, se presentan varios casos especiales. El primero es que no todos los issues pasan por el estado "In Progress". De hecho, es común que algunos vayan directamente de "Open" a "Done". Esto sucede por una razón: los desarrolladores a veces olvidan actualizar el estado a "In Progress" y marcan los issues directamente como "Done". De manera

similar, el estado "Done" también puede perderse por la misma razón mencionada para "In Progress". Debido a esto, no siempre se registran todos los estados por los que pasa un issue.

Cuando se presentan estos casos, el cálculo se realiza considerando el estado inmediatamente anterior o posterior. Esto significa que, si se pierde el estado "In Progress", se utiliza "Open" en su lugar. Si se pierde el estado "Done", se utiliza "QA" en su lugar.

Para ilustrar el concepto de tiempo de ciclo, se considera el siguiente ejemplo: si un issue tarda 4 horas en avanzar de "In Progress" a "Done", indica que el desarrollador empleó 4 horas para completar ese issue. No obstante, este es un escenario ideal; existen casos en los que el desarrollador podría requerir más de un día para finalizar un issue. En tales situaciones, es necesario descontar el tiempo correspondiente a las horas fuera del horario laboral, así como los fines de semana y días festivos. Estas horas no se incluyen en el cálculo del tiempo de ciclo.

2. *Tiempo de arranque individual*: Es necesario saber el tiempo promedio en la que un desarrollador se demora en pasar sus issues de estado Open a estado In Progress, por consiguiente, se calcula como sigue:

$$\bar{x} = \frac{\sum_{i=1}^n Fecha(\text{In Progress})_i - Fecha(\text{Open})_i}{n}$$

En donde la sumatoria indica la suma de los tiempos de arranque individuales de todos los issues, y n es el número total de issues por cada persona. Este tiempo promedio está medido en días.

El tiempo de arranque, similar al tiempo de ciclo, también puede experimentar pérdidas de estado debido a la razón mencionada anteriormente. El estado "Open" no se pierde, ya que los issues comienzan en ese estado por defecto. Sin embargo, el estado "In Progress" sí puede perderse. En estos casos, se utiliza "Done" o "QA" en su lugar, dependiendo de cuál sea el siguiente estado registrado.

8.4. Visualización de los datos

8.4.1. Definición de plataforma

Para seleccionar la mejor opción para desarrollar los paneles de control, la gerencia de NeoLab solicitó un análisis comparativo de plataformas que cumplieran con los siguientes criterios: ser gratuitas, de código abierto y ofrecer una amplia variedad de gráficos y tablas para

la visualización de datos empresariales. La investigación identificó dos opciones destacadas que cumplieran con estos requisitos: Metabase y Apache Superset.

Metabase es una plataforma de código abierto que se caracteriza por su configuración sencilla y experiencia interactiva. Ofrece la posibilidad de conectarse a una amplia gama de fuentes de datos, aunque tiene algunas limitaciones en cuanto a la variedad de gráficos disponibles. En contraste, Apache Superset, también de código abierto, destaca por su extensa gama de opciones de visualización gráfica. Sin embargo, su configuración puede ser más compleja y su capacidad de conexión a fuentes de datos es más restringida en comparación con Metabase.

NeoLab ya contaba con experiencia en el uso de Metabase, ya que la había implementado en un proyecto previo para desarrollar un panel de control para una empresa. La familiaridad con la plataforma y el hecho de que ya tenían un servidor en funcionamiento inclinaron la balanza a favor de Metabase, a pesar de sus limitaciones en cuanto a la variedad de gráficos. En comparación, Apache Superset, aunque ofrece una gama más amplia de opciones gráficas, presentaba una curva de aprendizaje más pronunciada y no se conectaba directamente a MariaDB sin modificaciones adicionales. Dado que los datos de la empresa se encuentran en MariaDB, Metabase resulta ser la opción más práctica para NeoLab.

8.4.2. Definición de paneles de control

Se eligieron tres paneles de control específicos—el panel de monitoreo de horas por proyecto, el panel de seguimiento para desarrolladores y el panel de monitoreo gerencial—basándose en necesidades identificadas y en la aprobación de la gerencia. La selección de estos paneles se realizó tras un proceso que incluyó reuniones con la gerencia y consultas con los desarrolladores para asegurar que las herramientas cumplieran con los requisitos esenciales para la gestión de proyectos.

Panel de monitoreo de horas por proyecto: Este panel es crucial para rastrear la dedicación de tiempo a cada proyecto y verificar si se están respetando las horas presupuestadas. Conocer las horas utilizadas y las horas restantes ayuda a la gerencia a tomar decisiones informadas sobre la distribución del personal y a garantizar que los recursos se utilicen de manera eficiente. La gerencia y los desarrolladores coincidieron en la necesidad de esta visibilidad para evitar desviaciones significativas en el tiempo.

Panel de seguimiento para desarrolladores: La necesidad de tener una visión clara de las tareas asignadas a cada desarrollador, junto con sus fechas de entrega y estado, fue identificada como fundamental para mejorar la supervisión del progreso del equipo. Este panel facilita la gestión del trabajo diario y permite a la gerencia y a los desarrolladores mantenerse al tanto del estado de las tareas individuales, contribuyendo a una coordinación más efectiva.

Panel de monitoreo gerencial: La identificación y gestión de retrasos en los proyectos es esencial, ya que responde directamente al objetivo general del proyecto. Este panel proporciona una visión detallada de los retrasos y permite a la gerencia tomar acciones concretas para abordar issues específicos que están causando demoras. La necesidad de este panel surgió durante las reuniones con la gerencia y las consultas con los desarrolladores, quienes señalaron que una herramienta de monitoreo de retrasos ayudaría a gestionar mejor los problemas y a mantener los proyectos en curso.

