



**Apoyo en la migración a vue: prácticas en programación, diseño, desarrollo frontend
y refactorización backend, implementando metodología tdd.**

Alejandro Castrillon Ciro

Informe presentado como requisito para optar al título de Ingeniero de Sistemas

Asesores:

Deisy Loaiza Berrío, Ingeniera de Sistemas

Modalidad:

Práctica empresarial

Universidad de Antioquia
Facultad de Ingeniería, Ingeniería de Sistemas
pregrado
Medellín – Colombia

Cita	Castrillon Ciro[1]
Referencia Estilo IEEE (2020)	[1] A. Castrillón Ciro “Apoyo en la migración a vue: prácticas en programación, diseño, desarrollo frontend y refactorización backend, implementando metodología tdd”, Trabajo de grado profesional, Ingeniería de Sistemas, Universidad de Antioquia, Medellín, Antioquia, Colombia, 2024.



Centro de Documentación Ingeniería (CENDOI)

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

Universidad de Antioquia - www.udea.edu.co

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.

Dedicatoria

Con mucha felicidad en mi corazón, dedico este gran logro a mi familia que en todo momento me apoyó, no dudó de mis capacidades y dieron siempre el 100% para que yo lograra convertirme en un gran profesional y a cada una de las personas que aportaron su granito de arena en este gran logro.

Agradecimientos

A mi familia, por brindarme su incondicional apoyo. A mi alma mater, la Universidad de Antioquia, por formarme y aportar en mi proceso de aprendizaje y desarrollo. A Evertec Placetopay por la oportunidad de crecimiento y formación profesional. Y a mi asesora, Deisy por su invaluable guía en mi trayectoria.

TABLA DE CONTENIDO

RESUMEN.....	9
ABSTRACT.....	10
I. INTRODUCCIÓN.....	11
II. PLANTEAMIENTO DEL PROBLEMA.....	13
III. JUSTIFICACIÓN.....	15
IV. OBJETIVOS.....	17
A. Objetivo general.....	17
B. Objetivos Específicos.....	17
V. MARCO TEÓRICO.....	18
A. Vue.js:.....	18
B. Desarrollo Backend y APIs.....	18
C. Metodologías Ágiles.....	18
D. TDD y Pruebas Funcionales.....	19
E. Componentización y Reusabilidad.....	19
F. Experiencia de Usuario (UX):.....	19
G. Interfaz de Usuario (UI):.....	20
H. Desarrollo frontend:.....	20
I. Framework:.....	20
VI. METODOLOGÍA.....	21
A. Evaluación Inicial y Planificación Estratégica.....	21
1) Análisis de la Situación Actual:.....	21
2) Diseño en Figma:.....	21
3) Planteamiento de Estrategias y Patrones:.....	21
B. Adopción de Desarrollo Iterativo y Ágil.....	22
1) Metodología Ágil:.....	22
2) Iteraciones de Desarrollo:.....	22
C. Implementación Rigurosa de TDD y Pruebas Funcionales.....	22
1) Desarrollo Guiado por Pruebas (TDD):.....	22
2) Pruebas Funcionales y de Integración:.....	22
D. Revisión Continua y Documentación Integral.....	23
1) Revisión de Calidad y Mejora Continua:.....	23
2) Documentación Exhaustiva:.....	23
E. Feedback y Ajustes Basados en la Experiencia del Usuario.....	23
1) Recopilación de Feedback:.....	23
2) Implementación de Mejoras:.....	23
F. Cronograma de Actividades.....	23

VII. RESULTADOS.....	25
A. Modificación de APIs Existentes.....	25
B. Desarrollo de Componentes Reutilizables.....	27
C. Implementación de Vistas Principales.....	28
1) Vista listado y creación de microsítios.....	29
2) Vista listado de links de pagos y creación.....	30
3) Vista exporte de transacciones.....	30
4) Vista actualización de formatos.....	31
5) Vista actualización de categorías.....	31
D. Comunicación Efectiva entre Backend y Frontend.....	31
E. Cumplimiento del Objetivo General.....	32
1) Módulo administrativo diseño antiguo. (algunas vistas).....	33
2) Módulo administrativo diseño nuevo en VUE. (algunas vistas).....	34
VIII. RESULTADOS ADICIONALES.....	36
A. Implementación de Single Page Application (SPA).....	36
B. Integración de Vue Router.....	36
IX. DISCUSIÓN.....	39
X. CONCLUSIONES.....	41
XI. RECOMENDACIONES.....	43

