



**Desarrollo de librería de componentes frontend reutilizables para la eficiencia y coherencia
en proyectos de Evertec Placetopay**

Juan David Zea Acevedo

Práctica empresarial para optar al título de Ingeniero en Sistemas otorgado por la UdeA
Modalidad presencial

Asesor interno

Sandra Patricia Zabala Orrego, Ingeniera informática especialista en gerencia

Universidad de Antioquia

Facultad de ingeniería

Ingeniería de sistemas

Medellín

2024

Referencia

- [1] J. D. Zea Acevedo, "Desarrollo de librería de componentes frontend reutilizables para la eficiencia y coherencia en proyectos de Evertec Placetopay", presencial, Ingeniería de sistemas, Universidad de Antioquia, Medellín, 2024.

Estilo IEEE (2020)



Centro de Documentación de Ingeniería -CENDOI-

Repositorio Institucional: <http://bibliotecadigital.udea.edu.co>

Universidad de Antioquia - www.udea.edu.co

Rector: John Jairo Arboleda Céspedes.

Decano/Director: Julio César Saldarriaga Molina.

Jefe departamento: Danny Alejandro Múnera Ramirez.

El contenido de esta obra corresponde al derecho de expresión de los autores y no compromete el pensamiento institucional de la Universidad de Antioquia ni desata su responsabilidad frente a terceros. Los autores asumen la responsabilidad por los derechos de autor y conexos.

TABLA DE CONTENIDO

I.	INTRODUCCIÓN	8
II.	PLANTEAMIENTO DEL PROBLEMA	9
III.	JUSTIFICACIÓN.....	11
IV.	OBJETIVOS.....	12
	A. Objetivo general	12
	B. Objetivos específicos.....	12
V.	MARCO TEÓRICO	14
VI.	METODOLOGÍA	16
VII.	RESULTADOS	17
VIII.	CONCLUSIONES	18
IX.	REFERENCIAS	20

LISTA DE TABLAS

TABLA I COMPONENTES.....	17
--------------------------	----

LISTA DE FIGURAS

FIGURA I POSTER.....	19
----------------------	----

SIGLAS, ACRÓNIMOS Y ABREVIATURAS

Git	Sistema de control de versiones distribuido
npm	Node Package Manager (Gestor de Paquetes de Node.js)
Vue	Framework de JavaScript para la construcción de interfaces de usuario
CI	Continuous Integration (Integración Continua)
CD	Continuous Deployment (Despliegue Continuo)
Scrum	Marco de trabajo ágil para la gestión de proyectos
UdeA	Universidad de Antioquia

RESUMEN

El presente trabajo detalla el desarrollo de la librería de componentes frontend "Spartan" para Evertec Placetopay, con el objetivo de optimizar la eficiencia y coherencia en sus aplicaciones web. La librería fue diseñada utilizando Vue.js y publicada en npm, proporcionando una solución estandarizada para los desarrolladores de la empresa. El proyecto busca resolver problemas relacionados con la creación repetitiva de componentes y la falta de uniformidad visual en las aplicaciones. Entre los resultados más destacados se encuentra la creación y publicación de más de 50 componentes, acompañados de documentación detallada y ejemplos prácticos, disponibles para su uso.

***Palabras clave* — Librería de componentes, Frontend, Vue.js, npm, Scrum, Pruebas unitarias, GitHub Actions.**

ABSTRACT

This work details the development of the frontend component library "Spartan" for Evertec Placetopay, aimed at optimizing efficiency and consistency in their web applications. The library was designed using Vue.js and published on npm, providing a standardized solution for the company's developers. The project seeks to address issues related to the repetitive creation of components and the lack of visual uniformity in applications. Among the most notable results is the creation and publication of over 50 components, accompanied by detailed documentation and practical examples, available for use.

***Keywords* — Component library, Frontend, Vue.js, npm, Scrum, Unit tests, GitHub Actions.**

I. INTRODUCCIÓN

El presente documento aborda el desarrollo de la librería de componentes frontend "Spartan", un proyecto destinado a Evertec Placetopay, una empresa líder en la industria de tecnología financiera que ofrece soluciones innovadoras y eficientes para facilitar transacciones electrónicas, gestión de riesgos y seguridad de la información. Este desarrollo responde a la necesidad de mejorar la eficiencia y la coherencia visual en las aplicaciones web de la empresa, permitiendo a sus equipos de desarrollo optimizar su trabajo sin comprometer la calidad de la experiencia de usuario.