8.4.3. Panel de monitoreo de horas por proyecto:

Uno de los paneles de control más importantes para NeoLab es aquel que muestra la dedicación de horas para cada proyecto. En consecuencia, se propone la siguiente visualización:

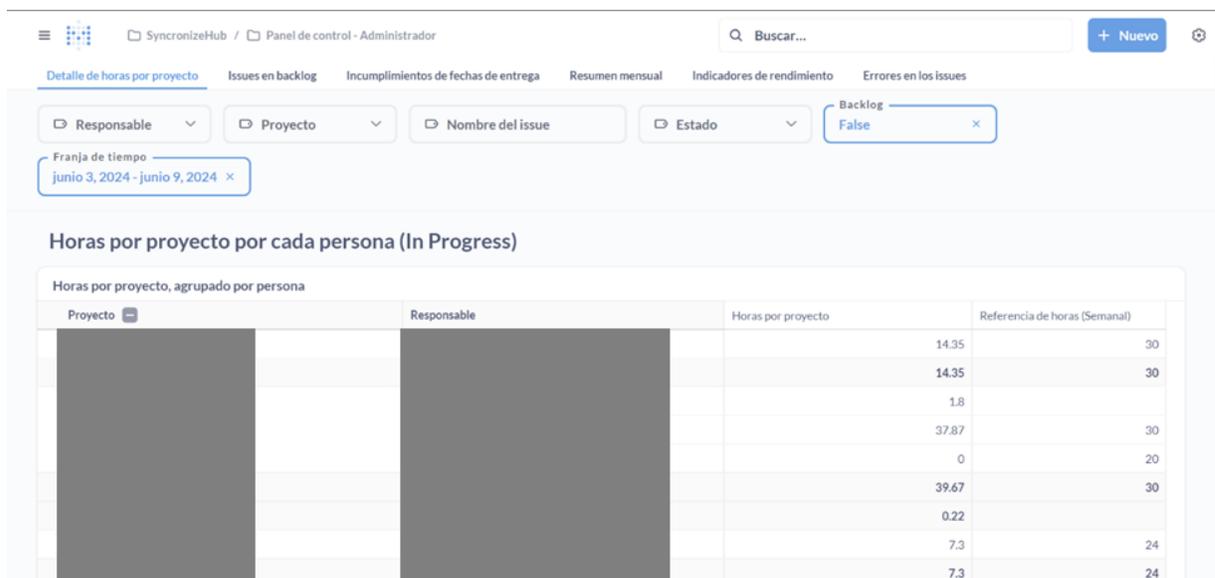


Figura 8.4. Visualización de horas por proyecto. Fuente: Elaboración propia en base a los datos en Metabase.

En el formato propuesto, se identifica cada proyecto individualmente y se enumera el personal asignado a ese proyecto, junto con las horas que cada uno ha dedicado. Esto proporciona una visión completa de la contribución de cada miembro del equipo al proyecto en cuestión. Además, se incluye una columna para las horas totales dedicadas al proyecto, que muestra la suma de las horas de todos los empleados involucrados.

Para alimentar las tablas correspondientes, se requiere utilizar el tiempo de ciclo individual medido en horas para cada issue, sin dividir la suma total de estos tiempos por el número de issues. Por ejemplo, supóngase que el Empleado 1 tiene los siguientes issues con los siguientes tiempos de ciclo:

Issue	Tiempo de ciclo (en horas)
Issue 1	5
Issue 2	12
Issue 3	7

Tabla 2. Ejemplo de cantidad de horas por proyecto para el Empleado 1. Fuente: Elaboración propia en base a los datos de NeoLab.

Al examinar la participación del Empleado 1 en el Proyecto 1, si ha invertido 5 horas en el Issue 1, 12 horas en el Issue 2 y 7 horas en el Issue 3, su aporte total sería de 24 horas. Esta cifra refleja el total de horas dedicadas por ese empleado al proyecto.

Es fundamental que este panel de control sea interactivo, permitiendo a los usuarios realizar las siguientes acciones:

1. Visualizar las horas dedicadas a cada proyecto al seleccionar dicho proyecto.
2. Acceder a la información detallada de las horas dedicadas en una franja de tiempo, pudiendo seleccionar los días que se quiere revisar.

Este enfoque interactivo brinda a los usuarios la capacidad de explorar y analizar los datos de una manera mucho más organizada.

8.4.4. Panel de seguimiento para desarrolladores:

Este panel se despliega específicamente los lunes y viernes, coincidiendo con las reuniones semanales con la gerencia. Su propósito es proporcionar una visión clara y detallada de las tareas pendientes y el estado de los proyectos para facilitar la planificación y el seguimiento del progreso.

El lunes, se ofrece una vista detallada que enumera todas las tareas pendientes de cada desarrollador en Space, clasificadas por proyecto. Cada tarea se presenta con su nombre correspondiente, su estado actual, la fecha esperada de finalización (Due Date Done) y la fecha de vencimiento (Due Date). Esta disposición brinda a los desarrolladores una comprensión clara y completa de las tareas que deben abordar durante la semana. Para mayor detalle, se adjunta la Figura 8.5:

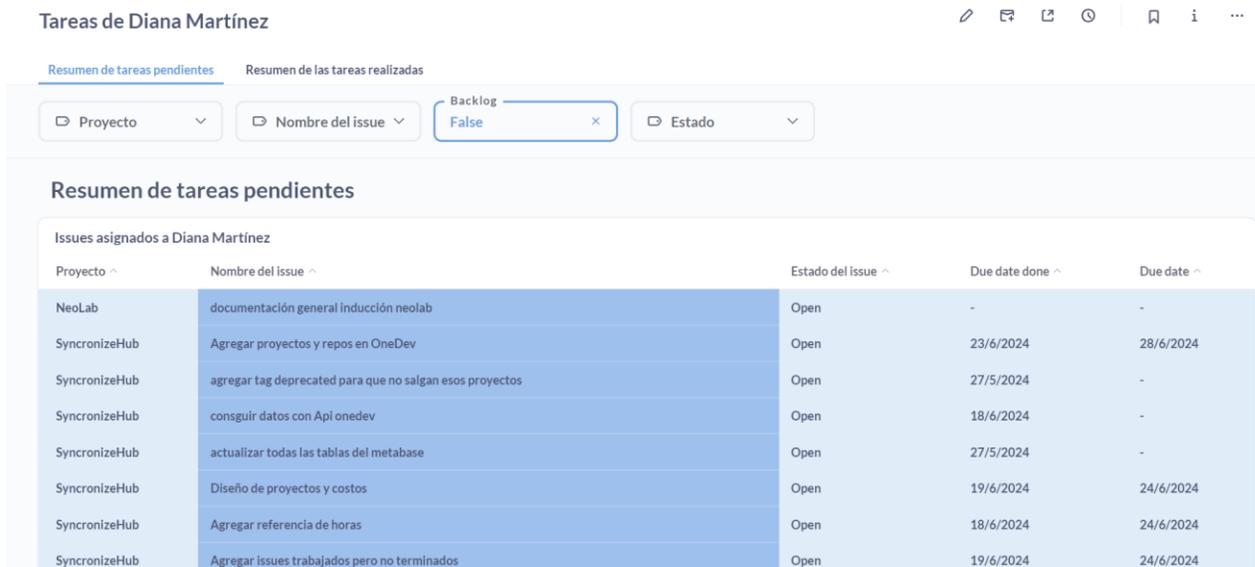


Figura 8.5. Visualización de panel de seguimiento para desarrolladores. Fuente: Elaboración propia en base a los datos en Metabase.