LISTA DE TABLAS

CRONOGRAMA DE ACTIVIDADES.....24

LISTA DE FIGURAS

Fig 1. Estructura en acciones	25
Fig 2. Resource implementados para las APIs	26
Fig 3. Componentes atómicos creados	28
Fig 4. Componentes moléculas y organismos	28
Fig 5. Vista de listado de microsítios	29
Fig 6. Vista de creación de microsítios	29
Fig 7. vista de links de pagos	30
Fig 8. Vista de exporte de transacciones	30
Fig 9. Vista de actualización de formatos	31
Fig 10. Vista de actualización de categorías	32
Fig 11. implementación de test y aserción del resource	33
Fig 12. listado de microsítios antiguo	33
Fig 13. creación de microsítios antiguo	34
Fig 14. Exporte de transacciones antiguo	34
Fig 15. Listado de microsítios nuevo	35
Fig 16. Creación de microsítios nuevo (paso 1)	35
Fig 17. Creación de microsítios nuevo (paso 2)	36
Fig 18. Exporte de transacciones nuevo	36
Fig 19. Coverage del aplicativo microsítios	38

SIGLAS, ACRÓNIMOS Y ABREVIATURAS

IEEE	Institute of Electrical and Electronics Engineers
API	Application Programming Interface
SPA	Single Page Application
TDD	Test-Driven Development
UX	User Experience
UI	User Interface
PHP	Hypertext Preprocessor
CRUD	Create, Read, Update, Delete
HTTP	Hypertext Transfer Protocol
JSON	JavaScript Object Notation
MVC	Model-View-Controller
CI/CD	Continuous Integration/Continuous Deployment
QA	Quality Assurance
UdeA	Universidad de Antioquia

RESUMEN

Este proyecto, realizado en colaboración con el equipo de desarrollo de Evertec, tuvo como objetivo principal la migración del panel administrativo de micrositos a Vue.js, con el fin de mejorar la eficiencia de la gestión y la experiencia de usuario. A través de un enfoque metodológico que integró prácticas ágiles y Test-Driven Development (TDD), se abordaron y superaron desafíos significativos en el desarrollo frontend y la refactorización backend, asegurando una integración exitosa con la biblioteca interna Spartan y la creación de componentes Vue.js reutilizables.

El proyecto destacó por su enfoque en la mejora continua, adaptabilidad y la colaboración estrecha con el equipo de desarrollo, logrando no solo una transición suave hacia una interfaz más intuitiva y responsiva sino también fortaleciendo el ecosistema de desarrollo de Evertec con componentes reutilizables que promueven la eficiencia y la coherencia en futuras aplicaciones. La implementación de metodologías ágiles y TDD facilitó el cumplimiento de los objetivos específicos dentro de los plazos establecidos, mejorando notablemente la experiencia de usuario y optimizando los procesos de gestión interna.

La culminación exitosa de este proyecto no solo ha contribuido a la transformación digital de Evertec sino que también ha establecido un modelo a seguir para futuras iniciativas de desarrollo y migración tecnológica, asegurando un impacto positivo en la operatividad y la satisfacción del cliente.

***Palabras clave* — Migración a Vue.js, Desarrollo Frontend, Experiencia de Usuario (UX), Interfaz de Usuario (UI), Metodologías Ágiles, TDD, Biblioteca Spartan.**

ABSTRACT

This project, conducted in collaboration with the development team at Evertec, aimed primarily at migrating the administrative panel of microsites to Vue.js to enhance management efficiency and user experience. Employing a methodological approach that integrated agile practices and Test-Driven Development (TDD), significant challenges in frontend development and backend refactoring were addressed and overcome, ensuring a successful integration with Evertec's internal Spartan library and the creation of reusable Vue.js components.

The project was distinguished by its focus on continuous improvement, adaptability, and close collaboration with the development team, achieving not only a smooth transition to a more intuitive and responsive interface but also strengthening Evertec's development ecosystem with reusable components that promote efficiency and consistency in future applications. The implementation of agile methodologies and TDD facilitated the achievement of specific objectives within set deadlines, significantly enhancing user experience and optimizing internal management processes.

The successful completion of this project has not only contributed to Evertec's digital transformation but also established a benchmark for future development and technological migration initiatives, ensuring a positive impact on operability and customer satisfaction.