El objetivo principal del proyecto es crear y publicar una librería de componentes reutilizables, accesible desde npm, que estandarice el diseño y la funcionalidad de las aplicaciones. Para esta implementación se plantearon varios objetivos específicos, entre ellos el diseño de más de 40 componentes, la realización de pruebas unitarias exhaustivas, y la creación de una documentación detallada que incluya ejemplos prácticos de uso. La metodología Scrum ha sido clave en este proceso, permitiendo una planificación eficiente y un enfoque iterativo que ha facilitado la adaptación a los desafíos encontrados a lo largo del desarrollo.

II. PLANTEAMIENTO DEL PROBLEMA

En el entorno tecnológico actual, caracterizado por su dinamismo y competencia, Evertec Placetopay se enfrenta al desafío de mantenerse a la vanguardia en la provisión de soluciones tecnológicas eficientes y seguras. Para continuar ofreciendo servicios de alta calidad en el procesamiento de pagos, la gestión de riesgos y la seguridad de la información, es esencial que la empresa optimice sus procesos de desarrollo y mantenga la coherencia en la experiencia del usuario.

El desarrollo de la librería de componentes frontend “Spartan” surge como una respuesta a esta necesidad. Actualmente, los equipos de desarrollo de Evertec Placetopay enfrentan dificultades relacionadas con la eficiencia y la coherencia visual de las aplicaciones que construyen. La creación de componentes desde cero en cada proyecto no solo consume tiempo valioso, sino que también puede llevar a inconsistencias en el diseño y la funcionalidad de las aplicaciones.

Además, la falta de una librería de componentes estándar puede dificultar la colaboración entre diseñadores y desarrolladores, así como la integración de nuevas funcionalidades. Sin una solución unificada, cada proyecto puede presentar variaciones que complican el mantenimiento y la evolución del software.

El proyecto “Spartan” tiene como objetivo abordar estos problemas proporcionando una librería de componentes reutilizables, basados en una guía de estilo y diseño común, y desarrollados utilizando el framework Vue. Esta librería estará disponible en npm, permitiendo su fácil integración en cualquier proyecto de la empresa.

A. Antecedentes

Evertec Placetopay, en particular, ha experimentado problemas relacionados con la inconsistencia visual en sus aplicaciones, causados principalmente por la creación ad-hoc de componentes en cada proyecto. Este enfoque no solo incrementa el tiempo de desarrollo, sino que también afecta la uniformidad en la presentación visual de las plataformas. A nivel global,

empresas como Airbnb, Microsoft y Shopify han implementado con éxito librerías de componentes frontend que estandarizan el desarrollo de interfaces de usuario, logrando mejoras en la velocidad del desarrollo, la coherencia y la mantenibilidad de sus aplicaciones. Estas iniciativas constituyen precedentes clave para el proyecto "Spartan", que busca solucionar problemas similares dentro de Evertec Placetopay mediante la creación de una librería de componentes reutilizables, siguiendo las mejores prácticas establecidas por la comunidad de desarrollo frontend.

III. JUSTIFICACIÓN

El desarrollo de la librería de componentes frontend "Spartan" surge como una necesidad para Evertec Placetopay, una empresa que enfrenta problemas recurrentes de inconsistencia visual y duplicación de esfuerzos en la creación de componentes para sus aplicaciones web. Este proyecto se seleccionó debido a la importancia creciente de la estandarización en el desarrollo de interfaces de usuario, una tendencia que ha demostrado aumentar la eficiencia y coherencia en el diseño de sistemas web empresariales.

El principal aporte de este trabajo radica en la creación de una solución modular y reutilizable que permite a los desarrolladores de la empresa optimizar sus procesos. Al reducir la repetición innecesaria de código y garantizar que los componentes visuales sigan una misma línea estética y funcional, se mejora tanto la experiencia del usuario como la mantenibilidad de las aplicaciones. Esto no solo tiene un impacto positivo en la eficiencia operativa, sino que también puede sentar un precedente en la forma en que se desarrollan soluciones de frontend en otras áreas de la empresa.

Además, este proyecto contribuye al campo del desarrollo de software al aplicar y adaptar prácticas modernas de creación de librerías de componentes, utilizando tecnologías como Vue.js y npm, dentro del contexto específico de una empresa de pagos en línea. Esta implementación servirá como caso de estudio para futuras iniciativas de desarrollo frontend que enfrenten desafíos similares, aportando conocimiento práctico a la comunidad de desarrolladores.

IV. OBJETIVOS

A. *Objetivo general*

Publicar una versión estable de la librería “Spartan” con tests y documentación, accesible desde npm para su uso en diversos proyectos de la empresa.