Además, se ofrece la capacidad de filtrar la información por estado o proyecto, lo que brinda a los usuarios la flexibilidad de mostrar solamente aquellos issues donde quieren darle más énfasis o consideren más importantes.

Los viernes se presentan cinco tablas adicionales que proporcionan información detallada sobre el progreso y estado de las tareas (Ver Anexo D):

1. Tareas en curso: Muestra las tareas que están en estado "Open" o "In Progress", pero en las cuales que se ha trabajado durante la semana (es decir, que han estado en "In Progress" en algún momento de la semana).
2. QA realizados: Lista los QA efectuados por cada persona durante la semana.
3. Tareas aceptadas: Enumera las tareas que han alcanzado el estado "Accepted" o posterior durante la semana.
4. Tareas pendientes de QA: Presenta las tareas propias que están a la espera de QA.
5. Tareas atrasadas: Detalla las tareas que han superado su fecha límite de entrega.

Se eligieron estos cinco puntos después de conversar con la gerencia y los desarrolladores. El objetivo era que el informe del viernes reflejara todo el trabajo realizado durante la semana, no solo las tareas completadas. De esta manera, se logra una visión más completa del progreso, incluyendo las tareas en proceso y los desafíos que surgen.

8.4.5. Panel de monitoreo gerencial:

Este panel de control complementa al panel de control utilizado para medir las horas por proyecto, ya que se enfoca en detectar posibles errores humanos, como la falta de actualización

de estados, que podrían afectar los resultados del panel principal. En este sentido, una de sus tareas es identificar los issues que presentan retrasos en sus entregas. Para ello, proporciona una vista que muestra todos los issues con incumplimientos de fechas de entrega (Due Date Done y Due Date), lo que permite detectar issues abandonados u olvidados que no han sido actualizados en Space.

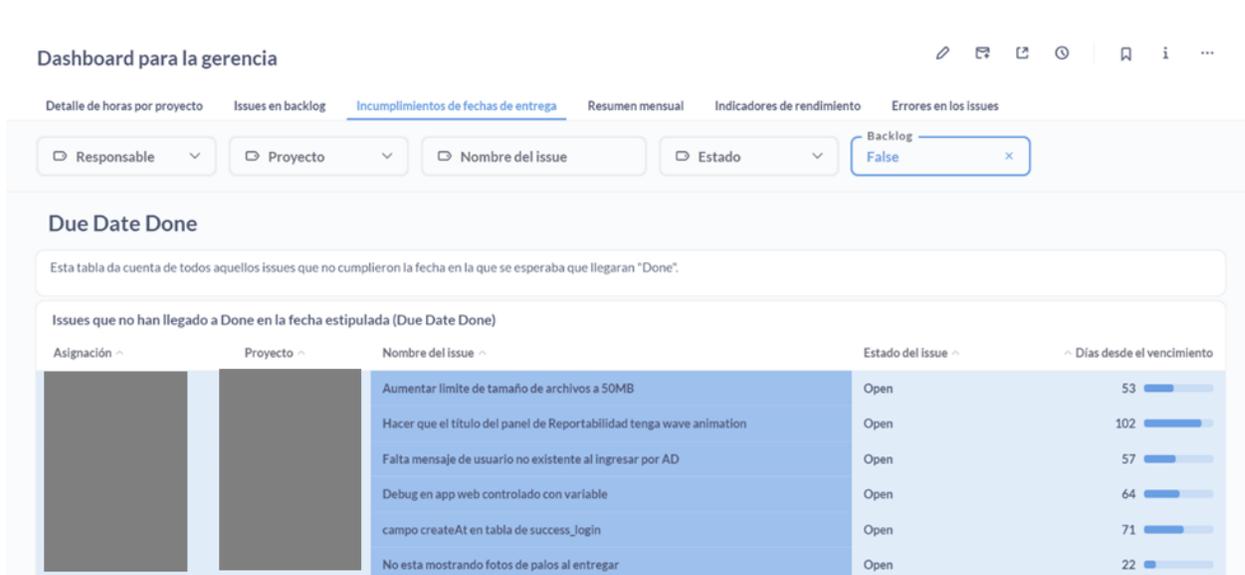


Figura 8.6, Visualización de panel de monitoreo gerencial. Fuente: Elaboración propia en base a los datos en Metabase.

Paralelamente, este panel de control muestra todos aquellos issues que tengan particularidades, tales como:

- Issues sin Due Date.
- Issues sin Due Date Done.

A continuación, se presenta una visualización en Metabase de estas particularidades:

Issues sin fecha "Due date "

Proyecto	Nombre del Issue	Estado del Issue	Due date	Responsable actual
	Probar plugin para optimizar imágenes	Open	None	
	buscar herramienta que permita saber los archivos usados y los que no	Open	None	
	Actualizacion Manual de Usuario(Infraestructura, Actividades y FeedBack)	Waiting for QA	None	
	Manual de usuario	Waiting for QA	None	
	Actualizar Manual de usuario(Reportabilidad y Educacion)	Open	None	
	investigar link de notificacion	Open	None	
	verificar mensajes de error en formulario de inscripción	Done	None	
	Agregar botones de exportación para queries	Open	None	
	Eliminar columna Created at de lista de tournaments en el admin	Open	None	

Filas 1-9 de 89

Figura 8.7. Visualización de tareas sin Due Date. Fuente: Elaboración propia en base a los datos en Metabase.

8.4.6. Utilidad de cada uno de los paneles de control

Tras la explicación del diseño de cada uno de los paneles de control, se procede a detallar el funcionamiento de cada panel y cómo estos contribuirán al cumplimiento del objetivo general propuesto en esta memoria.