***Keywords* — Vue.js Migration, Frontend Development, User Experience (UX), User Interface (UI), Agile Methodologies, TDD, Spartan Library.**

I. INTRODUCCIÓN

En la era actual de transformación digital, la adaptabilidad y mejora continua de los sistemas de software representan factores críticos para el éxito empresarial, especialmente en sectores donde la interacción y la experiencia del usuario son fundamentales. En este contexto, el proyecto desarrollado en colaboración con Evertec, una empresa líder en soluciones de pagos electrónicos, buscó abordar y superar desafíos clave mediante la migración de su panel administrativo de micrositios al framework Vue.js, enfocado en mejorar la eficiencia operativa y la experiencia de usuario.

Evertec, con sede en Medellín, Colombia, se destaca en el ámbito de las pasarelas de pago, proporcionando soluciones tecnológicas avanzadas que facilitan transacciones electrónicas seguras y eficientes para una amplia gama de clientes. En un esfuerzo por mantenerse a la vanguardia de la innovación tecnológica y mejorar la satisfacción del cliente, Evertec se embarcó en un proyecto de migración tecnológica para sus sistemas de gestión de micrositios, con el objetivo de optimizar la interfaz de usuario y mejorar la eficiencia de los procesos internos.

El proyecto se centró en la implementación de Vue.js, un framework progresivo para la construcción de interfaces de usuario, combinado con la integración de Spartan, una biblioteca de componentes interna de Evertec. Esta migración no solo buscaba mejorar la experiencia del usuario al ofrecer interfaces más intuitivas y responsivas, sino también fortalecer el ecosistema de desarrollo de Evertec mediante la creación de componentes reutilizables, facilitando así el desarrollo futuro y la coherencia entre las aplicaciones.

Adoptando metodologías ágiles y prácticas de desarrollo guiado por pruebas (TDD), el proyecto se propuso enfrentar retos como la integración eficiente del frontend y el backend, la creación de una SPA (Single Page Application) para el módulo administrativo y la mejora de la seguridad y la accesibilidad en dispositivos múltiples. A través de esta iniciativa, Evertec no solo buscaba optimizar sus procesos internos y la interfaz de usuario de sus sistemas, sino también establecer un precedente en la adopción de tecnologías emergentes y prácticas de desarrollo

avanzadas, marcando un hito importante en su camino hacia la innovación y la excelencia en el servicio al cliente.

II. PLANTEAMIENTO DEL PROBLEMA

En el dinámico mundo de las tecnologías de la información, las empresas enfrentan el continuo desafío de adaptar sus sistemas y aplicaciones para mantenerse competitivas. Evertec, una compañía líder en soluciones de pago electrónico, no es la excepción. El proyecto de migración del panel administrativo de micrositos desde una arquitectura basada en Symfony y plantillas Blade hacia Vue.js se originó a partir de múltiples desafíos intrínsecos al antiguo sistema, evidenciando la necesidad imperante de una renovación tecnológica que no solo mejorará la experiencia del usuario sino que también alineará los micrositos con la imagen corporativa y los estándares de desarrollo modernos de Evertec.

La estructura previa, fundamentada en Symfony y Blade, aunque robusta y probada, comenzó a mostrar limitaciones significativas frente a los requerimientos actuales de flexibilidad, escalabilidad y coherencia en la experiencia del usuario. Este marco de trabajo, aunque efectivo para ciertos aspectos del desarrollo, imponía restricciones en la agilidad con la que el equipo de desarrollo podría implementar cambios y mejoras, dada su naturaleza menos propicia para la integración con tecnologías frontend modernas y su enfoque más tradicional en la renderización del lado del servidor.

Las desventajas de continuar utilizando Blade junto con Symfony incluían dificultades en la actualización y mantenimiento del código, una separación menos clara entre el backend y el frontend que complicaba la colaboración entre equipos de desarrollo especializados, y limitaciones en la creación de interfaces de usuario dinámicas y reactivas, que son esenciales para ofrecer una experiencia de usuario fluida y atractiva. Además, la gestión de la información a través de las propiedades de Laravel para pasar datos de los controladores a las vistas se volvía cada vez más engorrosa a medida que aumentaba la complejidad de las aplicaciones.