B. *Objetivos específicos*

- i. Crear un set de al menos 40 componentes: Este objetivo implica el diseño y desarrollo de una variedad de componentes que cubran las necesidades comunes de las aplicaciones frontend. Estos componentes pueden incluir elementos de interfaz de usuario como botones, formularios, tarjetas, listas, entre otros. Cada componente será diseñado con atención al detalle para asegurar su funcionalidad y estética.
- ii. Realizar al menos un test para cada componente de la librería: Para garantizar la calidad y la fiabilidad de los componentes, se realizarán pruebas unitarias para cada uno de ellos. Estas pruebas verificarán que cada componente funcione como se espera en diferentes escenarios y ayudarán a identificar y corregir cualquier problema o error de manera oportuna.
- iii. Documentar la implementación de cada componente con sus propiedades: Cada componente de la librería será documentado detalladamente, incluyendo una descripción de su funcionalidad, las propiedades que acepta y cómo estas propiedades afectan el comportamiento y la apariencia del componente. Esta documentación será de gran ayuda para los desarrolladores al utilizar los componentes en sus proyectos.
- iv. Proporcionar ejemplos de uso en la documentación: Además de la documentación de las propiedades de los componentes, se proporcionarán ejemplos prácticos de cómo utilizar cada componente en una aplicación real. Estos ejemplos ayudarán a los desarrolladores a entender mejor cómo integrar y utilizar los componentes en sus propios proyectos.

- v. Implementar CI en el repositorio de código fuente: Se implementará la Integración Continua (CI) en el repositorio de código fuente del proyecto. Esto permitirá automatizar el proceso de pruebas y despliegue, asegurando que cualquier cambio en el código sea probado y validado antes de ser integrado al proyecto principal. Esto contribuirá a mantener la calidad del código y a detectar y corregir rápidamente cualquier problema o error.

V. MARCO TEÓRICO

El marco teórico de este proyecto está fundamentado en tecnologías y metodologías clave que han influido directamente en su desarrollo. Entre estas, se destacan:



Vue.js: Este framework de JavaScript ha sido la base para el desarrollo de la librería de componentes “Spartan”. Vue es conocido por su enfoque progresivo, permitiendo la integración gradual en proyectos ya existentes y facilitando la creación de interfaces de usuario reactivas y componentes reutilizables. La decisión de utilizar Vue fue clave para garantizar la modularidad y la reutilización eficiente de los componentes [1].

SonarCloud: Esta plataforma de análisis estático de código ha jugado un papel importante en la evaluación de la calidad del software desarrollado. SonarCloud permite realizar análisis automáticos para detectar errores, vulnerabilidades y code smells, proporcionando métricas claras y un coverage detallado del código fuente. En el caso de “Spartan”, el uso de SonarCloud ha permitido mantener un control riguroso sobre la calidad del código, con un enfoque en alcanzar un coverage mínimo del 80% [2].

GitHub: La integración de GitHub como plataforma de control de versiones ha facilitado el desarrollo colaborativo del proyecto, mientras que GitHub Actions ha permitido la implementación de un pipeline de Integración Continua (CI). Este pipeline automatiza las pruebas y el despliegue de nuevas versiones de la librería en npm, asegurando que cualquier cambio sea validado antes de ser integrado al proyecto principal. La automatización proporcionada por GitHub Actions ha optimizado significativamente el flujo de trabajo y la entrega continua [3].

Storybook: Para la documentación y visualización de los componentes de “Spartan”, se ha utilizado Storybook. Aunque originalmente diseñado para React, se ha adaptado con éxito para

Vue, permitiendo la creación de vistas interactivas y ejemplos de uso para cada componente. Storybook facilita a los desarrolladores la comprensión y el uso de los componentes en sus propios proyectos al proporcionar un entorno de pruebas visuales en tiempo real [4].

Vitest: Esta herramienta de testing, diseñada específicamente para proyectos de Vue.js, ha sido utilizada para la creación de pruebas unitarias. Vitest ha demostrado ser eficaz para garantizar que cada componente funcione correctamente en una variedad de escenarios, minimizando la posibilidad de errores en producción [5].

VI. METODOLOGÍA

En la metodología del proyecto “Spartan”, se siguió un enfoque iterativo y mixto, combinando principios cuantitativos y cualitativos. Se utilizó la metodología Scrum, que es un enfoque ágil para la gestión y desarrollo de proyectos. Este método permitió una ejecución organizada mediante la implementación de sprints de dos semanas y reuniones diarias. Además, el proyecto utilizó herramientas automatizadas como GitHub Actions para el despliegue continuo y pruebas, lo que permitió un análisis cuantitativo del rendimiento y la calidad del código. El enfoque cualitativo también fue clave, ya que el feedback continuo de los desarrolladores que utilizaban la librería ayudó a identificar áreas de mejora, optimizar componentes y mejorar la experiencia de usuario final. Esto se hizo posible a través de la documentación exhaustiva y el uso de herramientas como Storybook para interactuar con los componentes y proporcionar una vista previa de sus características. El proyecto aplicó una metodología mixta, combinando la recopilación y análisis de datos (pruebas unitarias, CI/CD) con una evaluación más subjetiva (retroalimentación del equipo) para garantizar que los componentes desarrollados cumplieran tanto con las especificaciones técnicas como con las expectativas de los usuarios.

VII. RESULTADOS

En esta sección se presentan los resultados obtenidos durante el desarrollo del proyecto “Spartan”. A continuación, se describen los avances más significativos en términos de creación, pruebas y publicación de componentes, así como la integración de herramientas de despliegue continuo.

El proyecto logró la creación de más de 50 componentes reutilizables, cumpliendo con el objetivo planteado de desarrollar un set de al menos 40 componentes. Entre estos se encuentran botones, formularios, tarjetas y listas, los cuales fueron diseñados siguiendo la guía de estilo de Evertec Placetopay. En la **TABLA I** se muestra un resumen de los componentes creados hasta la fecha.

TABLA I COMPONENTES

Componentes			
Accordion	Dropdown	Label	SelectBlock
Avatar	DTable	Link	Sidebar
Badge	Filter	Modal	Skeleton
Breadcrumbs	Input	ModalCard	StackedList
Button	InputAmount	ModalLeft	Steps
ButtonGroup	InputAmountBlock	PageTitle	Switch
Caption	InputBlock	Paginator	Tab
Card	InputDate	PlacetopayLogo	Table
CardBrand	InputDateBlock	Popover	TextArea
Checkbox	InputIncrement	Radio	TextAreaBlock
Combobox	InputIncrementBlock	RadioGroup	Toast
ComboboxBlock	InputMask	SectionDescription	Tooltip
DataTable	InputMaskBlock	SectionTitle	SelectBlock
DefinitionTerm	InputOTP	Select	Sidebar

VIII. CONCLUSIONES

El proyecto "Spartan" ha permitido consolidar importantes aprendizajes y resultados que impactan positivamente el desarrollo tecnológico en Evertec Placetopay. A lo largo del proceso, se evidenció la importancia de una planificación meticulosa y la aplicación rigurosa de la metodología Scrum para mantener el avance constante. Las reuniones diarias y los sprints de dos semanas permitieron una gestión eficiente del proyecto, enfrentando los desafíos de manera oportuna y eficaz.

La creación de una librería con más de 50 componentes reutilizables ha generado un impacto significativo en la optimización del tiempo de desarrollo y la coherencia visual de las aplicaciones. Además, el proceso de documentación exhaustiva, con ejemplos prácticos y propiedades claras, ha facilitado la implementación por parte de los desarrolladores.

La integración de GitHub Actions para la automatización de pruebas y despliegue ha sido clave en la estabilidad del proyecto, asegurando que cada versión lanzada cumpla con los estándares de calidad. A pesar de las dificultades iniciales en la publicación de la librería en npm y la adaptación de Storybook para Vue, se lograron resultados sólidos que reflejan el esfuerzo y la capacidad de adaptación del equipo.

El feedback recibido de los desarrolladores ha sido fundamental para la mejora continua del proyecto, optimizando la funcionalidad y usabilidad de los componentes. En definitiva, "Spartan" ha establecido una base sólida para futuros desarrollos en la empresa, garantizando componentes frontend reutilizables, coherentes y alineados con los estándares de calidad de Evertec Placetopay.

FIGURA I POSTER



IX. REFERENCIAS

- [1] "Vue.js - The Progressive JavaScript Framework | Vue.js," [Online]. Available: <https://vuejs.org/>. [Accessed 10 September 2024].
- [2] "Better Code & Better Software | Ultimate Security and Quality | Sonar," [Online]. Available: <https://www.sonarsource.com/>. [Accessed 10 September 2024].
- [3] "GitHub: Let's build from here · GitHub," [Online]. Available: <https://github.com/>. [Accessed 10 September 2024].
- [4] "Storybook: Frontend workshop for UI development," [Online]. Available: <https://storybook.js.org/>. [Accessed 10 September 2024].
- [5] "Vitest | Next Generation testing framework," [Online]. Available: <https://vitest.dev/>. [Accessed 10 September 2024].