Iniciando con el panel de monitoreo de horas por proyecto, se establece que este será visualizado exclusivamente por la gerencia todos los miércoles, que es el día en el cual la gerencia tiene más disponibilidad de tiempo libre. En estas reuniones semanales, se revisará la cantidad de horas dedicadas por cada empleado durante la semana anterior. Este proceso permitirá identificar a las personas que han invertido una cantidad de horas superior a la estimada para un proyecto específico, así como a aquellas que han dedicado menos tiempo del previsto a ciertas tareas. La identificación de estos patrones facilitará la reasignación estratégica del personal. Aquellos empleados que demuestren mayor eficiencia en sus tareas podrán ser reubicados para apoyar proyectos que estén experimentando retrasos. Esta estrategia contribuye a la implementación de medidas rápidas para abordar los proyectos que muestran signos de demora, especialmente en casos donde se están utilizando más horas de las originalmente planificadas.

El segundo panel es el de seguimiento para desarrolladores. Este se revisará dos veces por semana, los lunes y viernes, en reuniones con la gerencia. En estas reuniones, cada trabajador explicará en qué tareas está trabajando. Lo más importante de este panel es que mostrará claramente qué tareas están atrasadas. Esto permitirá a la gerencia identificar rápidamente los problemas y pedir explicaciones a los desarrolladores sobre por qué hay retrasos. Esta será una forma de mantener a todos informados y buscar soluciones a tiempo.

El tercer panel complementa a los dos anteriores. Tendrá gráficos que mostrará qué tan eficiente es cada empleado y cuánto tiempo llevan atrasadas sus tareas. Este panel se usará junto con el primero en las reuniones de los miércoles. Su objetivo será ayudar a la gerencia a decidir qué persona es la mejor opción para ayudar en otro proyecto y en qué tarea específica. Básicamente, proporcionará información extra para tomar mejores decisiones sobre cómo distribuir el trabajo.

Además, este último panel cumple una función crucial al identificar las tareas que carecen de fechas de Due Date y de Due Date Done. Esta información es esencial para determinar con precisión qué tareas están atrasadas, por lo que es crucial identificar los issues que no poseen estos campos. Para abordar esta problemática, se implementarán varias medidas clave. Primero, dado que NeoLab cuenta con 15 empleados y, por ende, es posible llevar un control, se designará a una persona específica para controlar que todas las tareas nuevas incluyan las fechas de Due Date y Due Date Done. Esta persona será responsable de verificar la correcta imputación de estos campos en cada tarea. Adicionalmente, se implementará un sistema de alertas automáticas. Utilizando Apache NiFi, se procesarán los datos recopilados para generar correos automáticos. Este sistema enviará correos electrónicos a los responsables y encargados de proyecto cuando una tarea se cree sin las fechas de Due Date y Due Date Done. De esta forma, se asegura que los responsables estén informados y puedan tomar las acciones necesarias para completar la información faltante y mantener el control sobre el progreso de las tareas.

8.4.7. Implementación de la solución

La implementación de la solución inició con la configuración de las herramientas esenciales. Primero, la gerencia se encargó de establecer un servidor Apache NiFi para la extracción y transformación de datos. Simultáneamente, la gerencia también configuró un servidor MariaDB para el almacenamiento de los datos procesados.

El siguiente paso fue la creación de scripts en Python para la extracción de datos de Space, un proceso explicado detalladamente en la etapa de recolección de datos. Estos scripts se diseñaron para capturar toda la información relevante de los issues, incluyendo sus estados, fechas de creación y modificación, asignaciones y comentarios. Una vez que los scripts de extracción funcionaron correctamente, se configuró el flujo de trabajo en Apache NiFi. Este flujo se programó para ejecutarse cuatro veces al día, según lo acordado con la gerencia. El proceso automatizado extrae los datos de Space, los procesa y los carga en las tablas correspondientes de MariaDB.

Con los datos disponibles en MariaDB, se procedió a la creación de las consultas SQL necesarias para alimentar los paneles de control. Las consultas se estructuraron de la siguiente manera:

1. Panel de monitoreo de horas por proyecto:
 - Consulta SQL para el cálculo de horas.

- Consulta SQL para desplegar las horas por proyecto y por empleado.

2. Panel de seguimiento de desarrolladores:

- Consulta SQL para las tareas en curso (tareas en progreso, pero no terminadas).
- Consulta SQL para los QA realizados.
- Consulta SQL para las tareas aceptadas por QA.
- Consultas SQL para las tareas pendientes de revisión.
- Consulta SQL para las tareas atrasadas.
- Consulta SQL para las tareas pendientes.

En esta etapa, es importante mencionar que estas seis consultas SQL se realizaron para cada trabajador. Es decir, se repitieron tantas veces como trabajadores hay en NeoLab.

3. Panel de monitoreo gerencial:

- Consulta SQL para mostrar los issues sin Due Date y/o Due Date Done.
- Consulta SQL para crear un gráfico de los tiempos de arranque y tiempos de ciclo.

El desarrollo de los paneles de control en Metabase constituyó la siguiente fase de la implementación. Se crearon visualizaciones para cada uno de los tres paneles: el de monitoreo de horas por proyecto, el de seguimiento para desarrolladores y el de monitoreo gerencial. Cada panel se diseñó con un enfoque en la usabilidad y la claridad de la información presentada, permitiendo filtros y opciones de personalización según las necesidades de los usuarios.

Los desarrolladores tienen acceso a su panel a través de un correo electrónico y una contraseña proporcionados. Además, para garantizar una gestión eficiente y mantener la información actualizada, se revisan conjuntamente los paneles los lunes y los viernes. Esta revisión periódica permite ajustar y verificar la información, asegurando que todos los datos relevantes estén correctamente reflejados y accesibles para los desarrolladores y la gerencia.

Ahora bien, para asegurar la precisión de los datos y la utilidad de los paneles, se llevó a cabo una fase de pruebas con un grupo selecto de usuarios, incluyendo desarrolladores y miembros de la gerencia. Sus comentarios se utilizaron para refinar las visualizaciones y ajustar los cálculos según fuera necesario. Esta fase iterativa fue crucial para garantizar que los paneles cumplieran con las necesidades de todos los implicados. Los desarrolladores aportaron dos sugerencias particularmente valiosas. Primero, propusieron dividir sus tareas en dos secciones: una para visualizar las pendientes y otra para las ya completadas. Esto facilita entender claramente cuáles tareas ya se completaron y cuáles siguen pendientes. Segundo, sugirieron añadir un filtro para buscar tareas por su "Estado". Esta función resulta especialmente útil cuando manejan numerosas tareas, ya que les permite localizar rápidamente las que se encuentran en una etapa específica, como "In Progress" o "Done".

8.5. Gestión del cambio

Crear un plan de gestión del cambio utilizando el modelo ADKAR es fundamental para lograr una transición exitosa en la implementación de los paneles de control en NeoLab. El modelo ADKAR se compone de cinco elementos fundamentales: conciencia, deseo, conocimiento, habilidad y refuerzo. Cada uno de estos elementos debe abordarse cuidadosamente para lograr una buena adopción del cambio.

El plan de gestión del cambio se desarrollará a lo largo de 8 semanas, siguiendo el modelo ADKAR. Durante las primeras dos semanas, la empresa se centrará en crear conciencia. El Project manager de la empresa liderará una reunión general de lanzamiento el primer día, explicando por qué es necesario implementar los nuevos paneles de control y cómo estos mejorarán la gestión de tareas en NeoLab. Paralelamente, se dispondrá toda la información importante en el chat grupal de Microsoft Teams de la empresa, ya que es el canal de comunicación más importante de la empresa.

La semana 3 se enfocará en fomentar el deseo de participar en el cambio entre los empleados de NeoLab. Para lograr este objetivo, se implementarán varias estrategias clave. En primer lugar, se presentarán ejemplos concretos que demuestren cómo los nuevos paneles de control mejorarán el trabajo diario y resolverán problemas actuales. Esto ayudará a aumentar la motivación del equipo al visualizar los beneficios tangibles. Se hará énfasis en que los paneles no son una herramienta de control, sino un recurso valioso para optimizar el trabajo. Por ejemplo, se explicará cómo pueden ayudar a identificar situaciones de sobrecarga laboral, permitiendo ofrecer apoyo oportuno a quienes lo necesiten. Este enfoque no solo aliviará el exceso de trabajo, sino que también fomentará un ambiente laboral más equilibrado y colaborativo. Adicionalmente, se introducirá un sistema de reconocimiento para aquellos empleados que adopten y utilicen eficazmente los nuevos paneles. Esta iniciativa servirá como un incentivo adicional para promover la participación en el cambio.

El gerente de NeoLab tiene la responsabilidad clave de seleccionar estratégicamente a las personas que se encargarán de generar el deseo de participar en el cambio. Esta selección es crítica para el éxito de la implementación de los nuevos paneles de control. Los individuos elegidos deben cumplir con un criterio esencial: ser cercanos a los trabajadores. Esta cercanía implica que sean personas que:

1. Gozan de la confianza y el respeto de sus compañeros y entienden las preocupaciones y necesidades diarias del equipo.
2. Pueden comunicarse de manera efectiva y empática con diferentes grupos dentro de la organización.
3. Tienen la capacidad de influir positivamente en sus colegas.

Al escoger líderes que cumplan con estas características, el gerente asegura que el mensaje sobre los beneficios de los nuevos paneles sea transmitido de manera convincente y auténtica. Estos líderes podrán abordar las inquietudes de los trabajadores desde una posición de

entendimiento mutuo, lo que aumentará significativamente las posibilidades de generar un deseo genuino de participar en el cambio.

Durante las semanas 4 y 5, NeoLab se enfocará en impartir el conocimiento necesario para la implementación exitosa de los nuevos paneles de control. Este proceso será liderado por las personas seleccionadas por el gerente para liderar el cambio. El equipo organizará al menos dos sesiones de capacitación para todos los empleados, diseñadas para abordar todos los aspectos relevantes del nuevo sistema. Paralelamente a estas capacitaciones, se elaborarán dos manuales: uno sobre cómo actualizar correctamente los issues en la plataforma Space, y otro que explicará cómo utilizar correctamente Metabase.

En las semanas 6 y 7, el enfoque se centrará en el desarrollo de habilidades prácticas. El equipo encargado del cambio, consciente de la importancia de un apoyo continuo, designará a una persona específica para dar soporte técnico a los desarrolladores. Esta persona será el punto de contacto principal para resolver dudas y problemas que puedan surgir durante el uso diario de los paneles.

A partir de la semana 8, se iniciará una fase crucial de refuerzo y mejora. Se establecerá un sistema de evaluaciones periódicas para medir el uso y la efectividad de los paneles. Este proceso no solo consistirá en recopilar datos, sino en comprender cómo los paneles están influyendo en el trabajo diario y en identificar áreas de mejora. Para asegurar el uso continuo de los paneles, el gerente general tomará un papel activo en la supervisión. Se organizarán reuniones regulares en las que cada desarrollador deberá presentar las tareas que está realizando, de modo de poder llevar un seguimiento de sus tareas.

Para asegurar una implementación exitosa de los paneles de control en NeoLab utilizando el modelo ADKAR, se definirán indicadores clave de rendimiento específicos para cada fase del plan de gestión del cambio, desarrollado a lo largo de 8 semanas:

1. Conciencia (Semanas 1 y 2): Asistencia a la reunión de lanzamiento. Tasa de lectura de las comunicaciones en Microsoft Teams.
2. Deseo (Semana 3): Participación en las sesiones informativas.
3. Conocimiento (Semanas 4 y 5): Asistencia a las sesiones de capacitación. Resultados de una prueba escrita sobre el entendimiento de los manuales.
4. Habilidad (Semanas 6 y 7): Tasa de adopción de paneles de control. Confianza en las habilidades mediante encuestas.
5. Refuerzo (Semana 8 en adelante): Satisfacción del usuario.

9. Discusión

El proyecto ha demostrado ser efectivo en gran medida, aunque existen ciertos aspectos que podrían optimizarse para aumentar su impacto. Actualmente, se están implementando mejoras adicionales, lo que marca el inicio de una Fase 2 del proyecto. Esta nueva fase incluye mejoras cruciales que no fueron contempladas en la memoria original, pero que son fundamentales para abordar las problemáticas identificadas. A continuación, se presenta un análisis detallado de estas mejoras.

En primer lugar, se decidió migrar de la plataforma Space a OneDev para la gestión de tareas. Aunque Space era la plataforma original, su cambio de enfoque (realizado por el mismo Space en mayo de 2024) hacia la gestión exclusiva de código dejó de ser adecuado para las necesidades de NeoLab. OneDev, con una interfaz similar a Space, ofrece mayor flexibilidad y mayor control de las tareas. Tras la migración a OneDev, se implementaron varias mejoras clave:

Incorporación de reglas de estado: OneDev permite establecer reglas para el cambio de estado de los issues, lo que garantiza que todos los issues sigan un flujo de trabajo predeterminado. Por ejemplo, un issue debe comenzar en el estado “Open” y pasar por estados intermedios antes de llegar a “Finished”. Esta función soluciona el problema de estados saltados o cambiados incorrectamente por los desarrolladores.

Tiempo estimado por issue: Ahora, al crear un issue, se requiere ingresar el tiempo estimado que tomará completarlo. Por ejemplo, si se trata de crear un botón de inicio de sesión, además de todos los detalles habituales, el desarrollador debe especificar cuántas horas estima que tomará esta tarea. Esto mejora la planificación y el seguimiento del tiempo dedicado a cada tarea.

Tags de tipo y urgencia: Se introdujeron nuevos tags para clasificar los issues en NeoLab según su tipo y nivel de urgencia, lo que facilita una mejor organización y priorización de las tareas. Los tipos de tags implementados son los siguientes:

1. *Task:* Este es el tag por defecto para un issue en NeoLab. Si no se especifica otro tag, se considerará como una tarea estándar. Incluye actividades como desarrollo de código, diseño de mockups, entre otras tareas comunes del ciclo de vida de un proyecto.
2. *Reunión:* Este tag indica que el issue está relacionado con una reunión. En el campo "estimation", se debe registrar el tiempo que efectivamente tomó la reunión, reflejando con precisión el tiempo invertido.
3. *Capacitación:* Este tag se asigna a issues que implican una capacitación o curso.
4. *Administrativo:* Este tag se utiliza para tareas administrativas, como enviar correos, gestionar firmas electrónicas, o solicitar permisos. Estas tareas suelen ser breves y no

directamente relacionadas con el desarrollo del proyecto, pero son necesarias para su correcto funcionamiento.

5. *Mejora Interna*: Este tag se asigna a issues que corresponden a mejoras dentro del proyecto. Aunque no son de máxima prioridad, estas tareas son importantes para el progreso y la calidad del proyecto.
6. *Bug*: Este tag identifica issues relacionados con bugs o errores detectados en el sistema. Dado que los bugs pueden impactar negativamente en el proyecto, es crucial gestionarlos con prioridad adecuada.
7. *Urgente*: Este tag adicional se incorpora para marcar issues que requieren atención inmediata, independientemente de su tipo. La asignación de este tag garantiza que estos issues se prioricen y se resuelvan rápidamente, minimizando el impacto en el proyecto.

Estas mejoras abordan aspectos no considerados en la fase inicial del proyecto, como la variabilidad en los tiempos de dedicación y la urgencia de las tareas. Issues de tipo “Administrativo” tienden a tomar menos tiempo que un “Bug”, que puede demorar en promedio cuatro horas. Asimismo, los issues etiquetados como “Urgente” suelen resolverse más rápidamente que los demás. La falta de estas etiquetas en la fase inicial dificultó un análisis preciso del tiempo de resolución de las tareas urgentes.

En cuanto al tag “Reunión”, se implementó para contabilizar adecuadamente las tareas relacionadas con reuniones o llamadas telefónicas, que también ocupan tiempo en los proyectos. El proceso es simple: el desarrollador ingresa la tarea en OneDev, le asigna el tag “Reunión” y el tiempo que se registre se considera como el tiempo real empleado, no solo como una estimación. Por ejemplo, si se estima una hora para una reunión, esa hora se contabiliza directamente como tiempo trabajado.

Es importante destacar las limitaciones de este proyecto, las cuales van más allá de la mera elección de plataforma, sea OneDev o Space. La eficacia de los paneles implementados depende fundamentalmente de la disposición de los desarrolladores para actualizar los estados en el sistema de manera precisa y oportuna. Independientemente de la complejidad de las reglas de cambios de estado o la diversidad de tags incorporados, si el equipo no utiliza el sistema de forma adecuada, su utilidad se ve severamente comprometida. En retrospectiva, se evidencia que la gestión del cambio debió haber recibido mayor énfasis desde el inicio del proyecto, en lugar de ser considerada como la fase final de implementación. Idealmente, la planificación de la gestión del cambio debería haber comenzado simultáneamente con la fase de recolección de datos. Para futuros proyectos de esta naturaleza, se recomienda encarecidamente priorizar la gestión del cambio desde las etapas iniciales. Además, un factor crucial a considerar es la curva de aprendizaje asociada con la implementación de estos paneles. El proceso de familiarización y adaptación a la nueva herramienta requiere tiempo, un recurso particularmente escaso en entornos de desarrollo dinámicos. Es importante reconocer que este tiempo de aprendizaje no debe comprometer la productividad ni los plazos de entrega de los proyectos en curso. En consecuencia, se identifica como trabajo futuro prioritario el diseño de una estrategia de capacitación que permita al equipo de desarrollo comprender y utilizar efectivamente los paneles de control.

10. Conclusiones

El objetivo principal del proyecto era diseñar y desarrollar paneles de control que permitieran a la gerencia de NeoLab detectar de forma anticipada los issues con retrasos, posibilitando la toma de acción en un plazo inferior a una semana. Este objetivo se ha cumplido considerablemente, ya que se ha logrado identificar y gestionar tareas atrasadas a través del panel de horas por proyecto, incluso antes de que ocurran.

Desde el 9 de febrero de 2024, se han registrado 458 issues en la empresa. De estos, 369 (80.57%) tienen una fecha de vencimiento asignada, lo que permite un seguimiento preciso de cada uno de estos issues. Esta mejora en la gestión de tareas es notable, considerando que antes del proyecto, NeoLab carecía de un sistema formal para detectar el estado de los issues y sus retrasos. El análisis de los datos revela que 106 issues (28.72% de los que tienen fecha de vencimiento) han experimentado un retraso de un día. Sin embargo, lo más destacable es que el 60.26% de estos issues atrasados se han gestionado dentro de los primeros 7 días de retraso. Esto demuestra la eficacia de los nuevos paneles en la identificación temprana y manejo oportuno de los retrasos.

Es importante señalar que 89 issues aún carecen de una fecha de vencimiento asignada, lo que representa un área de mejora para el futuro. No obstante, el progreso general es sustancial, especialmente al considerar que antes no existía un método sistemático para rastrear y manejar los retrasos. Aunque no es posible realizar una comparación numérica directa con la situación previa al proyecto, la implementación de estos paneles de control ha transformado claramente la capacidad de NeoLab para gestionar sus proyectos de manera más eficiente y proactiva.

Ahora bien, la flexibilidad del sistema desarrollado para NeoLab es una de sus fortalezas clave. En caso de que la empresa decidiera migrar de Space a otra plataforma de gestión de proyectos, el impacto en los paneles de control sería mínimo y potencialmente beneficioso. El cambio principalmente afectaría la fuente de datos, pero la estructura y funcionalidad de los paneles se mantendrían intactas. De hecho, esta transición podría simplificar el proceso de recopilación y análisis de datos.

En cuanto a los aspectos estratégicos de estas implementaciones, varios puntos fuertes han emergido. Uno de los más destacados es la capacidad de los paneles de control para identificar con precisión las cargas de trabajo desbalanceadas entre los diferentes equipos y trabajadores. Esta información ha permitido una redistribución de tareas más equitativa, asegurando que ningún miembro del equipo esté sobrecargado ni subutilizado. Este enfoque estratégico no solo optimiza el rendimiento de los proyectos al mejorar la eficiencia del equipo, sino que también proporciona una herramienta valiosa para medir el rendimiento individual. Con la capacidad de analizar detalladamente la contribución de cada persona dentro de la empresa, se pueden identificar aquellos que están trabajando menos de lo inicialmente contratado. Esto tiene

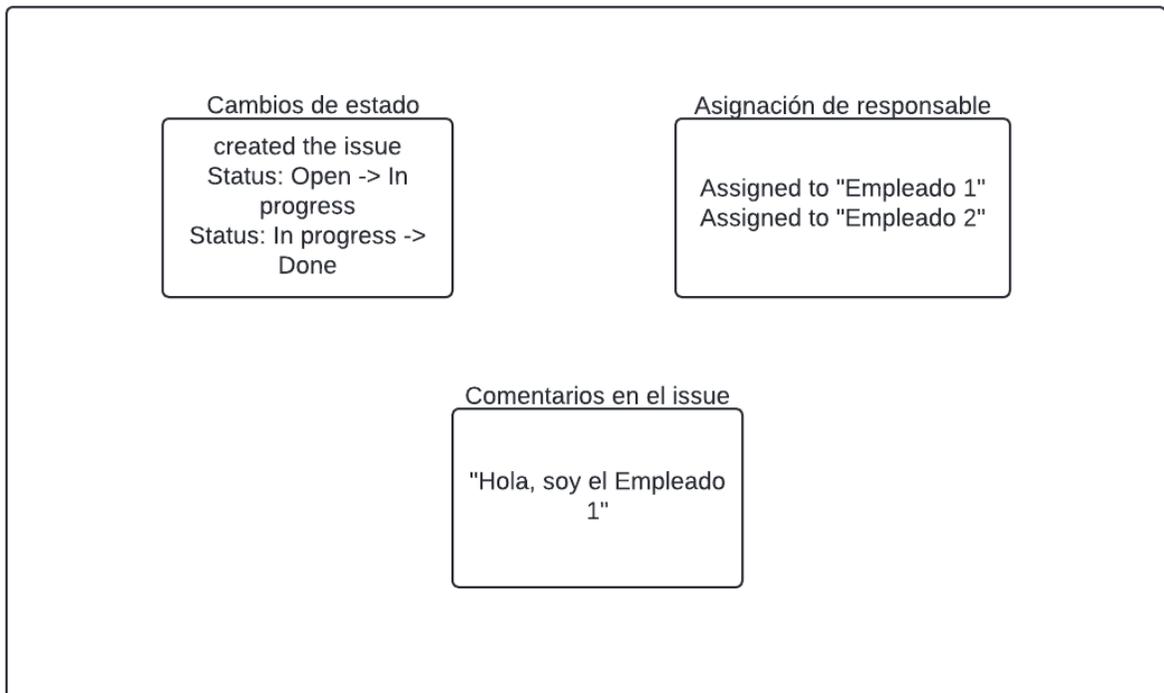
implicaciones directas en la gestión de recursos humanos, ya que permite tomar decisiones informadas respecto a la reestructuración de contratos, la reasignación de roles, y en casos necesarios, la terminación de contratos. La implementación de estos paneles, por tanto, no solo fortalece la eficiencia operativa, sino que también mejora la gestión del talento dentro de NeoLab.

Además, los paneles de control permiten a NeoLab tener una visión detallada del tiempo que se toma en proyectos similares, lo que resulta esencial para mejorar las negociaciones y la planificación de futuros contratos. Al analizar los datos históricos de horas trabajadas en proyectos pasados que sean similares, la empresa puede estimar con mayor precisión el tiempo necesario para completar tareas en nuevos proyectos. Esta información es invaluable para establecer contratos más realistas y competitivos, ya que permite ajustar las horas presupuestadas con base en la experiencia previa. Así, NeoLab puede ofrecer propuestas más ajustadas a la realidad, evitando tanto sobreestimaciones que puedan desalentar a los clientes, como subestimaciones que podrían resultar en sobre costos para la empresa. De esta manera, la capacidad de prever y controlar con precisión las horas dedicadas a cada proyecto mejora la eficiencia operativa y fortalece la posición de NeoLab en el mercado.

11. Bibliografía

1. Achá, V., & Bravo, C. (2009). Chile: Desarrollo endógeno con proyecciones externas. En Tigre P. & Silveira, F. (Eds.), *Desafíos y oportunidades de la industria del software en América Latina* (pp. 105-137). Mayol Ediciones.
2. Cano, J. L. (2007). *Business Intelligence: Competir con información*. ESADE Business School, Universidad Ramon Llull.
3. Duan, C. M. (2014). Design of big data processing system architecture based on Hadoop under the cloud computing. *Applied Mechanics and Materials*, 556-562, 6302. <https://www.scientific.net/AMM.556-562.6302>
4. Fernández, A. (2013). *Phyton 3 al descubierto* (2a ed.). Alfaomega Grupo Editor. https://books.google.cl/books?hl=es&lr=&id=f4BNDAAAQBAJ&oi=fnd&pg=PT3&dq=python&ots=Uche0cH4oD&sig=sWBBBT-vkV_rnHshIlgNeFHdbPM&redir_esc=y#v=onepage&q=python&f=false
5. Galli, B. J. (2018). Change Management Models: A Comparative Analysis and Concerns. *IEEE Engineering Management Review*, 46(3), 124-132. <https://doi.org/10.1109/EMR.2018.2866860>
6. Hiatt, J. M. (2006). ADKAR: A model for change in business, government and our community. Prosci Research.
7. Joyanes, L. (2019). *Inteligencia de negocios y analítica de datos* (1a. ed.). Alfaomega Grupo Editor. <https://books.google.ca/books?id=ifR5EAAAQBAJ&printsec=frontcover&hl=es#v=onepage&q&f=false>
8. Khan, A., Ghose, A., Dam, H. & Syed, A. (2023). *A Survey of Process-Oriented Data Science and Analytics for supporting Business Process Management*. <https://doi.org/10.48550/arXiv.2301.10398>
9. Khan, M. (2023). *Business Analytics*. Lovely Professional University. <https://www.lpude.in/SLMs/Master%20of%20Computer%20Applications/Sem%204/DEMG%20N801%20BUSINESS%20ANALYTICS.pdf>
10. Kenler, E., & Razzoli F. (2015). *MariaDB Essentials*. Packt Publishing Ltd. https://books.google.cl/books?hl=es&lr=&id=VPh_CwAAQBAJ&oi=fnd&pg=PP1&dq=mariadb&ots=v5O5nYo3_Z&sig=8bye6fsxK9CG74pR5XOJUXQIR_I&redir_esc=y#v=onepage&q=mariadb&f=false
11. Lauer, T. (2021). *Change management: Fundamentals and success factors*. Springer. <https://doi.org/10.1007/978-3-662-62187-5>
12. Sharda, R., Delen, D. & Turban, E. (2017). *Business Intelligence, Analytics, and Data Science: A Managerial Perspective* (4a ed.). Pearson.
13. Racka, K. (2022). Apache NiFi as a tool for stream processing of measurement data. *Economic Sciences / Nauki Ekonomiczne*, 35, 115. [https://doi.org/10.19251/ne/2022.35\(7\)](https://doi.org/10.19251/ne/2022.35(7))
14. Silva, L. (2017). Business Intelligence: un balance para su implementación. *InnovaG*, (3), 27-36. <https://revistas.pucp.edu.pe/index.php/innovag/article/view/19742>
15. Tunowski, R. (2015). *Business Intelligence in organization: Benefits, risks, and developments*. *Przedsiębiorczość i Zarządzanie (Entrepreneurship and Management)*, 16(2), 133-144. <https://doi.org/10.1515/eam-2015-0022>

Anexo C. Tipos de mensajes de "text".



Fuente: Elaboración propia en base a la información de NeoLab en Space.

Anexo D. Ejemplos de visualización de tareas aceptadas en el panel de seguimiento de desarrolladores

Tareas trabajadas durante la semana, pero no terminadas

Proyecto ^	Nombre del issue ^	Estado del issue ^	Tag ^	Due date done ^	Due date ^
SynchronizeHub	Cambiar print por log	Open	Task	5/7/2024	5/7/2024
SynchronizeHub	bug: filtro de tiempo cuando cruza meses	Open	Bug	12/7/2024	16/7/2024
SynchronizeHub	Cambiar "tareas aceptadas"	Open	Mejora	19/7/2024	24/7/2024
SynchronizeHub	carátula de Juan Manuel	Open	Task	17/7/2024	23/7/2024

Code Review / QAs realizados a otras personas

Code Review / QAs realizados a otras personas

Proyecto ^	Responsable del paso anterior ^	Tipo ^	Nombre del issue ^	Estado del issue ^	Due date done ^	Due date ^
ASENAV	Tomás Ianchina	QA	Probar deploy local	Finished	19/7/2024	19/7/2024
ASENAV	Matías Norambuena	QA	Creación de modelo de datos	Finished	17/7/2024	17/7/2024
BenditoResiduo	Tomás Ianchina	Code Review	Mantenedor de Rubros en Clientes	Waiting for QA	27/6/2024	28/6/2024
BenditoResiduo	Tomás Ianchina	Code Review	Implementar delete en el page.server.ts, formulario módulo noticias.	Accepted	5/7/2024	12/7/2024
BenditoResiduo	Tomás Ianchina	Code Review	Implementar delete en el page.server.ts, formulario módulo educación.	Accepted	8/7/2024	12/7/2024
BenditoResiduo	Tomás Ianchina	Code Review	Logs en Svelte	Accepted	19/7/2024	19/7/2024

Filas 1-6 de 18 < >

Tareas propias aceptadas

Tareas aceptadas de Diana Martínez

Proyecto ^	Nombre del issue ^	Estado del issue ^	Tag ^	Due date done ^	Due date ^
SynchronizeHub	Manual para nuevas personas - V2	Finished	Task	3/7/2024	8/7/2024
SynchronizeHub	Resumen mensual de issues	Finished	Task	11/7/2024	16/7/2024
SynchronizeHub	Revisar que se usa como llave para los usuarios	Finished	Bug	9/7/2024	9/7/2024
SynchronizeHub	Devolver issue a Open luego de 9 horas	Finished	Task	4/7/2024	10/7/2024

Tareas propias en Done / Code Review / Waiting for QA / QA

Tareas en revisión de Diana Martínez

Proyecto ^	Nombre del issue ^	Estado del issue ^	Tag ^	Due date done ^	Due date ^
SynchronizeHub	Agregar tag deprecated para que no salgan esos proyectos	Waiting for QA	Task	1/7/2024	24/7/2024
SynchronizeHub	URGENTE - Due date done y due date permiten null	Waiting for QA	Bug	19/7/2024	22/7/2024
SynchronizeHub	Agregar referencia de horas	Waiting for QA	Task	17/7/2024	23/7/2024
SynchronizeHub	Usar estimation para tag reuniones - in progress	Waiting for QA	Task	10/7/2024	15/7/2024
SynchronizeHub	Corrección de correo para issues	Waiting for QA	Bug	19/7/2024	26/7/2024

Tareas atrasadas

Tareas atrasadas de Diana Martínez

Proyecto ^	Nombre del issue ^	Estado del issue ^	Tag ^	Due date done ^	Due date ^
NeoLab	Considerar todos los involucrados para una reunión	Open	Task	19/7/2024	19/7/2024
SynchronizeHub	Cambiar "tareas aceptadas"	Open	Mejora	19/7/2024	24/7/2024
SynchronizeHub	carátula de Juan Manuel	Open	Task	17/7/2024	23/7/2024
SynchronizeHub	Cambiar print por log	Open	Task	5/7/2024	5/7/2024
SynchronizeHub	bug: filtro de tiempo cuando cruza meses	Open	Bug	12/7/2024	16/7/2024

Fuente: Elaboración propia en base a los datos en Metabase.