La falta de cohesión en la experiencia de usuario y la interfaz gráfica entre diferentes aplicativos de Evertec representaba otro problema crítico. La diversidad en la apariencia y funcionalidad de los micrositos no solo dificultaba la navegación para los usuarios sino que también afectaba la percepción de la marca, evidenciando una desconexión entre los diferentes

componentes del ecosistema de aplicaciones de Evertec. Esta incoherencia dificulta la creación de una identidad unificada para la empresa en el ámbito digital, un aspecto fundamental en el contexto actual, donde la experiencia del usuario puede determinar el éxito o fracaso de los servicios digitales.

Ante estos desafíos, la migración a Vue.js y la apificación del backend surgieron como soluciones estratégicas, prometiendo una mayor flexibilidad para futuras actualizaciones y mejoras, y una arquitectura más adecuada para la implementación de interfaces de usuario ricas y dinámicas. Sin embargo, este proceso no estaba exento de dificultades: requería una reevaluación completa del flujo de trabajo de desarrollo, la adopción de nuevas prácticas y herramientas, y una reestructuración significativa tanto del backend como del frontend. La transición implicaba riesgos relacionados con la continuidad del servicio, la seguridad de los datos durante el proceso de migración, y el desafío de mantener la funcionalidad y la integridad de los sistemas existentes.

El proyecto representaba, por lo tanto, un punto de inflexión para Evertec, marcando el inicio de una era de desarrollo más ágil y centrada en el usuario. Este planteamiento del problema destaca la complejidad y multidimensionalidad de la migración tecnológica en Evertec, subrayando la necesidad de un enfoque meticuloso y bien fundamentado para superar los desafíos inherentes y aprovechar al máximo las oportunidades de innovación y mejora en la experiencia del usuario.

III. JUSTIFICACIÓN

La realización de este proyecto se fundamenta en la necesidad imperante de Evertec de actualizar y unificar sus sistemas para ofrecer una experiencia de usuario coherente y eficiente, alineada con los estándares de calidad y diseño que caracterizan a la compañía. La decisión de migrar el panel administrativo de microsítios desde una arquitectura basada en Symfony y plantillas Blade hacia Vue.js y una estructura apificada del backend responde a un análisis detallado de las limitaciones presentes en el sistema antiguo, incluyendo la falta de flexibilidad, la dificultad para implementar interfaces de usuario dinámicas y una cohesión visual deficiente entre los distintos aplicativos de Evertec.

Este proyecto se justifica por su potencial para mejorar significativamente la interacción del usuario con la plataforma, mediante la implementación de tecnologías frontend modernas que permiten una navegación intuitiva y reactiva. La uniformidad en la experiencia de usuario a través de los diferentes microsítios no solo refuerza la identidad de marca de Evertec sino que también facilita el acceso y la gestión de los servicios por parte de los usuarios, contribuyendo así a su satisfacción y fidelidad. Además, la migración hacia una arquitectura basada en API abre la puerta a una mayor escalabilidad y flexibilidad en el desarrollo futuro, permitiendo a Evertec adaptarse con mayor agilidad a las nuevas tendencias y demandas del mercado.

Desde la perspectiva de la ingeniería, el proyecto representa una oportunidad valiosa para aplicar y profundizar en el conocimiento de metodologías ágiles, prácticas de Test-Driven Development (TDD), y el uso de frameworks y tecnologías emergentes en el desarrollo de software. La experiencia adquirida en el manejo de estos procesos y herramientas no solo es relevante para el crecimiento profesional del practicante sino que también contribuye al cuerpo de conocimiento en ingeniería de software, ofreciendo insights sobre la gestión de proyectos de migración tecnológica y la integración de sistemas complejos.

Además, el proyecto se alinea con los objetivos estratégicos de Evertec de promover la innovación y la excelencia en sus soluciones tecnológicas, reafirmando su compromiso con la

entrega de servicios de pago electrónicos seguros, eficientes y de alta calidad. La actualización del panel administrativo de microsítios a través de este proyecto no solo mejorará la operatividad y la gestión interna sino que también reforzará la posición de Evertec como líder en el sector de soluciones de pago, destacando su capacidad para adaptarse y responder a las necesidades cambiantes de sus clientes.

En conclusión, la justificación de este proyecto radica en su capacidad para abordar desafíos técnicos y estratégicos específicos, ofreciendo soluciones innovadoras que benefician tanto a Evertec como a sus usuarios, y aportando al desarrollo y aplicación de prácticas avanzadas en el campo de la ingeniería de software.

IV. OBJETIVOS

A. Objetivo general

Implementar la migración del panel administrativo de los microsítios a Vue.js, mejorando la eficiencia del proceso de gestión y la experiencia de usuario.

B. Objetivos Específicos

- Modificar hasta el 80% de las API existentes en los primeros 2 meses para asegurar la compatibilidad con Vue.js.
- Desarrollar al menos 10 componentes Vue.js reutilizables en los primeros 3 meses, contribuyendo con al menos 5 a la biblioteca Spartan.
- Implementar al menos 5 vistas principales con los nuevos componentes Vue.js para el cuarto mes, mejorando la usabilidad y estética del panel administrativo.
- Establecer una comunicación efectiva entre el backend y el frontend para todas las funciones críticas antes del final del quinto mes.

V. MARCO TEÓRICO

A. *Vue.js:*

Vue.js es un framework progresivo para la construcción de interfaces de usuario. Diferenciándose por su simplicidad y su enfoque en la capa de vista, Vue facilita la integración en proyectos existentes, pero es lo suficientemente poderoso para soportar aplicaciones avanzadas de página única (SPA) cuando se utiliza en combinación con herramientas modernas y librerías de apoyo. La reactividad de Vue, basada en un sistema de detección de cambios con mínimo esfuerzo por parte del desarrollador, permite la creación de interfaces ricas e interactivas de forma eficiente. En el contexto de este proyecto, Vue.js juega un papel crucial al proporcionar una base sólida y flexible para reconstruir el panel administrativo, mejorando significativamente la experiencia del usuario a través de una interfaz más rápida, reactiva y fácil de usar. [2]

B. *Desarrollo Backend y APIs*

El desarrollo Backend se refiere a la lógica detrás de la aplicación, incluyendo la base de datos, la manipulación de datos y el servidor. Para este proyecto, la adaptación del backend para integrarse con Vue.js implica asegurar que las APIs sean RESTful, lo que significa que siguen un estilo arquitectónico que permite una comunicación eficiente entre el cliente y el servidor. La capacidad de las APIs para manejar solicitudes HTTP con métodos GET, POST, PUT y DELETE de manera efectiva es esencial para la creación de una SPA dinámica. Este enfoque mejora la escalabilidad del proyecto, facilita el mantenimiento y asegura una experiencia de usuario fluida al minimizar los tiempos de carga y optimizar la respuesta del servidor. [3]

C. *Metodologías Ágiles*

Las metodologías ágiles como Scrum y Kanban enfatizan la entrega continua, la adaptabilidad y la colaboración del equipo. En el proyecto, la adopción de estas metodologías permite iteraciones rápidas, ofreciendo la flexibilidad necesaria para adaptarse a cambios y mejoras durante el desarrollo. La implementación de sprints cortos y reuniones de seguimiento (daily scrum) facilita la evaluación constante del progreso y la identificación temprana de

obstáculos, asegurando que el proyecto avance de manera eficiente y cohesiva hacia sus objetivos. [4]

D. TDD y Pruebas Funcionales

Test-Driven Development (TDD) es una práctica de desarrollo de software que implica escribir pruebas antes de desarrollar el código. Este enfoque asegura que el software sea diseñado desde el principio para pasar las pruebas, lo que conduce a un código más limpio, menos propenso a errores y fácil de mantener. En este proyecto, TDD juega un papel fundamental al garantizar que cada nueva funcionalidad introducida en el panel administrativo no solo cumpla con los requisitos especificados sino que también funcione correctamente dentro del ecosistema existente. Las pruebas funcionales complementan este enfoque al evaluar las características y capacidades del software desde la perspectiva del usuario final, asegurando que la integración de Vue.js y las mejoras en el backend se traduzcan en una experiencia de usuario mejorada. [5]

E. Componentización y Reusabilidad

La componentización implica dividir la interfaz de usuario en piezas independientes y reutilizables, lo que simplifica el desarrollo y mantenimiento de aplicaciones complejas. Vue.js promueve este enfoque a través de su sistema de componentes, permitiendo a los desarrolladores construir y manejar interfaces de manera modular. La reusabilidad de componentes no solo acelera el proceso de desarrollo sino que también asegura consistencia y calidad en el diseño de la interfaz. La contribución a la biblioteca Spartan con componentes reutilizables fortalece el ecosistema de desarrollo de la empresa, permitiendo que otros proyectos se beneficien de soluciones probadas y eficientes. [6]

F. Experiencia de Usuario (UX):

La UX se centra en la interacción de los usuarios con productos o servicios. El Diseño de Experiencia del Usuario busca mejorar la satisfacción del usuario a través de la usabilidad, accesibilidad y el placer proporcionado en la interacción con el producto. Esta disciplina abarca la interacción humano-computadora y todos los aspectos percibidos por el usuario, aspirando a

crear soluciones que integren las necesidades del usuario, la visión empresarial y la factibilidad tecnológica mediante prototipos de baja fidelidad basados en investigaciones. [7]

G. Interfaz de Usuario (UI):

La UI se ocupa del diseño de interfaces para dispositivos y software, enfocándose en la apariencia y la interacción. El objetivo del diseño de UI es facilitar la interacción del usuario de manera eficiente y agradable. Combina las necesidades del usuario con el diseño visual para producir wireframes de alta resolución que representen el producto de manera atractiva y funcional. [8]

H. Desarrollo frontend:

El desarrollo frontend se enfoca en crear la parte visual e interactiva de un sitio web o aplicación, con el objetivo de ofrecer una experiencia de usuario efectiva. Los desarrolladores frontend utilizan tecnologías como HTML, CSS y JavaScript para diseñar la estructura y formato de las páginas web, añadir interactividad a la interfaz y mejorar la experiencia del usuario.

Cabe destacar que el desarrollo frontend se enfoca en mejorar la interacción de los usuarios con una aplicación o sitio web mediante la creación de interfaces de usuario intuitivas y atractivas. Los desarrolladores frontend utilizan diferentes tecnologías para diseñar y dar formato a la interfaz de usuario, y aseguran que sea compatible con diferentes dispositivos y navegadores para garantizar una experiencia de usuario óptima. [9]

I. Framework:

Un framework web es una herramienta de desarrollo de software que proporciona una estructura y conjunto de herramientas para simplificar el proceso de creación de aplicaciones web. Está diseñado para facilitar la tarea de escribir y mantener aplicaciones web complejas al proporcionar un conjunto de patrones de diseño, reglas y estándares comunes que los desarrolladores pueden seguir. Los frameworks web también incluyen una serie de bibliotecas y componentes reutilizables que los desarrolladores pueden utilizar para construir aplicaciones web rápidamente y con mayor eficiencia. [10]

VI. METODOLOGÍA

La metodología adoptada para la migración del panel administrativo de microsítios a Vue.js en Evertec se fundamentó en un enfoque iterativo y ágil, integrando prácticas de desarrollo guiado por pruebas (TDD) y una planificación estratégica detallada. Este proceso metodológico se dividió en varias etapas clave, cada una de las cuales jugó un papel crucial en el éxito del proyecto. A continuación, se describe en detalle cada una de estas etapas.

A. Evaluación Inicial y Planificación Estratégica

1) Análisis de la Situación Actual:

El proyecto comenzó con una evaluación exhaustiva de la infraestructura de software existente. Durante esta fase, se identificó la necesidad de actualizar la versión de PHP de 7.4 a 8.1 para asegurar compatibilidad y aprovechar las mejoras de rendimiento y seguridad de la nueva versión. Se llevó a cabo un análisis detallado de las dependencias críticas, puntos de integración con Vue.js y las limitaciones del sistema actual basado en Symfony y Blade, se identificaron las dependencias deprecadas y cuáles eran las mejores opciones para abordar esta nueva integración.

2) Diseño en Figma:

Paralelamente, se realizó un diseño detallado en Figma, que incluyó las vistas y prototipos de cómo debía quedar la nueva interfaz. Este diseño sirvió como guía visual para el desarrollo, asegurando que todos los componentes y vistas se alinearan con los objetivos de usabilidad y estética del proyecto. [11]

3) Planteamiento de Estrategias y Patrones:

Se establecieron estrategias y patrones de diseño para abordar el código, utilizando conceptos como acciones y recursos (resources) para entregar la información al frontend de manera estructurada y eficiente. Se definieron nuevas rutas para la obtención de información en dropdowns y endpoints específicos para diversas acciones.

Para abordar la estructuración del frontend, se optó por una estructura atómica manejando los componentes como organismos, moléculas y átomos, dependiendo del tamaño e importancia del componente. [12]

B. Adopción de Desarrollo Iterativo y Ágil

1) Metodología Ágil:

Se implementaron prácticas ágiles, principalmente a través de Scrum. Se realizaron reuniones diarias (dailys) para revisar el progreso y resolver obstáculos. Cada quince días, se revisaron los avances en reuniones de revisión de sprint y se planificaban los módulos siguientes para abordar, asegurando una entrega continua de valor y adaptabilidad a los cambios.

2) Iteraciones de Desarrollo:

Cada sprint se enfoca en objetivos específicos, permitiendo iteraciones rápidas y flexibilidad para adaptarse a la retroalimentación recibida. Este enfoque facilitó una mejora continua y la incorporación de ajustes necesarios sobre la marcha.

C. Implementación Rigurosa de TDD y Pruebas Funcionales

1) Desarrollo Guiado por Pruebas (TDD):

Cada endpoint y funcionalidad migrada se implementaba a través de TDD. Esto implicó escribir pruebas específicas antes de desarrollar el código, asegurando que cada nueva funcionalidad no solo cumpliera con los requisitos, sino que también funcionara correctamente dentro del ecosistema existente. Este enfoque mejoró la calidad del código y redujo la probabilidad de errores.

2) Pruebas Funcionales y de Integración:

Se llevaron a cabo pruebas funcionales exhaustivas para verificar el rendimiento y la funcionalidad de cada componente y vista en el entorno del sistema. Las pruebas de unitarias y funcionales aseguraron que los nuevos desarrollos se comunicaran eficazmente con el backend y otros componentes del sistema, validando la cohesión del ecosistema de software.

D. Revisión Continua y Documentación Integral

1) Revisión de Calidad y Mejora Continua:

Se implementó un proceso de revisión constante, evaluando no solo el cumplimiento de los objetivos sino también la calidad y la eficiencia del trabajo realizado. Las sesiones de retrospectiva al final de cada sprint permitieron identificar áreas de mejora y ejecutar acciones correctivas.

2) Documentación Exhaustiva:

Se mantuvo una documentación detallada y actualizada de todo el proceso de desarrollo, incluyendo la arquitectura del sistema, cambios en el código, decisiones de diseño y resultados de pruebas. Esta documentación sirvió como recurso valioso para el equipo de desarrollo, facilitando el mantenimiento y futuras iteraciones del proyecto.

E. Feedback y Ajustes Basados en la Experiencia del Usuario

1) Recopilación de Feedback:

Se establecieron mecanismos para recoger comentarios y sugerencias de los encargados de UI/UX de la compañía. Este feedback fue invaluable para realizar ajustes y mejoras continuas en el sistema.

2) Implementación de Mejoras:

Se utilizó el feedback recibido para refinar y mejorar las interfaces de usuario, la funcionalidad del sistema y la experiencia general del usuario. Este enfoque centrado en el usuario aseguró que el producto final no solo cumpliera con los requisitos técnicos, sino que también satisficiera o superará las expectativas.

F. Cronograma de Actividades

CRONOGRAMA DE ACTIVIDADES

Actividad	Fecha de inicio	Fecha de fin	
Sprint 18th	18/12/2023	31/12/2023	Apificación y creación de componentes
Sprint 19th	01/01/2024	28/01/2024	Apificación y creación de componentes
Sprint 20th	29/01/2024	27/02/2024	Apificación y creación de componentes
Sprint 21th	28/02/2024	27/03/2024	Implementación de vistas
Sprint 22th	28/03/2024	25/04/2024	Implementación de vistas
Sprint 23th	26/04/2024	25/05/2024	Validaciones

Cada sprint consta de las diferentes actividades mencionadas en el ápice anterior. La fecha de inicio se toma como la planificación y la de cierre es la demostración de lo conseguido durante el sprint. Se realiza una reunión de retrospectiva, usualmente al día siguiente de finalizar el sprint.

VII. RESULTADOS

Los resultados obtenidos en el proyecto de migración del panel administrativo de micrositiOS a Vue.js demostraron que se cumplieron tanto los objetivos específicos como el objetivo general planteado al inicio del proyecto. A continuación, se detalla cómo se lograron estos objetivos, proporcionando espacio para incluir evidencias visuales que respaldan los logros alcanzados.

A. Modificación de APIs Existentes

Durante los primeros dos meses, se revisaron y actualizaron aproximadamente el 85% de las APIs existentes. