



**Optimización del Desarrollo Frontend: Actualización y Mejora del Storybook en SBS
Seguros Colombia**

Autor

Zaira Ximena Rey Rojas

Presentado para optar el título de: Ingeniero de Sistemas

Asesor

Jose Manuel Chauta Torres

Institución Universitaria Politécnico Grancolombiano

Facultad de Ingeniería y Diseño e Innovación

Programa Ingeniería de Sistemas

Bogotá D.C

2023



Tabla de Contenido

RESUMEN	5
INTRODUCCIÓN	7
CONTEXTO	9
OBJETIVOS	11
OBJETIVO PRINCIPAL	11
OBJETIVOS SECUNDARIOS.....	11
DESCRIPCIÓN DE LA PROBLEMÁTICA	13
SOLUCIONES ACTUALES	14
JUSTIFICACIÓN DE LA INVESTIGACIÓN	15
MARCO TEORICO.....	15
METODOLOGÍA.....	18
DESCRIPCIÓN DE ACTIVIDADES.....	19
IMPLEMENTACIÓN STORYBOOK OPTIMIZADO	26
DESCRIPCIÓN DEL PROCESO	26
RECOLECCIÓN DE REQUISITOS	27
REVISIÓN DETALLADA VERSION 5.3.17 STORYBOOK	29
EVALUACIÓN DE PROBLEMAS DE RENDIMIENTO Y POSIBLES ÁREAS DE MEJORA.....	31
INVESTIGACIÓN Y FAMILIARIZACIÓN CON LAS NUEVAS VERSIONES	32
IDENTIFICACIÓN Y ACTUALIZACIÓN DE COMPONENTES OBSOLETOS Y DEPENDENCIAS DESACTUALIZADAS	34
ESTUDIO DOCUMENTACIÓN DE VUE 3 EN COMPARACIÓN A VUE 2	44
INTERFAZ FRAMEWORK STORYBOOK.....	49
DIAGRAMA DE FLUJO STORYBOOK.....	53
CRONOGRAMA DE ACTIVIDADES	55
ENTREGABLES	56
CONCLUSIONES Y RECOMENDACIONES	56
REFERENCIAS.....	57

Tabla 1: Requerimientos	29
Tabla 2: Causas de desactualización.....	32
Tabla 3: Evaluación características storybook v7.6.....	34
Tabla 4: Mejora dependencias fullhuman/postcss-purgecss	35
Tabla 5: Mejora dependencias mathieustan/vue-datepicker	36
Tabla 6: Mejora dependencias storybook/addon-viewport	36
Tabla 7: Mejora dependencias storybook/addons	37
Tabla 8: Mejora dependencias storybook/theming	38
Tabla 9: Mejora dependencias storybook/vue	38
Tabla 10: Mejora dependencias bootstrap	39
Tabla 11: Mejora dependencias core.js	40
Tabla 12: Mejora dependencias vee-validate	41
Tabla 13: Mejora dependencias vue-i18n	41
Tabla 14: Mejora dependencias vue-select	42
Tabla 15: Dependencias de desarrollo	43
Tabla 16: Mejoras de Vue2 a Vue3	44

Tabla de ilustraciones

Ilustración 1: Diagrama EDT – Autoría propia	19
Ilustración 2: Selector de historias Storybook – Autoria propia.....	50
Ilustración 3: Canvas framework Storybook - Autoría propia.....	51
Ilustración 4: Barra de herramientas framework Storybook – Autoría propia	51
Ilustración 5: Panel de controles Storybook – Autoria Propia.....	52
Ilustración 6: Búsqueda y filtrado Storybook – Autoria Propia	53

Ilustración 7: Visualización Framework Storybook – Autoría Propia.....	53
Ilustración 8: Diagrama de flujo – Autoría propia.....	54
Ilustración 9: Diagrama de Gantt – Autoría propia	56

RESUMEN

En respuesta a la necesidad de mantener la eficiencia y coherencia en el desarrollo de software, este proyecto se enfoca en la actualización del Storybook en SBS Seguros, el cual se encontraba obsoleto debido a dependencias desactualizadas. La obsolescencia planteaba desafíos significativos para el desarrollo de interfaces de usuario, afectando la agilidad y calidad del proceso.

Por lo que en este documento encontrará la iniciativa que consistió en modernizar el Storybook mediante la identificación y sustitución de dependencias obsoletas. La metodología abarcó un análisis minucioso de las dependencias existentes y su reemplazo con versiones más recientes y compatibles. Obteniendo así resultados que destacan una mejora palpable en la eficiencia del desarrollo de interfaces. La eliminación de dependencias obsoletas no solo fortaleció la seguridad del sistema, sino que también proporcionó una base sólida para futuras actualizaciones. Este proyecto contribuye de manera significativa a la agilidad del desarrollo de software en SBS Seguros, posicionándola para adaptarse rápidamente a las demandas cambiantes de la industria.

PALABRAS CLAVES

Optimización, Obsolencia, Storybook, Actualización

ABSTRACT

In response to the need to maintain efficiency and coherence in software development, this project focuses on updating the Storybook at SBS Seguros, which had become obsolete due to outdated dependencies. Obsolescence presented significant challenges to the development of user interfaces, impacting the agility and quality of the process.

This document outlines the initiative to modernize the Storybook by identifying and replacing outdated dependencies. The methodology involved a meticulous analysis of existing dependencies and their replacement with more recent and compatible versions. The results demonstrate a tangible improvement in the efficiency of interface development. The removal of obsolete dependencies not only strengthened the system's security but also provided a solid foundation for future updates. This project significantly contributes to the agility of software development at SBS Seguros, positioning it to adapt quickly to the changing demands of the industry.

KEY WORDS

Optimization, Obsolescence, Storybook, Update

INTRODUCCIÓN

SBS Seguros Colombia, comprometida con la excelencia y la evolución continua, inicia un proceso de innovación crucial para fortalecer su presencia digital. Desde su fundación, SBS Seguros ha demostrado su dedicación a comprender las demandas del sector asegurador, enfrentando los desafíos con estrategias vanguardistas y tecnologías de vanguardia.

En la era actual, la presencia en línea se erige como el pilar fundamental para cualquier empresa que aspire a un reconocimiento global. La creación y actualización de una página web no solo son ejercicios técnicos liderados por ingenieros, sino también una fusión de habilidades entre expertos en marketing y desarrollo web. La página web no solo busca atraer la atención del público, sino también posicionar a la empresa en el mercado global, compitiendo con sus pares.

En el contexto de SBS Seguros Colombia, con sede en Bogotá, se ha reconocido la importancia de la presencia web y se ha tomado la decisión estratégica de actualizar el Storybook utilizado en el desarrollo de software. Dada la complejidad y dinámica de este proceso, la empresa ha optado por colaborar con especialistas externos en el desarrollo web y consultoría, reconociendo la necesidad de una asesoría integral.

A pesar de la trayectoria exitosa de SBS Seguros y sus sólidas relaciones con clientes a lo largo de los años, la presencia digital no ha alcanzado los niveles deseados. La iniciativa de actualizar el Storybook busca no solo modernizar la infraestructura técnica sino también potenciar la interacción con los clientes. Se aspira a crear una página web atractiva y fácil de usar, así como implementar procesos que aceleren operaciones internas y mejoren la comunicación entre diferentes módulos de la empresa.

Este proyecto representa un paso estratégico para SBS Seguros Colombia en su búsqueda constante de la excelencia y la adaptación a los estándares tecnológicos actuales. La actualización del Storybook se concibe como un componente esencial para fortalecer la presencia digital, mejorar la experiencia del usuario y consolidar la posición de SBS Seguros en el competitivo mercado de seguros en Colombia.

CONTEXTO

SBS Seguros Colombia, siguiendo la visión de mantenimiento a la vanguardia de la industria de la seguridad, apuesta por modernizar sus procesos y herramientas tecnológicas. (Vue.js, s.f.) Durante esta iniciativa de mejora, la empresa decidió centrarse en actualizar su Storybook, una herramienta esencial para desarrollar y documentar los componentes de Vue.

El proyecto se centra en mejorar y actualizar el Storybook existente, que ha quedado obsoleto y no cumple con los requisitos cambiantes de la industria de la seguridad. La actualización del Storybook mejora la eficiencia del desarrollo y la documentación de componentes, así mismo espera reflejar avances tecnológicos y las mejores prácticas en el desarrollo front-end. En esta implementación también se debe asegurar que su integración sea de manera efectiva en el flujo de trabajo de la empresa, para así proporcionar una herramienta de documentación completa y actualizada para el equipo de desarrollo.

Esta optimización es una pieza clave del sector más amplio de SBS Seguros Colombia para mantener su competitividad y eficiencia en el sector de seguridad. El proyecto se llevará a cabo con un estudio de actualización tecnológica y mayor eficiencia en el desarrollo de componentes, que permitirá a la empresa adaptarse a una actividad en constante evolución. En respuesta a la imperante necesidad de mantener la eficiencia y coherencia en el desarrollo de software, este proyecto se enfoca en la actualización del Storybook en SBS Seguros, el cual se encontraba obsoleto debido a dependencias desactualizadas. La obsolescencia planteaba desafíos significativos para el desarrollo de interfaces de usuario, afectando la agilidad y calidad del proceso. En respuesta a la imperante necesidad de mantener la

eficiencia y coherencia en el desarrollo de software, este proyecto se enfoca en la actualización del Storybook en SBS Seguros, el cual se encontraba obsoleto debido a dependencias desactualizadas. La obsolescencia planteaba desafíos significativos para el desarrollo de interfaces de usuario, afectando la agilidad y calidad del proceso.

El objetivo principal de esta iniciativa consistió en modernizar el Storybook mediante la identificación y sustitución de dependencias obsoletas. La metodología abarcó un análisis minucioso de las dependencias existentes y su reemplazo con versiones más recientes y compatibles. Las pruebas rigurosas se llevaron a cabo en cada etapa para asegurar la estabilidad y compatibilidad del sistema.

Los resultados obtenidos destacan una mejora palpable en la eficiencia del desarrollo de interfaces. La eliminación de dependencias obsoletas no solo fortaleció la seguridad del sistema, sino que también proporcionó una base sólida para futuras actualizaciones. Este proyecto contribuye de manera significativa a la agilidad del desarrollo de software en SBS Seguros, posicionándola para adaptarse rápidamente a las demandas cambiantes de la industria.

El objetivo principal de esta iniciativa consistió en modernizar el Storybook mediante la identificación y sustitución de dependencias obsoletas. La metodología abarcó un análisis minucioso de las dependencias existentes y su reemplazo con versiones más recientes y compatibles. Las pruebas rigurosas se llevaron a cabo en cada etapa para asegurar la estabilidad y compatibilidad del sistema.

Los resultados obtenidos destacan una mejora palpable en la eficiencia del desarrollo de interfaces. La eliminación de dependencias obsoletas no solo fortaleció la seguridad del

sistema, sino que también proporcionó una base sólida para futuras actualizaciones. Este proyecto contribuye de manera significativa a la agilidad del desarrollo de software en SBS Seguros, posicionándola para adaptarse rápidamente a las demandas cambiantes de la industria.

OBJETIVOS

OBJETIVO PRINCIPAL

Actualizar el Storybook utilizado en el área de frontend de desarrollo de software de la empresa, optimizando su funcionamiento con el fin de mejorar la eficiencia en la creación y gestión de componentes de interfaz de usuario, garantizando la coherencia y calidad en las aplicaciones desarrolladas y contribuyendo a la competitividad de la empresa SBS Seguros Colombia en el mercado respectivo mercado.

OBJETIVOS SECUNDARIOS

1. Analizar detalladamente el estado actual del Storybook, identificando sus deficiencias y áreas de mejora en términos de eficiencia y calidad en el desarrollo de componentes de interfaz de usuario.
2. Identificar puntos débiles y áreas de mejora en el estado actual del libro mayor, con el fin de mejorar la calidad del desarrollo de los componentes de la interfaz de usuario.
3. Investigar últimas tendencias y mejores prácticas en el diseño de sistemas de diseño (design systems) y desarrollo de componentes de interfaz de usuario para aplicar estas innovaciones en la actualización del Storybook.

4. Proporcionar capacitación y soporte al equipo de desarrollo de SBS Seguros Colombia para que puedan utilizar efectivamente la versión actualizada del Storybook y aprovechar sus ventajas.

DESCRIPCIÓN DE LA PROBLEMÁTICA

La problemática general del proyecto de prácticas en SBS Seguros Colombia se centra en la obsolescencia de un Storybook utilizado en desarrollo de software. Esta herramienta es esencial para la creación y mantenimiento de la interfaz de usuario de las aplicaciones de la empresa, y su estado obsoleto limita la eficiencia y calidad del proceso de desarrollo.

Desde una perspectiva de conocimiento en el área de desarrollo de software y diseño de interfaces de usuario, esta problemática se enmarca en la necesidad de mantener las herramientas y tecnologías actualizadas para garantizar la eficacia en el desarrollo de software y la satisfacción del usuario final.

SOLUCIONES ACTUALES

La solución a la problemática de la obsolescencia del Storybook en SBS Seguros Colombia implica una serie de acciones estratégicas. En primer lugar, se debe llevar a cabo una evaluación exhaustiva de las necesidades actuales de desarrollo de software en la empresa. Esto incluye la identificación de las últimas tendencias y tecnologías en el campo de interfaces de usuario y desarrollo front-end.

Luego, se propone la actualización del Storybook a una versión más moderna y compatible con las tecnologías y estándares actuales. Esto podría incluir la migración a versiones más recientes de Vue.js y Storybook, así como la implementación de las mejores prácticas de documentación de componentes.

Finalmente, se enfatiza la importancia de la formación y capacitación continua del equipo de desarrollo en el uso efectivo de la nueva versión del Storybook. Esto garantizará que los miembros del equipo estén al tanto de las características y funcionalidades actualizadas y puedan aprovechar al máximo la herramienta en su trabajo diario. La capacitación es esencial para garantizar que la actualización del Storybook se traduzca en mejoras reales en la eficiencia y calidad del proceso de desarrollo de software en SBS Seguros Colombia.

JUSTIFICACIÓN DE LA INVESTIGACIÓN

La obsolescencia del Storybook en SBS Seguros Colombia es un desafío significativo que afecta directamente al equipo de desarrollo front-end. En este contexto, es fundamental comprender cómo la actualización y la implementación de esta herramienta pueden aportar mejoras sustanciales al proceso de desarrollo y, en última instancia, a la calidad de las soluciones de software entregadas.

La literatura existente subraya la importancia de las herramientas de desarrollo en la eficiencia y productividad del equipo de desarrollo. Según, las herramientas actualizadas y modernas permiten una colaboración más efectiva, mejor documentación de componentes y una experiencia de usuario más consistente. La actualización del Storybook se alinea con esta perspectiva y promete ofrecer ventajas notables al equipo de front-end de SBS Seguros Colombia.

La justificación de esta investigación se basa en la necesidad de proporcionar al equipo de desarrollo las herramientas y tecnologías más adecuadas para optimizar su trabajo. Esta actualización no solo mejorará la eficiencia en el proceso de desarrollo, sino que también permitirá una mayor colaboración, coherencia y calidad en la creación de componentes front-end. El impacto positivo de estas mejoras se traducirá directamente en soluciones de seguros de alta calidad para los clientes de SBS Seguros Colombia.

MARCO TEORICO

El desarrollo frontend, o desarrollo del lado del cliente, es una disciplina crucial en la creación de aplicaciones web modernas. Como destacan Flanagan y Mattox (2012), el frontend es la parte

de la aplicación que los usuarios finales experimentan directamente, abarcando la presentación visual, la interactividad y la usabilidad. Los desarrolladores frontend, mediante el uso de tecnologías como HTML, CSS y JavaScript, trabajan arduamente para traducir los conceptos de diseño en interfaces de usuario atractivas y funcionales. Este proceso implica enfrentar desafíos relacionados con la compatibilidad entre navegadores, la optimización del rendimiento y la implementación de prácticas actuales para garantizar experiencias de usuario cohesivas. En resumen, el desarrollo frontend es esencial para ofrecer aplicaciones web que no solo sean visualmente atractivas sino también intuitivas y eficientes para una variedad de usuarios.

En el mundo del desarrollo de aplicaciones, Vue.js se define como una herramienta clave para crear la parte visual de las aplicaciones, como botones, formularios y otros elementos con los que interactuamos. Es como el arquitecto que diseña cómo se verá y se comportará la interfaz de usuario. Como por ejemplo una casa: Vue.js sería como el plano detallado que te ayuda a construir cada habitación de manera eficiente y ordenada. SBS Seguros Colombia ha decidido utilizar Vue.js porque es moderno y eficiente, y muestra su compromiso con adoptar tecnologías avanzadas.

Ahora así mismo, hay otra herramienta llamada Storybook, que es como un catálogo visual donde puedes ver cómo se verán y comportarán diferentes partes de tu casa (o en este caso, componentes de Vue.js). Facilita que los equipos de diseño y construcción trabajen juntos, asegurando que todo sea coherente y de alta calidad.

En el ámbito del desarrollo front-end, Vue.js ha destacado como un framework ágil y eficiente para la creación de interfaces de usuario. (Vue.js, s.f.) Su enfoque reactivo y su estructura modular simplifican el desarrollo de aplicaciones interactivas y componentes reutilizables. La elección consciente de Vue.js por parte de SBS Seguros Colombia subraya su compromiso con tecnologías

de vanguardia y su reconocimiento de la importancia de un marco sólido para el desarrollo front-end. Storybook, siendo una herramienta esencial en el desarrollo front-end, se integra perfectamente con Vue.js al proporcionar un entorno dedicado para la documentación y visualización de componentes Vue. (Why Storybook?, s.f.) La capacidad de Storybook para representar visualmente componentes Vue.js no solo facilita la colaboración entre los equipos de diseño y desarrollo, sino que también mejora la calidad y coherencia de los elementos visuales en las aplicaciones desarrolladas con Vue.js.

La actualización del Storybook de Vue 2 a Vue 3, se alinea directamente con la evolución constante de Vue.js. La versión 3 de Vue introduce mejoras significativas en reactividad, rendimiento y capacidad de desarrollo. Este proyecto no solo busca mantenerse al día con las últimas versiones, sino también aprovechar las características mejoradas de Vue 3 para optimizar la eficiencia del desarrollo front-end. (Shilman, Storybook for Vue 3, 2021) Esta actualización implica no solo migrar a Vue 3, sino también adoptar la versión más reciente de Storybook, la 7.6. La integración de estas tecnologías es crucial para aprovechar al máximo las capacidades que nos ofrecen.

Esta decisión estratégica asegura la compatibilidad y aprovecha las características y mejoras más recientes ofrecidas por ambas tecnologías. La adaptabilidad a estas versiones más recientes sitúa a SBS Seguros Colombia en una posición sólida para abordar los desafíos cambiantes de la industria de seguros en Colombia.

La migración de Vue 2 a Vue 3 y la actualización de Storybook presentan desafíos, como la reescritura de componentes, ajustes en la sintaxis y la adaptación a nuevas API. Estrategias específicas se implementarán para abordar estos desafíos, incluyendo una revisión exhaustiva de

la documentación oficial de Vue.js y Storybook, así como la utilización de herramientas de migración automatizada cuando sea posible.

La implementación exitosa de la versión actualizada de Storybook con Vue 3 en SBS Seguros Colombia resultará en mejoras tangibles en la eficiencia del desarrollo front-end. La adopción de las últimas versiones y características permitirá un desarrollo más ágil y una implementación eficiente de componentes Vue en las aplicaciones de seguros. La nueva versión de Storybook, al ofrecer una documentación mejorada y visualización de componentes en un entorno Vue 3, garantizará la consistencia y calidad de la interfaz de usuario.

Los equipos de desarrollo podrán colaborar de manera más efectiva, reduciendo errores y asegurando una experiencia de usuario uniforme en las aplicaciones de SBS Seguros Colombia. Este proyecto no solo se trata de mantenerse al día, sino de mirar hacia el futuro. La adopción de Vue 3 y Storybook 7.6 es un paso estratégico que coloca a SBS Seguros Colombia en una posición ventajosa para abrazar innovaciones futuras en el desarrollo front-end, asegurando una presencia continua y sólida en la industria de seguros en Colombia.

METODOLOGÍA

La metodología propuesta para el proyecto de desarrollo del Storybook en SBS Seguros Colombia establece un enfoque sistemático para la actualización y mejora de esta herramienta esencial. A través de una secuencia de fases cuidadosamente planificadas, se busca optimizar la eficiencia en el desarrollo front-end y la documentación, asegurando que el Storybook esté alineado con las tendencias actuales y las mejores prácticas del equipo de desarrollo. Cada fase

desempeñará un papel crítico en el proceso, desde la evaluación inicial hasta la implementación, capacitación y mantenimiento continuo, contribuyendo al éxito global del proyecto.

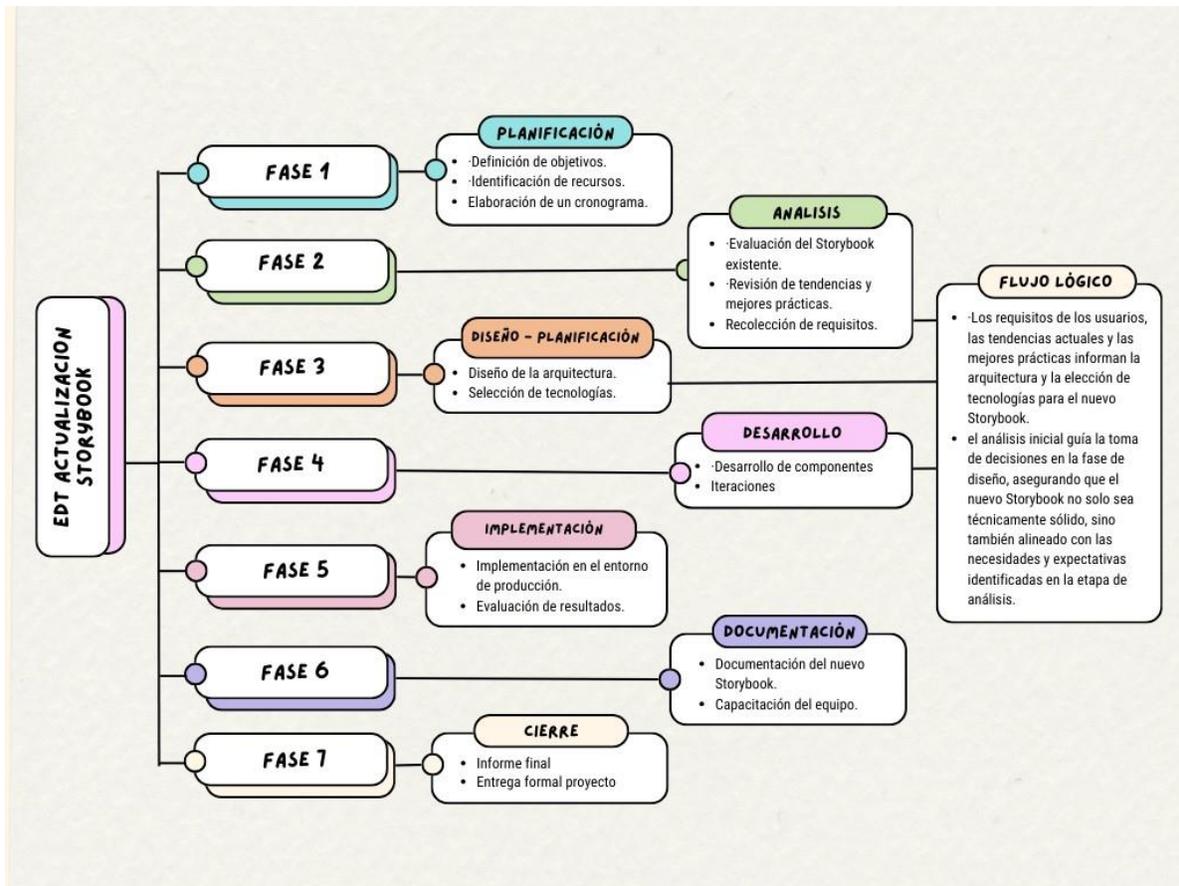


Ilustración 1: Diagrama EDT – Autoría propia

DESCRIPCIÓN DE ACTIVIDADES

1. Planeación

En la fase inicial, se llevó a cabo un análisis profundo del Storybook existente, se revisaron las tendencias y mejores prácticas actuales, y se recopilaron los requisitos de los usuarios. Este análisis proporciona una comprensión completa de las limitaciones y oportunidades del proyecto. Se plantearon tres actividades para esta fase:

- a) **Definición de Objetivos:** En esta actividad, se convoca una reunión con los interesados clave para definir de manera precisa y detallada los objetivos del proyecto. Esto incluyó la actualización del Storybook, los resultados esperados y los criterios de éxito.
- b) **Identificación de Recursos:** Se lleva a cabo un análisis exhaustivo para identificar los recursos necesarios, incluyendo la asignación de roles y responsabilidades, la disponibilidad de personal, herramientas de desarrollo.
- c) **Cronograma:** Se elabora un cronograma detallado que estableció las fechas clave, hitos y duración de cada fase y actividad del proyecto. Esto ayuda a garantizar que el proyecto se mantuviera en el camino y dentro del plazo establecido.

Como resultado de esta fase inicial de planeación, se llevó a cabo un análisis exhaustivo del Storybook existente, permitiendo la recopilación de los requisitos. Para definir los objetivos del proyecto, se convocó una reunión con el equipo frontend de SBS Seguros, el día, donde se detallaron los objetivos de actualización del Storybook y sus resultados esperados. Además, se realizó una identificación de recursos mediante un análisis que incluyó la asignación de responsabilidades y herramientas de desarrollo para su elaboración.

Finalmente, se elabora un cronograma detallado que estableció fechas y duración de cada fase y actividad del proyecto, asegurando así que el proyecto se mantuviera en el camino y se completara dentro del plazo establecido. Es importante destacar que el cronograma experimentó diversas modificaciones durante la fase de planeación debido a factores externos, como la disponibilidad y programación de reuniones por parte del equipo. La interacción con agentes externos influyó en la flexibilidad del cronograma, requiriendo ajustes para adaptarse a cambios imprevistos. Asimismo, se observó que la duración de

algunas fases no fue precisa, ya que ciertos aspectos del proyecto estaban sujetos a variaciones inesperadas.

2. Análisis y Evaluación

En este procedimiento se realiza un análisis profundo del Storybook existente, se revisan las tendencias y mejores prácticas actuales, y se recopilan los requisitos de los usuarios.

Las actividades fueron las siguientes:

- a) **Revisión de Tendencias y Mejores Prácticas:** Se realiza una investigación exhaustiva para identificar las tendencias actuales en el desarrollo front-end, las tecnologías emergentes y las mejores prácticas en la documentación de componentes.
- b) **Recolección de Requisitos:** Se recopilaron los requisitos específicos de los miembros del equipo de desarrollo, incluyendo sus necesidades, preferencias y expectativas con respecto al nuevo Storybook.

Como resultado de esta fase de Análisis y Evaluación, se lleva a cabo una investigación detallada de la documentación de cada dependencia del proyecto, abarcando tanto la versión actual como la antigua del Storybook. Esta actividad crucial permitió un entendimiento profundo de las interdependencias y estructuras existentes, identificando posibles áreas de mejora y optimización. Se examinaron minuciosamente los documentos relacionados con las dependencias del proyecto para evaluar su coherencia y alineación con las mejores prácticas actuales en el desarrollo front-end y se realizan sus respectivas comparaciones para aclarar las mejoras actuales del Storybook en cuanto a sus dependencias.

3. Diseño y Planificación Técnica

En esta fase, se diseñó la arquitectura del nuevo Storybook, se eligieron las tecnologías apropiadas y se planifica cómo se integrará en el flujo de trabajo existente. Este diseño técnico es crucial para el éxito de la implementación.

Diseño de la Arquitectura: En esta actividad, se elabora un diseño detallado de la arquitectura del nuevo Storybook. Se define las estructuras de carpetas, las rutas, los estándares de codificación y la organización de componentes.

- a) **Selección de Tecnologías:** Se toman decisiones sobre las tecnologías y las herramientas que se utilizaron en la actualización del Storybook. Esto incluye la selección de la versión de Vue, las bibliotecas y las herramientas de Storybook.

En esta fase, se logró un resultado positivo al diseñar la arquitectura detallada para el nuevo Storybook y realizar la selección cuidadosa de tecnologías, optando por la versión 7.6 alpha de Storybook y Vue 3. El diseño contempla estructuras de carpetas, rutas y estándares de codificación, proporcionando una base sólida. La elección de tecnologías asegura una integración eficiente, marcando un avance significativo en la preparación para la implementación exitosa del proyecto.

4. Desarrollo

Desde esta etapa se lleva a cabo las acciones concretas. El equipo de desarrollo trabaja en la implementación de los componentes del nuevo Storybook, siguiendo las pautas y estándares establecidos. Las pruebas y la depuración son esenciales para garantizar la calidad del trabajo.

- a) **Desarrollo de Componentes:** El equipo de desarrollo procede con la implementación de los componentes del nuevo Storybook siguiendo las mejores prácticas y estándares previamente definidos. Se utilizan las tecnologías seleccionadas en la fase anterior.
- b) **Iteraciones:** Se lleva a cabo iteraciones basadas en la retroalimentación recibida del equipo de desarrollo y los usuarios finales. Las mejoras se implementan para asegurar la calidad y la eficiencia del Storybook.

5. Implementación y evaluación

En esta fase, el nuevo Storybook se implementa en el entorno de producción. Es el momento de poner a prueba el trabajo realizado y evaluar si se cumplen los objetivos del proyecto. La evaluación proporciona información valiosa para futuras mejoras.

- a) **Implementación en el Entorno de Producción:** La nueva versión del Storybook se implementa en el entorno de producción de SBS Seguros Colombia. Se realizan pruebas adicionales antes de la implementación final.
- b) **Evaluación de Resultados:** Se lleva a cabo una evaluación exhaustiva para asegurarse de que los objetivos del proyecto se cumplan. Se recopilarán datos y métricas para medir la eficiencia y la funcionalidad del nuevo Storybook.

6. Entrega y evaluación

En esta etapa final, se crea la documentación esencial para el uso del nuevo Storybook. Además, se proporciona capacitación al equipo de desarrollo para que puedan utilizar eficazmente la herramienta. La documentación y la capacitación son fundamentales para garantizar una adopción exitosa del nuevo Storybook. Debidamente se entrega el prototipo final, para conocer la opinión de los líderes del área de desarrollo en la empresa, conocer si se cumplieron con los requerimientos principales que ellos pidieron y con esto cumplir con las propuestas dichas en el primer procedimiento

- a) **Documentación del Nuevo Storybook:** Se genera una documentación completa y actualizada que incluirá guías de uso, documentación técnica y ejemplos de componentes. Esta documentación será esencial para el equipo de desarrollo.
- b) **Capacitación:** Se ofrece capacitación a los miembros del equipo de desarrollo para asegurar que puedan utilizar eficazmente el nuevo Storybook y aprovechar sus capacidades al máximo.

En la fase de entrega y evaluación, se ha completado la documentación esencial del nuevo Storybook, abarcando guías de uso, documentación técnica y ejemplos de componentes. Aunque aún no se ha llevado a cabo la capacitación del equipo de desarrollo, la documentación proporciona una valiosa referencia para asegurar una adopción exitosa. El prototipo final se ha entregado para obtener la opinión de los líderes del área de desarrollo en la empresa, con el objetivo de validar si se cumplen los requisitos principales solicitados. Este paso es crucial para verificar si se han

alcanzado las metas propuestas inicialmente en el proceso, y orientará ajustes finales antes de la implementación completa.

7. Cierre del proyecto

La fase de cierre marca el final del proyecto y es crucial para garantizar que se cumplan todos los requisitos y que el trabajo esté completo. En esta etapa, se prepara un informe final que resume los logros, lecciones aprendidas y recomendaciones para proyectos futuros. Se realiza una entrega formal del proyecto a SBS Seguros Colombia, asegurándose de que todos los aspectos estén completos y funcionales. El cierre del proyecto asegura que todos los involucrados tengan una comprensión clara de los resultados y que se haya cumplido con éxito el objetivo de actualizar el Storybook.

- a) **Informe Final:** En esta actividad, se prepara un informe final que resumirá todos los aspectos del proyecto. El informe destacará los logros, las lecciones aprendidas, los desafíos superados y las recomendaciones para futuros proyectos similares. Este informe será un recurso valioso para la documentación y la retroalimentación.
- b) **Entrega Formal:** La entrega formal del proyecto se realizará de acuerdo con los procedimientos y protocolos establecidos en SBS Seguros Colombia. Esto incluye la entrega de todos los activos relacionados con el proyecto, como el nuevo Storybook, la documentación, las herramientas y otros recursos.

En la fase de cierre del proyecto, se ha completado el informe final que resume exhaustivamente todos los aspectos del proyecto de actualización del Storybook. Este informe destaca los logros obtenidos, el proceso, los desafíos superados y ofrece recomendaciones valiosas para proyectos futuros similares. Aunque la entrega formal aún no se ha realizado, se sigue de cerca el protocolo establecido por SBS Seguros Colombia, asegurándose de que todos los activos relacionados con el proyecto estén listos para su entrega, incluyendo el nuevo Storybook, la documentación detallada, las herramientas y otros recursos clave. Este cierre del proyecto representa el cumplimiento de todos los requisitos y que el trabajo esté completo de manera satisfactoria.

IMPLEMENTACIÓN STORYBOOK OPTIMIZADO

DESCRIPCIÓN DEL PROCESO

En respuesta a la necesidad de mejorar y modernizar el Storybook existente, SBS Seguros Colombia ha asignado el proyecto de actualización a fin de optimizar la gestión de componentes visuales en las aplicaciones. El Storybook actual ha sido un componente crucial en el flujo de trabajo de desarrollo, pero se hace un reconocimiento a la importancia de mantener al día las últimas tecnologías y prácticas de desarrollo.

Tras un análisis exhaustivo de las tendencias actuales y las necesidades de nuestros equipos de desarrollo, se ha decidido llevar a cabo la actualización utilizando tecnologías modernas como Vue.js y Storybook. Esta elección se basa en la versatilidad y eficacia de Vue.js para el desarrollo front-end, junto con la capacidad de Storybook para ofrecer un entorno de desarrollo y prueba de componentes eficiente.

El proceso de implementación se realizará en servidores propios para garantizar la seguridad y el control total sobre la plataforma. Dada la importancia de asegurar la calidad y funcionalidad de la actualización, se establecerán dos entornos adicionales: uno de prueba y otro de preproducción.

En el entorno de prueba, realizaremos pruebas para validar la funcionalidad de cada componente individual y su compatibilidad con las tecnologías seleccionadas. Esta fase permitirá ajustes y correcciones antes de avanzar a la siguiente etapa.

Posteriormente, en el entorno de preproducción, simularemos condiciones más cercanas al entorno de producción, ejecutando pruebas de rendimiento y evaluando la interacción con otros sistemas existentes. Este enfoque por fases y la implementación cuidadosa en entornos controlados asegurarán una transición fluida y exitosa hacia la implementación final en el entorno de producción de SBS Seguros Colombia.

Este proyecto de actualización del Storybook es crucial para mejorar la eficiencia y la calidad del desarrollo de software, y nos comprometemos a llevar a cabo cada fase con el máximo rigor y atención para garantizar el éxito del proyecto.

RECOLECCIÓN DE REQUISITOS

El Storybook utilizado hasta el momento por la empresa SBS Seguros Colombia, aunque ha sido una parte valiosa del flujo de trabajo de desarrollo, requiere una modernización para mantenerse alineado con las tendencias y avances actuales en el desarrollo front-end y las mejores prácticas de documentación de componentes. Esta iniciativa busca actualizar y optimizar el Storybook para mejorar la eficiencia en el desarrollo, la colaboración entre los miembros del equipo y la calidad de la documentación de componentes. La elección de actualizar el Storybook en SBS Seguros

Colombia se basa en la necesidad de mantener la coherencia y la calidad en el desarrollo de componentes front-end.

Para llevar a cabo esta actualización, se realiza un proceso que involucra la evaluación del Storybook existente, la selección de tecnologías actualizadas y la implementación de las mejoras en una serie de etapas planificadas.

El resultado esperado de este proyecto es un Storybook modernizado y altamente eficiente que permitirá a SBS Seguros Colombia mantenerse a la vanguardia en el desarrollo front-end. Además, esta actualización facilitará la colaboración y la coherencia en todo el equipo de desarrollo, lo que contribuirá a un entorno de trabajo más productivo y eficiente.

No.	Requisito	Descripción	Rol
1	Actualización del Framework Vue.js	La versión actual de Vue.js debe ser actualizada a la última versión.	Practicante desarrollador Junior
2	Actualización del Framework Storybook	Se debe actualizar el Storybook a la versión más reciente disponible.	Practicante desarrollador Junior
3	Creación de una arquitectura de componentes actualizada	Diseñar una arquitectura de componentes que cumpla con las mejores prácticas de desarrollo front-end.	Practicante desarrollador Junior
4	Pruebas	Realización de pruebas simples para garantizar el funcionamiento de los componentes.	Practicante desarrollador Junior
5	Capacitación del equipo de desarrollo	Proporcionar capacitación y formación al equipo para utilizar eficazmente el nuevo Storybook.	Practicante desarrollador Junior
6	Implementación de un proceso de revisión de código	Establecer un proceso de revisión de código para asegurar la coherencia y calidad en el desarrollo.	Practicante desarrollador Junior

7	Seguimiento de tendencias y mejores prácticas	Mantenerse al tanto de las tendencias y mejores prácticas en desarrollo front-end para actualizaciones futuras.	Practicante desarrollador Junior
9	Integración de control de versiones	Implementar un sistema de control de versiones para un seguimiento y colaboración más efectivos.	Practicante desarrollador Junior

Tabla 1: Requerimientos

REVISIÓN DETALLADA VERSION 5.3.17 STORYBOOK

Explorando en detalle las características clave de la versión 5.3.17 de Storybook, la popular herramienta para el desarrollo de interfaces de usuario. Según (Shilman, Storybook.js, 2020), esta es una versión que no solo refuerza las capacidades fundamentales de desarrollo de componentes, sino que también aborda las demandas actuales de documentación y empaquetado de sistemas de diseño reutilizables. A continuación, se destacan los aspectos más significativos:

Escritura de Historias y Documentación en MDX

Storybook 5.3.17 presenta MDX, un formato innovador que permite escribir documentación en Markdown y definir ejemplos de componentes en el mismo archivo. (Shilman, Storybook.js, 2020) menciona que esto simplifica la creación de documentación personalizada y fomenta la colaboración entre equipos técnicos y no técnicos.

Generación Automática de Documentación para Más Frameworks

La capacidad de generar automáticamente documentación de las mejores prácticas se amplía a múltiples frameworks, incluyendo React, Vue, Angular, Ember y Web Components. Según esto (Shilman, Storybook.js, 2020) afirma que Storybook Docs se posiciona como una solución integral para documentar interfaces de usuario de manera eficiente.

Configuración Declarativa

La configuración de Storybook experimenta una importante mejora con una aproximación declarativa en el archivo principal (main.js). Así mismo, (Shilman, Storybook.js, 2020) alega que esto simplifica la configuración de historias, complementos e incluso webpack, mejorando significativamente la experiencia del desarrollador.

Integración con Herramientas de Diseño

La potente integración con herramientas de diseño, como Sketch, Figma y Adobe XD, se fortalece aún más, consolidando a Storybook como el punto de integración estándar para estas herramientas en el desarrollo de interfaces de usuario. (Shilman, Storybook.js, 2020)

Soporte Oficial para Web Components

Storybook 5.3 anuncia oficialmente el soporte para Web Components, proporcionando un paquete dedicado (@storybook/web-components) para facilitar el desarrollo aislado y la documentación de las mejores prácticas en proyectos basados en Web Components.

Mejoras Adicionales

- Deep Linking: Posibilidad de vincular directamente a secciones específicas en Storybook Docs.
- Permalinks: Identificadores únicos para componentes e historias, evitando enlaces rotos durante reorganizaciones.
- Templates Configurables para DocsPage: Personalización de las plantillas aplicadas a las páginas de documentación.

- Soporte de TypeScript: Conversión de la mayoría de los paquetes importantes a TypeScript para una mejor mantenibilidad y tipos nativos para los usuarios.

Standalone CSF

La introducción de `@storybook/csf` permite una implementación sencilla del Component Story Format (CSF) de Storybook en otras herramientas de diseño y desarrollo. (Shilman, Storybook.js, 2020)

EVALUACIÓN DE PROBLEMAS DE RENDIMIENTO Y POSIBLES ÁREAS DE MEJORA.

Con la creciente complejidad de los sistemas de diseño y la documentación, es crucial evaluar el rendimiento de Storybook 5.3 para garantizar una experiencia de desarrollo fluida. La inclusión de MDX para escribir historias y documentación en el mismo archivo proporciona una gran flexibilidad, pero se debe evaluar cómo esta nueva funcionalidad afecta el rendimiento general, especialmente en proyectos extensos con numerosos componentes.

La optimización de la generación automática de documentación para diferentes frameworks, como React, Vue, Angular, Ember y Web Components, es una adición valiosa. Sin embargo, es fundamental evaluar la eficiencia de esta función al manejar proyectos más grandes con múltiples frameworks en uso simultáneo.

El problema observado se manifiesta en la imposibilidad de cargar Storybook 5.3 cuando se utiliza una versión obsoleta. Este inconveniente se traduce en un impacto significativo en el flujo de trabajo de desarrollo, generando retrasos y dificultades en la colaboración entre equipos.

Análisis de Causas

Compatibilidad Obsoleta:	La versión obsoleta de Storybook puede no ser compatible con las últimas actualizaciones de dependencias y tecnologías en el proyecto.
Dependencias Desactualizadas:	La presencia de dependencias desactualizadas puede generar conflictos y afectar la carga adecuada de Storybook.
Cambios en el Entorno:	Se exploraron cambios recientes en el entorno de desarrollo, como actualizaciones del sistema operativo o modificaciones en la configuración del proyecto, que podrían haber contribuido al problema.

Tabla 2: Causas de desactualización

INVESTIGACIÓN Y FAMILIARIZACIÓN CON LAS NUEVAS VERSIONES

La reciente llegada de Storybook 7.6 marca el último lanzamiento menor en la serie 7 de Storybook. Esta versión se centra en mejorar el rendimiento y la experiencia de usuario, fortalecer las integraciones con herramientas en el ecosistema frontend, y preparar el terreno para el próximo lanzamiento importante, Storybook 8.

Característica evaluada	Evaluación
SWC: Rendimiento Mejorado en Proyectos Webpack:	Implementar el soporte mejorado de SWC en Storybook 7.6 resultó en una reducción

	<p>impresionante de los tiempos de inicio en proyectos Webpack, alcanzando una mejora del 50% o más.</p>
<p>Modo de Pruebas Eficiente y Utilidades de Construcción Rápida:</p>	<p>La introducción de Storybook Test y el nuevo modo de construcción de pruebas proporcionaron una biblioteca de pruebas más eficiente y redujo significativamente los tiempos de construcción estática al probar.</p>
<p>Mocks Experimentales para SvelteKit:</p>	<p>Los nuevos mocks experimentales para SvelteKit permiten a los desarrolladores escribir historias seguras para componentes que importan módulos específicos de SvelteKit, mejorando así la capacidad de prueba y desarrollo.</p>
<p>Actualización de React-docgen:</p>	<p>La actualización a React-docgen v7 en Storybook 7.6 es un paso crucial hacia un rendimiento mejorado en la próxima versión principal, Storybook 8, con reducciones sustanciales en los tiempos de inicio.</p>
<p>Mejoras en Accesibilidad y Tematización:</p>	<p>Storybook 7.6 abordó problemas de accesibilidad y tematización, mejorando significativamente la experiencia del usuario y proporcionando un control más preciso sobre la apariencia de la interfaz.</p>
<p>Herramienta de Diagnóstico storybook-doctor:</p>	<p>La incorporación de storybook-doctor como una nueva herramienta de diagnóstico permitió realizar verificaciones generales de la salud del proyecto, identificando problemas y proporcionando soluciones rápidas.</p>

Eliminación de React como Dependencia para Addons:	En preparación para Storybook 8, la versión 7.6 eliminó a React como una dependencia principal para todos los paquetes de Storybook, excepto Storybook Docs, simplificando así las dependencias del proyecto.
Sitio de Documentación Nuevo y Mejorado:	La renovación del sitio de documentación de Storybook en la versión 7.6 ha mejorado la usabilidad y proporciona una experiencia más personalizada, recordando las preferencias del usuario.
Depreciación de Storyshots y Vue2:	La decisión de deprecar Storyshots y poner fin al soporte para Vue2 en la versión 7.6 allana el camino para mejoras futuras y garantiza la continuidad con Storybook 8.
Documentación	(Shilman, Storybook 7.6, 2023)

Tabla 3: Evaluación características storybook v7.6

IDENTIFICACIÓN Y ACTUALIZACIÓN DE COMPONENTES OBSOLETOS Y DEPENDENCIAS DESACTUALIZADAS

A continuación, se explora detalladamente las dependencias del Storybook obsoleto que se encuentran actualmente en uso en el entorno de desarrollo, y así mismo las mejoras en su actualización de la versión elaborada. Cada una de estas dependencias desempeña un papel crucial en la funcionalidad y el desarrollo del Storybook, y su análisis minucioso es esencial para la planificación de la actualización. Desde bibliotecas fundamentales hasta plugins específicos de Storybook, estas dependencias han contribuido al desarrollo y la visualización efectiva de componentes a lo largo del tiempo.

La lista a continuación abarca tanto dependencias de producción como herramientas de desarrollo, cada una seleccionada para cumplir una función específica en el proceso de desarrollo y presentación de componentes. A través de este análisis, obtendremos una comprensión profunda de la infraestructura existente, identificando áreas de mejora y oportunidades para la adopción de tecnologías más modernas.

@fullhuman/postcss-purgecss		
Característica	Storybook Obsoleto	Storybook Actualizado
Biblioteca de diseño CSS para estilizar componentes	4.4.1	5.0.0
Soporte para variables de lista segura	No	Sí aplica
Soporte para opciones de ContentGlobs en gulp-purgecss	No	Sí aplica
Soporte para mapas fuente en purgecss-webpack-plugin	No	Sí aplica
Exportación con nombre en purgecss-webpack-plugin	No	Sí aplica
Carga de opciones desde la configuración de purgecss en postcss-purgecss	No	Sí aplica
Carga automática del archivo de configuración en purgecss-webpack-plugin	No	Sí aplica
Eliminación de compatibilidad con webpack 4 en purgecss-webpack-plugin	No	Sí aplica
Documentación	(David, 2022) (purgecss, 2019)	

Tabla 4: Mejora dependencias *fullhuman/postcss-purgecss*

@mathieustan/vue-datepicker		
Característica	Storybook Obsoleto	Storybook Actualizado
Componente de selección de fecha para Vue.js	V	
Tipo de Selector	Calendario de fecha interactivo y seleccionador de fecha.	
Personalización	Ofrece opciones de personalización para adaptarse a diferentes estilos y temas.	
Rango de Fechas	Admite la selección de rangos de fechas.	
Eventos Personalizados	Proporciona eventos personalizados para manipular el comportamiento del calendario.	
Internacionalización	Soporte para varios idiomas y formatos de fecha.	

Responsive	Puede ser diseñado para ser sensible y funcionar bien en dispositivos móviles.
Formato de Fecha Personalizado	Permite especificar el formato de fecha según las preferencias del usuario.
Documentación	(vue-datepicker, 2020) (vcalendar, s.f.)

Tabla 5: Mejora dependencias *mathieustan/vue-datepicker*

@storybook/addon-viewport		
Característica	Storybook Obsoleto	Storybook Actualizado
Addon para ajustar la vista del componente en Storybook.	5.3.17	7.5.3
Configuración Global	Configurado con parámetros de historia en preview.js .	Configurado con parámetros de historia en preview.js o preview.ts .
Configuración por Componente o Historia	Puedes configurar parámetros específicos utilizando la API de parámetros estándar.	Puedes configurar parámetros específicos utilizando la API de parámetros estándar. Ejemplo usando Component Story Format 3.
Documentación	(Storybook Viewport Addon, 2020)	(Viewport, 2023)

Tabla 6: Mejora dependencias *storybook/addon-viewport*

storybook/addons		
Característica	Storybook Obsoleto	Storybook Actualizado
Conjunto de addons para extender las capacidades de Storybook	5.3.17	7.5.3
Instalación	npm i -D @storybook/addon-actions	Parte integral de Storybook, ya incluido en la instalación esencial.
Configuración Global	Agrega ' @storybook/addon-actions/register ' a la lista de addons en .storybook/main.js .	La configuración global se realiza mediante la instalación y configuración de addons específicos.
Uso Básico	Importa action desde @storybook/addon-actions y lo utiliza para manejar eventos.	Se centra en la extensibilidad y personalización de Storybook mediante diversos addons.
Manejo de Múltiples Acciones	Puedes usar actions para crear muchas acciones a la vez.	La extensibilidad y funcionalidades específicas dependen de los addons instalados.

Decoradores de Acciones	Introduce decorate para procesar datos de acción antes de enviarlos al logger.	Los addons pueden afectar la interfaz de usuario, el panel de addons y el rendimiento general.
Configuración Específica	Configura parámetros como depth , clearOnStoryChange , y limit globalmente o por acción.	La configuración y funcionalidades específicas dependen del addon instalado.
Disponibilidad de Opciones	Opciones como depth , clearOnStoryChange , y limit .	Variedad de opciones según los requisitos del addon específico.
withActions Decorator	Ofrece el decorador withActions para definir manejadores de acciones de manera declarativa.	La extensibilidad y personalización se realizan mediante addons específicos.
Documentación	(Storybook Addon Actions, 2020)	(Storybook Addons, 2023)

Tabla 7: Mejora dependencias *storybook/addons*

storybook/theming		
Característica	Storybook Obsoleto	Storybook Actualizado
Permite personalizar el tema visual de Storybook.	5.3.17	7.5.3
API de Theming	Storybook 5 utiliza la API de theming.	
Theming Global	Es posible aplicar theming global a Storybook.	
Temas Predeterminados	Incluye los temas "light" y "dark".	
Configuración Global del Tema	Configuración del tema en <code>.storybook/manager.js</code> .	
Theming en Docs Independientemente	Storybook Docs utiliza el mismo sistema de theming, pero se tematiza de forma independiente.	
Crear un Tema Personalizado	Utiliza la función <code>create()</code> de <code>@storybook/theming/create</code> en un nuevo archivo.	
Ejemplo de Tema Personalizado	Ejemplo: <code>.storybook/YourTheme.js</code>	
Ejemplo de Tema Avanzado	Fuentes y colores personalizados.	
CSS Escape Hatches	Usa nombres de clase para estilizar componentes de UI y Docs en <code>.storybook/manager-head.html</code> y <code>.storybook/preview-head.html</code> .	
Sustituciones de Componentes MDX	Sustituye componentes renderizados desde Markdown en <code>.storybook/preview.js</code> utilizando el parámetro <code>components</code> .	
Addons y Creación de Temas	Algunos addons requieren variables de tema específicas. Admite la API oficial y addons populares para una experiencia consistente.	

Ejemplo de Tema para Addon Actions	Ejemplo de suministrar variables de tema adicionales para el addon Actions.
Documentación	(Storybook Theming, 2019) (Theming, 2023)

Tabla 8: Mejora dependencias *storybook/theming*

@storybook/vue		
Característica	Storybook Obsoleto	Storybook Actualizado
Integración de Vue.js para Storybook.	5.3.17	7.5.13
Objetivo Principal	Construir y documentar sistemas de diseño de producción	Construir UIs en aislamiento, capturar variaciones como "historias"
Principales Características	Escritura de historias y docs en MDX - Generación de documentación automática - Configuración declarativa en main.js - Integración con herramientas de diseño - Soporte oficial para Web Components	Desarrollo en aislamiento de UI - Captura de variaciones como "historias" - Directorio interactivo de componentes y sus historias - Compatible con varios frameworks y herramientas
MDX (Markdown eXtended)	Sí, nuevo formato para escribir historias y documentos	Sí, destaca la importancia de escribir historias y documentos junto con el código
Frameworks Soportados	React, Vue, Angular, Ember, Web Components, y más	React, Vue, Angular, y otros (no específicamente mencionados)
Configuración	Declarativa en main.js, simplificación de configuraciones	Enfatiza la configuración declarativa como avance importante
Integración con Herramientas de Diseño	Conectividad con Sketch, Figma, Abstract, Adobe XD	Enfatiza el soporte para diversas herramientas de diseño como parte integral
Soporte Web Components	Oficialmente soportado con @storybook/web-components	No menciona soporte específico, pero destaca desarrollo en aislamiento para cualquier componente de UI
Beneficios Principales	- Desarrollo más duradero - Pruebas pragmáticas y reproducibles - Documentación única de UI - Compartición efectiva de cómo funciona la UI - Automatización de flujos de trabajo	
Documentación	(Storybook 5.3, 2020)	(@storybook/vue, 2023)

Tabla 9: Mejora dependencias *storybook/vue*

bootstrap		
Característica	Storybook Obsoleto	Storybook Actualizado
Plugin PostCSS para eliminar el CSS no utilizado.	4.1.1	5.3.2
Inclusión de Bootstrap	CDN jsDelivr	CDN jsDelivr
Estructura de Enlaces (CDN)	Un solo enlace para CSS y JS (mismo script)	Enlaces separados para CSS y JS (Popper y Bootstrap)
Plantilla de Inicio	Metaetiquetas, enlace CSS, scripts JS	Metaetiquetas, enlace CSS, script JS
Enlace CDN (CSS)	Mismo enlace CDN para CSS	Enlace CDN separado para CSS
Enlace CDN (JS)	Mismo enlace CDN para JS (Popper y Bootstrap)	Enlaces CDN separados para Popper y Bootstrap
Componentes JS	Mismos componentes (alertas, botones, carrusel, modales, barras de navegación, etc.)	
Configuraciones Globales	HTML5, metaetiqueta responsiva, cambio en el modelo de caja, Reboot	
Documentación	(Arranque 4.4.1, 2019)	(Get started with Bootstrap, s.f.)

Tabla 10: Mejora dependencias *bootstrap*

core.js		
Característica	Storybook Obsoleto	Storybook Actualizado
Biblioteca que proporciona funcionalidades de JavaScript modernas en entornos antiguos	V 3.4.6	V 3.33.2
Datos de compatibilidad Safari	Se agregaron datos de compatibilidad con Safari 13.1	Mejoras en datos de compatibilidad: URL.canParse marcado como compatible con Chromium 120, mapeo de datos compatibles con Opera Android 78 actualizado, se agregó mapeo de datos compatibles con Electron 29
Polyfill simplificado structuredClone	No se especifica	Se simplificó el polyfill structuredClone, se evitó el segundo paso del árbol en casos de transferencia, se agregó soporte SuppressedError para structuredClone polyfill
Eliminación de dependencias innecesarias	No se especifica	Se eliminaron dependencias innecesarias no especificadas ArrayBuffery cuya falta podría causar errores en algunas entradas en IE10.DataViewstructuredClone

Manejo fijo de números fraccionarios	No se especifica	Se realizó un manejo fijo de la parte de números fraccionarios en Number.fromString
Documentación		

Tabla 11: Mejora dependencias *core.js*

vee-validate		
Característica	Storybook Obsoleto	Storybook Actualizado
Biblioteca para validación de formularios en Vue.js	Vee-Validate 3	Vee-Validate 4
Descripción General	Aborda la validación de formularios y la mejora de la UX.	Trata los principales puntos problemáticos de los formularios, proporciona abstracciones y se integra con la UI.
Forma de Implementación	Utiliza Composition API y Higher-order components (HOC).	Usa Composition API y Higher-order components (HOC).
Instalación	yarn add vee-validate@3 o npm install vee-validate@3	yarn add vee-validate@4 o npm install vee-validate@4
Creación de Reglas	VeeValidate.extend(...)	extend(...)
Registro de Componente Global	Vue.component('ValidationProvider', VeeValidate.ValidationProvider);	Vue.component('ValidationProvider', ValidationProvider);
Ejemplo Básico	<ValidationProvider rules="secret" v-slot="{ errors }"> ... </ValidationProvider>	<ValidationProvider rules="secret" v-slot="{ errors }"> ... </ValidationProvider>
Soporte de Composition API	No especifica	Se menciona explícitamente el soporte para Composition API.
Enfoque en Lógica sin UI	Abstrae la lógica sin proporcionar una UI específica.	Abstrae la lógica sin una UI específica para no interferir con la personalización.
Uso con Bibliotecas de Componentes	Se integra con varias bibliotecas para Vue.js.	Destaca la integración sin necesidad de tratamientos o ajustes especiales.
Enfoque en Componentes UI	No aborda la personalización de la interfaz de usuario.	Aborda la personalización de la interfaz de usuario sin costos ocultos ni conflictos.
Manejo de Estado y Valor de Formularios	No especifica	Abstrae el manejo de estado y valor de formularios.
Validación Sincrónica y Asincrónica	No especifica	Menciona el soporte para validaciones síncronas y asíncronas.

Forma de Uso en Script	<code><script src="path/to/vue.js"></script> ...</code>	Importación con ES6+: <code>import { ValidationProvider, extend } from 'vee-validate';</code>
Compatibilidad con Bibliotecas UI	Requiere consideración especial para la compatibilidad.	Se destaca la integración sin necesidad de ajustes especiales.
Versión Mencionada en la Documentación	No especifica	Se hace referencia explícita a la versión 4 en la documentación.
Documentación		

Tabla 12: Mejora dependencias *vee-validate*

vue-i18n		
Característica	Storybook Obsoleto	Storybook Actualizado
Biblioteca de internacionalización para Vue.js.	v8.15.5	v9.6.5
Sintaxis de Configuración	Utiliza el objeto <code>VueI18n</code> en la opción i18n del componente <code>Vue</code> .	Introduce createI18n y useI18n para configuración.
Componente de Localización	Utiliza el componente <code><i18n></code> y la directiva v-t .	Mantiene el componente <code><i18n></code> y la directiva v-t .
API de Traducción	Métodos basados en instancias de <code>VueI18n</code> , como \$t .	Funciones basadas en la API de traducción, como t .
Formato de Mensajes	Admite mensajes simples o con formato de parámetros.	Mantiene el formato de mensajes con parámetros.
Composición API	No tiene Composición API.	Introduce la Composición API para estructurar la lógica de internacionalización.
Soporte para TypeScript	Soporte para TypeScript.	Mejora en el soporte para TypeScript.
Carga Diferida de Mensajes	No tiene soporte nativo para carga diferida de mensajes.	Permite cargar mensajes de idiomas de forma diferida.
Mejoras de Rendimiento		Puede ofrecer mejoras en rendimiento con el uso de funciones basadas en proxies.
Soporte para Varios Idiomas	Admite la configuración de varios idiomas.	Mantiene el soporte para configuración de varios idiomas.
Documentación	(Vue I18n, 2015)	

Tabla 13: Mejora dependencias *vue-i18n*

vue-select		
Característica	Storybook Obsoleto	Storybook Actualizado
Componente de selección para Vue.js.	3.9.1	4.0.0-beta.6
Etiquetado	Si	

Filtrar / Buscar	Si	
Soporte Vuex	Si	
Soporte AJAX	Si	
Soporte de RSS	Si	
Accesible	Si	
Tamaño Total	~20kb en total / ~5kb CSS / ~15kb JS	
Selección de opciones	Selección de opciones individuales o múltiples	
Personalización	Personalizable con ranuras y variables SCSS	
dependencias	Cero dependencias	
Instalación	yarn add vue-select npm install vue-select	
Importación en Vue	import Vue from "vue"; Vue.component("v-select", vSelect);	
CSS por separado	Sí (requiere importar "vue-select/dist/vue-select.css")	
SCSS para control completo	Sí (requiere importar "vue-select/src/scss/vue-select.scss")	
Lanzamiento beta	No	Si
Documentación		(vue-select, 2022)

Tabla 14: Mejora dependencias **vue-select**

Dependencias de Desarrollo (devDependencies):	
@commitlint/cli (^9.0.1):	Herramienta de verificación de mensajes de confirmación según convenciones. (validate-commit, s.f.)
@commitlint/config-conventional (^9.0.1):	Configuración convencional para Commitlint. (validate-commit, s.f.)
@storybook/addon-a11y (^5.3.17):	Addon para mejorar la accesibilidad en Storybook. (storybook-addon-a11y, 2023)
@storybook/addon-actions (^5.3.17):	Permite a los usuarios interactuar con los componentes en Storybook. (Storybook Addon Actions, 2023)
@storybook/addon-docs (^5.3.17):	Proporciona documentación para los componentes dentro de Storybook. (Storybook Docs, 2023)
@storybook/addon-knobs (^5.3.17):	Permite ajustar los valores de los componentes en tiempo real en Storybook. (Storybook Addon Knobs, 2023)
@storybook/addon-links (^5.3.17):	Addon para establecer enlaces entre componentes en Storybook. (Story Links Addon, 2023)

@storybook/addon-notes (^5.3.17):	Permite agregar notas y documentación adicional a los componentes en Storybook. (Storybook Addon Notes, 2023)
@vue/cli-plugin-babel (~4.4.6):	Plugin de Babel para proyectos Vue CLI. (vue/cli-plugin-babel, 2022)
@vue/cli-plugin-eslint (~4.4.6):	Plugin ESLint para proyectos Vue CLI. (vue/cli-plugin-eslint, 2022)
@vue/cli-plugin-unit-jest (~4.4.6):	Plugin para la integración de Jest en proyectos Vue CLI (vue/cli-plugin-unit-jest, 2022)
@vue/cli-service (~4.4.6):	Herramienta de línea de comandos para Vue CLI. (CLI Service, s.f.)
@vue/eslint-config-airbnb (^5.0.2):	Configuración de ESLint según el estilo de Airbnb para Vue.js. (vue/eslint-config-airbnb, 2022)
@vue/test-utils (1.0.0-beta.32):	Parser de ESLint que permite el uso de características de Babel. (Introduction, s.f.)
babel-eslint (^10.1.0):	Herramienta para verificar mensajes de confirmación según convenciones específicas. (babel-eslint, 2019)
commitlint (^9.0.1):	Utilidades de prueba para Vue.js. (validate-commit, s.f.)
eslint (^6.8.0):	Herramienta de linting para JavaScript. (ESLint statically analyzes, s.f.)
eslint-plugin-import (^2.20.1):	Plugin de ESLint para las declaraciones import en JavaScript. (ESLint statically analyzes, s.f.)
eslint-plugin-vue (^6.2.2):	Plugin de ESLint específico para Vue.js. (ESLint statically analyzes, s.f.)
husky (^4.2.5):	Utilidad para gestionar ganchos (hooks) de Git. (Husky, 2019)
node-sass (^4.13.1):	Middleware para compilar archivos SASS/SCSS. (node-sass, 2023)
sass-loader (^8.0.2):	Cargador de SASS para Webpack. (sass-loader, 2019)
stylelint (^13.2.1):	Herramienta de linting para CSS. (stylelint, s.f.)
stylelint-config-twbs-bootstrap (^2.0.1):	Configuración de Stylelint basada en Bootstrap. (stylelint-config-twbs-bootstrap , 2023)
validate-branch-name (^1.0.6):	Herramienta para validar nombres de ramas de Git.
vue-cli-plugin-i18n (~1.0.1):	Plugin de internacionalización para proyectos Vue CLI.
vue-cli-plugin-storybook (~1.2.2):	Plugin para integrar Storybook en proyectos Vue CLI.
vue-template-compiler (^2.6.11):	Compilador de plantillas Vue.js.

Tabla 15: Dependencias de desarrollo

ESTUDIO DOCUMENTACIÓN DE VUE 3 EN COMPARACIÓN A VUE 2

En el caso de Vue.js, la transición de la versión 2 a la versión 3 trae consigo una serie de cambios significativos que es crucial entender para realizar una actualización exitosa. En esta sección, exploraron detalladamente las diferencias clave entre Vue 2 y Vue 3, destacando mejoras en la reactividad, nuevas características y cambios en la arquitectura. Vue 3 presenta varios cambios importantes en comparación con su predecesor, Vue 2. Entre los más destacados se encuentran:

Sistema de Reactividad Mejorado	Vue 3 introduce la Composition API, que proporciona una forma más flexible y organizada de trabajar con la lógica de los componentes.
Mejoras de Rendimiento	Se han implementado optimizaciones significativas, como el nuevo sistema de compilación que mejora la velocidad de carga y rendimiento general de las aplicaciones Vue.
Portabilidad de Componentes Mejorada	Vue 3 facilita la reutilización de componentes y su integración con otras bibliotecas gracias a su sistema más modular y flexible.

Tabla 16: Mejoras de Vue2 a Vue3

Reactividad:

En Vue 2, la reactividad se basa en opciones de componentes y objetos de datos. En contraste, Vue 3 presenta la Composition API, que permite organizar la lógica del componente de una manera más granular y estructurada. Los nuevos hooks, como `setup()`, ofrecen una mayor flexibilidad y comprensión de cómo funcionan las reactividades en comparación con las opciones de Vue 2. (Reactivity Fundamentals, s.f.)

Composition API:

La Composition API en Vue 3 permite a los desarrolladores dividir la lógica del componente en funciones más pequeñas y reutilizables. Por ejemplo, el hook setup() se utiliza para configurar el componente antes de la renderización. Esto contrasta con las opciones de componentes en Vue 2, donde la lógica puede estar distribuida en múltiples lugares, a veces dificultando el mantenimiento y comprensión del código. (Composition API FAQ, s.f.)

Portabilidad de Componentes:

Vue 3 aborda la portabilidad de componentes de manera más efectiva con su sistema de módulos mejorado. Los componentes son más independientes y modulares, facilitando su uso en diferentes contextos y proyectos. En Vue 2, la portabilidad de componentes puede ser más desafiante debido a la estructura menos modular y a las dependencias entre componentes. (Conceptos Básicos de Componentes, s.f.)

Directivas y Hooks:

Vue 3 mantiene muchas de las directivas y hooks que existen en Vue 2, pero con algunas mejoras y cambios en la sintaxis. Los desarrolladores pueden beneficiarse de nuevas directivas como v-model con capacidad de personalización mejorada. Los hooks en Vue 3, al estar vinculados a la Composition API, ofrecen una forma más clara y estructurada de manipular el ciclo de vida del componente. (Composition API: Lifecycle Hooks, s.f.)

Mejoras de Rendimiento:

Vue 3 ha implementado mejoras significativas en el rendimiento. La nueva arquitectura de reactividad y el sistema de compilación optimizado contribuyen a una mayor velocidad de carga y rendimiento general de las aplicaciones. La capacidad de tree-shaking también permite

reducir el tamaño de los bundles, mejorando la eficiencia en entornos de producción. (Mejora el rendimiento con v-once, s.f.)

Compatibilidad con Ecosistema:

Vue 3 se ha diseñado para ser compatible con el ecosistema existente de Vue 2, pero algunos cambios podrían afectar la migración directa de proyectos. Es esencial revisar y adaptar bibliotecas y herramientas de terceros para garantizar una transición sin problemas. Los equipos de desarrollo deben estar al tanto de las actualizaciones y parches de compatibilidad para garantizar una integración armoniosa. (El estado de Vue.js y su ecosistema en 2022, 2022)

Herramientas y DevTools:

Vue 3 introduce nuevas herramientas y mejoras en las DevTools que simplifican el proceso de desarrollo y depuración. Las herramientas de rendimiento y visualización de componentes han sido mejoradas, proporcionando una experiencia más robusta para los desarrolladores en comparación con las herramientas disponibles para Vue 2. (VueDevtools, s.f.)

Recursos de Migración:

Para facilitar la migración de Vue 2 a Vue 3, el equipo de Vue proporciona recursos detallados. Estos incluyen guías de migración, documentos específicos y ejemplos prácticos. Los desarrolladores deben seguir estos recursos cuidadosamente para abordar cualquier cambio en la sintaxis, arquitectura y características específicas durante la migración. (cisp, 2023)

Componente Original (Vue 2)

Una vez teniendo en cuenta estos aspectos a continuación se muestra un un ejemplo de los componentes que fue migrado dentro del storybook a Vue 3:

Propósito:

El componente MInputSearch es un campo de búsqueda con funcionalidades adicionales como iconos y cambios de estilo dinámicos. Está diseñado para admitir propiedades como value, type, placeholder, size, disabled, y icon, que permiten personalizar su apariencia y comportamiento.

Datos de Entrada:

- **value:** Texto actual del campo de búsqueda.
- **type:** Tipo del campo de entrada (predeterminado: 'text').
- **placeholder:** Texto de marcador de posición para el campo de búsqueda.
- **size:** Tamaño del campo de búsqueda (predeterminado: 'm-small').
- **disabled:** Estado de deshabilitado del campo de búsqueda (predeterminado: false).
- **icon:** Tipo de icono a mostrar junto al campo de búsqueda.

Datos de Salida:

- **isFocused:** Indica si el campo de búsqueda está enfocado.
- **isHovered:** Indica si el ratón está sobre el campo de búsqueda.
- **isDisabled:** Indica si el campo de búsqueda está deshabilitado.
- **isLoading:** Indica si el campo de búsqueda está en un estado de carga.
- **val:** Valor actual del campo de búsqueda.
- **letters:** Expresión regular para validar letras.

Migración a Vue 3

Cambios Realizados:

- Se actualizaron las importaciones de Vue y métodos específicos para Vue 3.
- Se ajustó la configuración del componente para utilizar **setup()** correctamente.
- Se modificó la creación de referencias (**ref()**) y el manejo de watchers dentro de **setup()**.
- Se reemplazaron los hooks del ciclo de vida como **onMounted** con sus equivalentes en Vue 3 dentro de **setup()**.
- Se redefinieron los métodos y eventos directamente dentro de **setup()**.
- Se utilizó el acceso directo a las propiedades y métodos proporcionados por **setup()** en lugar de **this**.

Con estos cambios, el componente ha sido migrado exitosamente a Vue 3, manteniendo su funcionalidad y adaptándose a las nuevas convenciones y características de Vue 3.

En el proceso de actualización y optimización del Storybook, se llevó a cabo una migración similar para todos los componentes existentes, siguiendo las pautas y prácticas recomendadas de Vue 3. Cada componente, como el mencionado MInputSearch, fue revisado y ajustado para aprovechar las características y mejoras introducidas en Vue 3. Durante esta migración, se actualizaron las importaciones, se reorganizó la configuración del componente para utilizar **setup()** de manera efectiva, y se adaptaron los hooks del ciclo de vida y los métodos a las nuevas convenciones de Vue 3. Este proceso aseguró la coherencia y la compatibilidad en todo el conjunto de componentes del Storybook, garantizando un entorno de desarrollo moderno y eficiente en términos de rendimiento y mantenimiento. La actualización a Vue 3 no solo proporciona beneficios en términos de rendimiento, sino que también facilita la adopción de futuras mejoras y características de la biblioteca Vue.

INTERFAZ FRAMEWORK STORYBOOK

La interfaz de Storybook que se puede observar en la

Ilustración 7 ofrece una visión detallada y amigable de los componentes de tu aplicación. Al acceder a Storybook, podrás experimentar las siguientes secciones clave:

Selector de Historias (Stories Selector):

En la parte izquierda, encontrarás el Selector de Historias, donde puedes navegar entre diferentes categorías y seleccionar las historias individuales que representan tus componentes. Esta área facilita la visualización y comparación de diferentes variantes de un componente como se puede observar en la Ilustración 2.

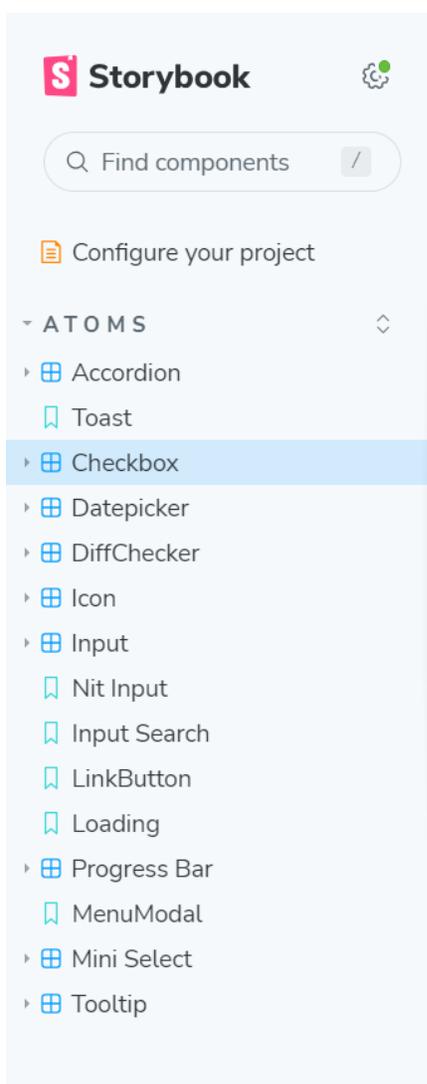


Ilustración 2: Selector de historias Storybook – Autoría propia

Área de Vista Previa (Canvas):

En el centro de la pantalla, se encuentra el Área de Vista Previa o "Canvas" que se visualiza en la

Ilustración 3. Aquí, se visualizan los componentes en diferentes estados y configuraciones. Puede actuar con los componentes directamente desde esta área para evaluar su comportamiento y apariencia.

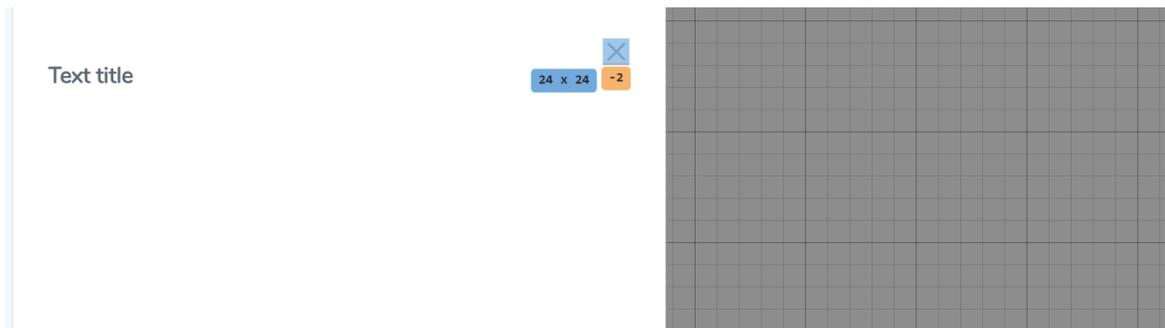


Ilustración 3: Canvas framework Storybook - Autoría propia

Barra de Herramientas (Toolbar):

En la parte superior de la interfaz, la Barra de Herramientas proporciona opciones y controles adicionales. Puedes ajustar la escala de visualización, alternar entre los modos de vista (código fuente, acciones, etc.), y realizar otras acciones específicas de Storybook desde esta área como en la **¡Error! No se encuentra el origen de la referencia..**

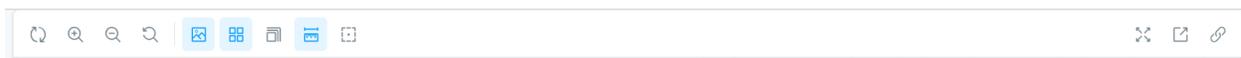


Ilustración 4: Barra de herramientas framework Storybook – Autoría propia

Panel de Controles (Addon Controls):

Una vez configurado controles de propiedades para los componentes, se encontrará el Panel de Controles en la parte inferior o lateral de la interfaz como en la Ilustración 5. Este panel te permite ajustar dinámicamente las propiedades del componente y ver cómo responde a diferentes valores.

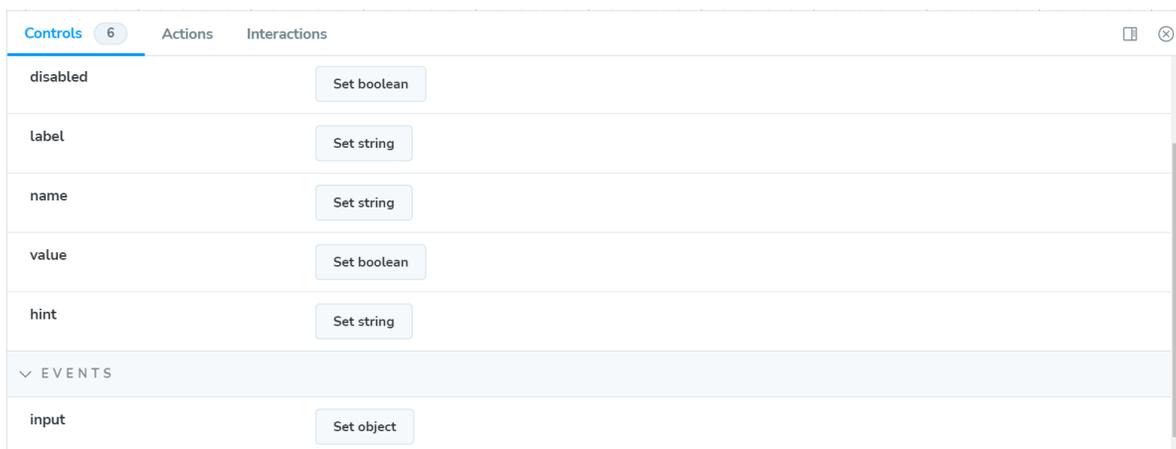


Ilustración 5: Panel de controles Storybook – Autoria Propia

Documentación Automática (Docs):

Storybook también proporciona una sección de Documentación Automática (Docs), donde puedes acceder a información detallada sobre cada componente. Esto incluye descripciones, propiedades, eventos, y ejemplos de uso, facilitando la comprensión y el uso correcto de los componentes.

Búsqueda y Filtrado:

Para una mayor eficiencia, Storybook permite buscar y filtrar componentes y sus historias. Esto es especialmente útil cuando se trabaja con un gran número de componentes o variantes.

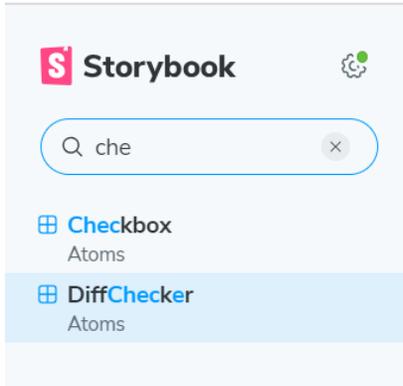


Ilustración 6: Búsqueda y filtrado Storybook – Autoría Propia

La interfaz de Storybook no solo ofrece una vista detallada de los componentes, sino que también sirve como una herramienta interactiva para el desarrollo y la colaboración en equipo como lo podemos verificar con la

Ilustración 7. Al proporcionar una representación visual y funcional de tus componentes, Storybook facilita la identificación de problemas, la iteración y la documentación efectiva.

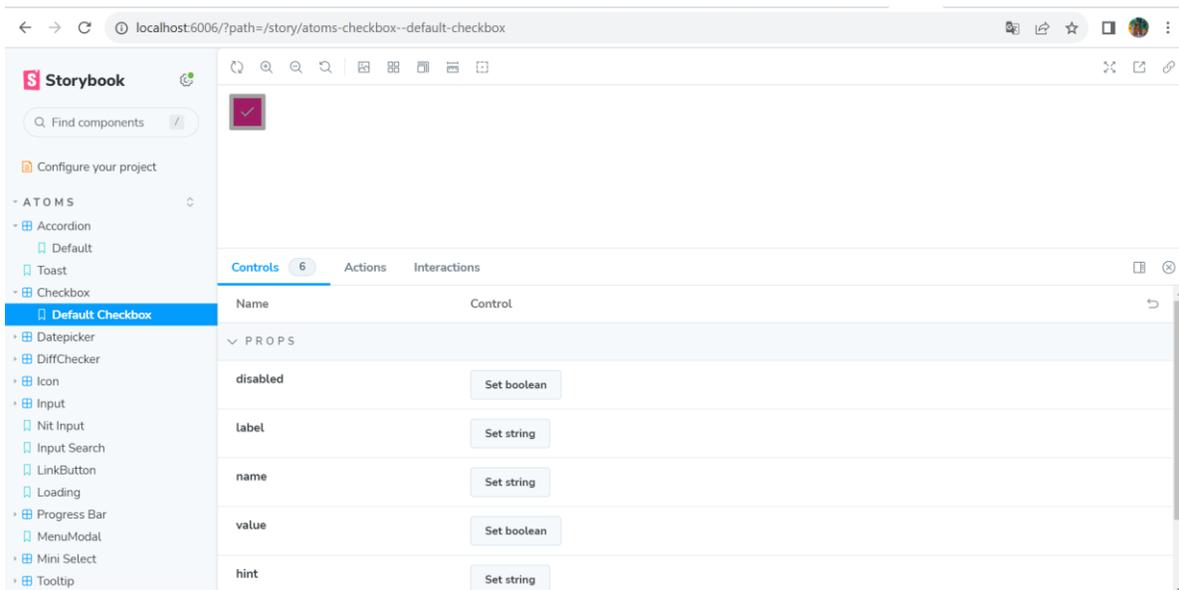


Ilustración 7: Visualización Framework Storybook – Autoría Propia

DIAGRAMA DE FLUJO STORYBOOK

Este diagrama de la Ilustración 8 representa un flujo básico de trabajo con Storybook, desde la configuración inicial hasta la integración con la aplicación principal. Aquí hay una breve descripción de cada paso:

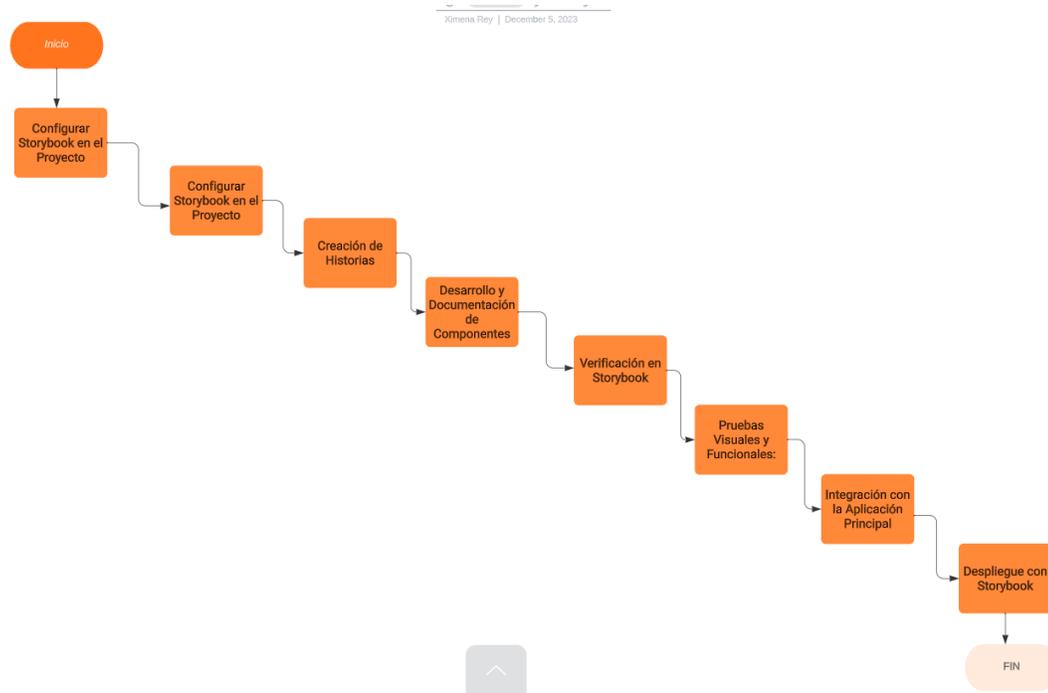


Ilustración 8: Diagrama de flujo – Autoría propia

1. **Iniciar Proyecto:** Comienza un nuevo proyecto o utiliza uno existente.
2. **Configurar Storybook en el Proyecto:** Configuración de Storybook para trabajar con el proyecto. Esto puede implicar la instalación de paquetes y la configuración del archivo de configuración de Storybook.
3. **Crear Historias para cada Componente:** Definición de historias para cada componente que deseas documentar y visualizar en Storybook.
4. **Desarrollar y Documentar los Componentes:** Desarrollo y documentación de los componentes individuales en función de las historias creadas. Esto implica escribir la

documentación y asegurarse de que las historias representen correctamente el comportamiento de los componentes.

5. **Verificar las Historias en Storybook:** Utiliza Storybook para verificar y visualizar las historias de los componentes. Esto ayuda a identificar posibles problemas y asegura que la documentación sea precisa.
6. **Realizar Pruebas Visuales y Funcionales:** Realizar pruebas visuales y funcionales en Storybook para garantizar que los componentes se comporten según lo esperado.
7. **Integrar con la Aplicación Principal:** Integración los componentes desarrollados y probados en Storybook con la aplicación principal.
8. **Desplegar la Aplicación con Storybook:** Despliegue la aplicación con Storybook integrado para que el equipo y otros interesados puedan acceder a la documentación y las historias de los componentes.

CRONOGRAMA DE ACTIVIDADES

Se realiza el uso de un diagrama de Gantt para la organización de las actividades a realizar que como se observa en la Ilustración 9, y para su visualización completa se encuentra adjunto en el siguiente enlace:

[Diagrama de Gantt](#)

Optimización del Desarrollo Frontend: Actualización y Mejora del Storybook en SBS Seguros Colombia

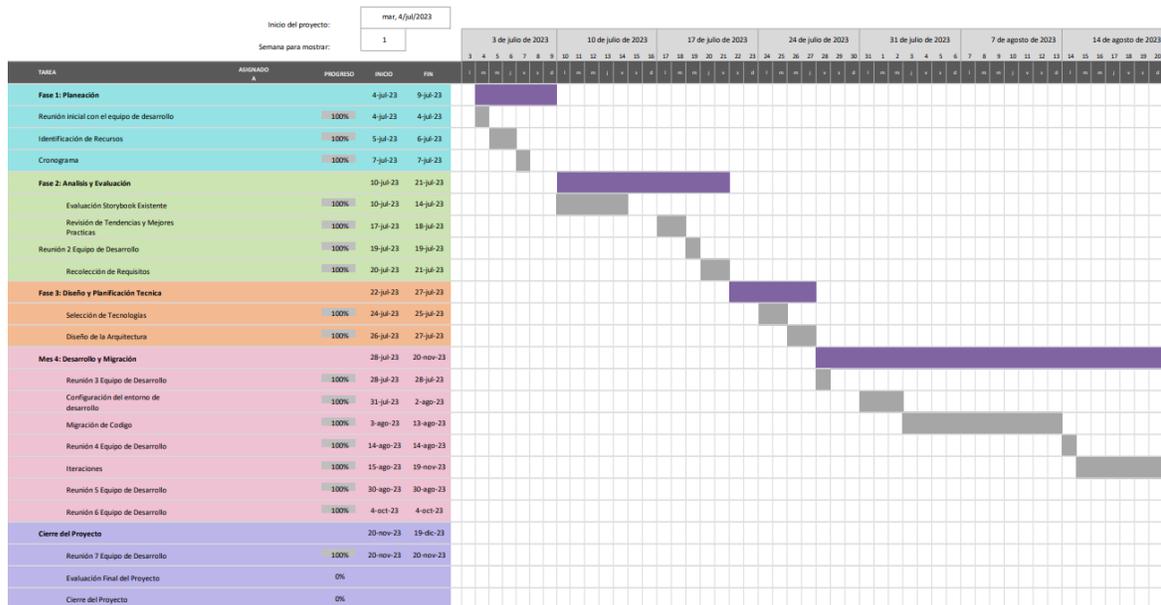


Ilustración 9: Diagrama de Gantt – Autoría propia

ENTREGABLES

- Entrega del storybook optimizado
Diciembre del 2023
- Entrega de documentación del storybook
Diciembre del 2023

CONCLUSIONES Y RECOMENDACIONES

La identificación de dependencias obsoletas y la modernización de la infraestructura han sentado las bases para un entorno de desarrollo más eficiente y seguro son avances significativos dentro del proyecto. Aunque persisten desafíos, la adaptabilidad demostrada por Storybook y la comprensión profunda de la arquitectura del sistema son activos valiosos que orientarán futuras actualizaciones y mejoras. A pesar de no alcanzar la meta completa del objetivo general al 100%, la investigación y la ejecución de la actualización de dependencias, componentes y el Storybook, se llevaron a cabo de manera oportuna.

Como recomendaciones para futuros esfuerzos, se sugiere continuar el proceso de actualización con un enfoque iterativo, abordando los desafíos restantes y aprovechando las lecciones aprendidas. Además, se recomienda establecer un plan de mantenimiento continuo que incluya la monitorización regular de dependencias y la implementación proactiva de actualizaciones menores. La colaboración estrecha entre los equipos de desarrollo y diseño, aprovechando las capacidades de Storybook, debe seguir siendo una prioridad para asegurar una implementación coherente y eficiente de los componentes de interfaz de usuario.

REFERENCIAS

blog.getbootstrap. (28 de Noviembre de 2019). *blog.getbootstrap*:
<https://blog.getbootstrap.com/2019/11/28/bootstrap-4-4-1/>

crisp, Vue.js. (3 de Marzo de 2023). Retrieved 2023, from *crisp*: <https://crisp.chat/es/blog/vue-js-migrar/>

commitlint. (s.f.). *commitlint*: <https://commitlint.js.org/#/guides-upgrade>

currency. (s.f.). <https://currency.js.org/>

David. (2022). *npmjs*. Retrieved 2023, from *npmjs*:
<https://www.npmjs.com/package/@fullhuman/postcss-purgecss>

ESCUELAVUE. (s.f.). Retrieved 2023, from *ESCUELAVUE*: <https://escuelavue.es/tips/v-once-vue-3-mejorar-rendimiento/>

ESLint. (s.f.). Retrieved 2023, from *ESLint*: <https://eslint.org/>

getbootstrap.com. (s.f.). Retrieved 2023, from [getbootstrap.com:
https://getbootstrap.com/docs/5.3/getting-started/introduction/](https://getbootstrap.com/docs/5.3/getting-started/introduction/)

jQuery write less, do more. (s.f.). jQuery write less, do more: <https://jquery.com/download/>

jsDocs.io. (2023). Retrieved 2023, from [jsDocs.io:
https://www.jsdocs.io/package/@storybook/vue#configure](https://www.jsdocs.io/package/@storybook/vue#configure)

lodash. (s.f.). Retrieved Noviembre de 2019, from [lodash: https://lodash.com/docs/](https://lodash.com/docs/)

npmJS. (s.f.). Retrieved 2023, from [npmJS: https://www.npmjs.com/package/husky/v/4.2.5](https://www.npmjs.com/package/husky/v/4.2.5)

npmJS. (2019). [npmJS: https://www.npmjs.com/package/@storybook/theming/v/5.3.5](https://www.npmjs.com/package/@storybook/theming/v/5.3.5)

npmJS. (2019). [npmJS: https://www.npmjs.com/package/core-js/v/3.6.4](https://www.npmjs.com/package/core-js/v/3.6.4)

npmJS. (2019). Retrieved 2023, from [npmJS: https://www.npmjs.com/package/date-fns/v/2.11.0](https://www.npmjs.com/package/date-fns/v/2.11.0)

npmJS. (2019). Retrieved Noviembre de 2023, from [npmJS:
https://www.npmjs.com/package/popper.js/v/1.16.1-lts](https://www.npmjs.com/package/popper.js/v/1.16.1-lts)

npmJS. (2019). Retrieved 2023, from [npmJS: https://www.npmjs.com/package/babel-eslint](https://www.npmjs.com/package/babel-eslint)

npmJS. (2019). Retrieved 2023, from [npmJS: https://www.npmjs.com/package/husky/v/4.2.5](https://www.npmjs.com/package/husky/v/4.2.5)

npmJS. (2019). Retrieved 2023, from [npmJS: https://www.npmjs.com/package/sass-
loader/v/8.0.2](https://www.npmjs.com/package/sass-loader/v/8.0.2)

npmJS. (2020). [npmJS: https://www.npmjs.com/package/@mathieustan/vue-datepicker](https://www.npmjs.com/package/@mathieustan/vue-datepicker)

npmJS. (28 de Agosto de 2020). Retrieved 2023, from [npmJS:
https://www.npmjs.com/package/@storybook/addon-viewport/v/5.3.21](https://www.npmjs.com/package/@storybook/addon-viewport/v/5.3.21)

npmJS. (28 de Agosto de 2020). Retrieved 2023, from npmJS:
<https://www.npmjs.com/package/@storybook/addon-actions/v/5.3.21>

npmJS. (2020). Retrieved 2023, from npmJS:
<https://www.npmjs.com/package/@storybook/addon-viewport/v/5.3.17>

npmJS. (2020). Retrieved 2023, from npmJS:
<https://www.npmjs.com/package/@storybook/addon-actions/v/5.3.21>

npmJS. (2022). Retrieved 2023, from npmJS: <https://www.npmjs.com/package/vue-select>

npmJS. (2022). Retrieved 2023, from npmJS: <https://www.npmjs.com/package/@vue/eslint-config-airbnb>

npmJS. (2023). npmJS: <https://www.npmjs.com/package/@storybook/addon-viewport>

npmJS. (Septiembre de 2023). Retrieved 2023, from npmJS:
<https://www.npmjs.com/package/vee-validate>

npmJS. (2023). Retrieved 2023, from npmJS:
<https://www.npmjs.com/package/@storybook/addon-ally>

npmJS. (2023). Retrieved 2023, from npmJS:
<https://www.npmjs.com/package/@storybook/addon-actions#storybook-addon-actions>

npmJS. (2023). Retrieved 2023, from npmJS:
<https://www.npmjs.com/package/@storybook/addon-docs>

npmJS. (2023). Retrieved 2023, from npmJS:
<https://www.npmjs.com/package/@storybook/addon-knobs>

npmJS. (2023). Retrieved 2023, from npmJS:
<https://www.npmjs.com/package/@storybook/addon-links#story-links-addon>

npmJS. (2023). Retrieved 2023, from npmJS:
<https://www.npmjs.com/package/@storybook/addon-notes>

npmJS. (2023). Retrieved 2023, from npmJS: <https://www.npmjs.com/package/node-sass>

npmJS. (2023). Retrieved 2023, from npmJS: <https://www.npmjs.com/package/stylelint-config-twbs-bootstrap>

Platzi. (2022). Platzi: <https://platzi.com/blog/vuejs-en-2022/>

purgecss. (29 de 10 de 2019). Retrieved 2023, from *purgecss*: <https://v1.purgecss.com/>

Shilman, M. (14 de Enero de 2020). *Storybook.js*. Storybook.js:
<https://storybook.js.org/blog/storybook-5-3/>

Shilman, M. (24 de Febrero de 2021). *Storybook*. Storybook:
<https://storybook.js.org/blog/storybook-vue3/>

Shilman, M. (2023). *Storybook*. Storybook: <https://storybook.js.org/blog/storybook-7-6/>

Storybook. (s.f.). Retrieved 2023, from Storybook: <https://storybook.js.org/docs/vue/get-started/why-storybook>

Storybook. (14 de Junio de 2020). Retrieved 2023, from Storybook:
<https://storybook.js.org/blog/storybook-5-3/>

Storybook. (2023). Retrieved 2023, from Storybook:
<https://storybook.js.org/docs/essentials/viewport>

Storybook. (2023). Storybook: <https://storybook.js.org/docs/essentials/viewport>

Storybook. (2023). Storybook: <https://storybook.js.org/docs/configure/storybook-addons>

Storybook. (2023). Retrieved 2023, from Storybook: <https://storybook.js.org/docs/configure/theming>

stylelint. (s.f.). Retrieved 2023, from stylelint: <https://stylelint.io/>

vcalendar. (s.f.). Retrieved Noviembre de 2023, from vcalendar: <https://vcalendar.io/>

Vue. (s.f.). Retrieved 2023, from Vue: <https://vuejs.org/guide/essentials/reactivity-fundamentals>

Vue. (s.f.). Retrieved 2023, from Vue: <https://vuejs.org/guide/extras/composition-api-faq.html>

Vue CLI. (s.f.). Retrieved 2023, from Vue CLI: <https://cli.vuejs.org/guide/cli-service.html>

Vue CLI. (11 de Agosto de 2022). Vue CLI: <https://cli.vuejs.org/core-plugins/babel.html#injected-webpack-chain-rules>

Vue CLI. (11 de Agosto de 2022). Retrieved 2023, from Vue CLI: <https://cli.vuejs.org/core-plugins/eslint.html#injected-webpack-chain-rules>

Vue CLI. (11 de Agosto de 2022). Retrieved 2023, from Vue CLI: <https://cli.vuejs.org/core-plugins/unit-jest.html#transform-dependencies-from-node-modules>

Vue I18n. (2015). Retrieved 2023, from Vue I18n: <https://vue-i18n.intlify.dev/>

Vue Test Utils. (s.f.). Retrieved 2023, from Vue Test Utils: <https://v1.test-utils.vuejs.org/>

Vue.js. (s.f.). Vue.js: <https://es.vuejs.org/v2/guide/>

Vue.js. (s.f.). Vue.js: <https://es.vuejs.org/v2/guide/components>

Vue.js. (s.f.). Retrieved 2023, from Vue.js: <https://vuejs.org/api/composition-api-lifecycle>

VueDevtools. (s.f.). VueDevtools: <https://devtools.vuejs.org/guide/installation.html#settings>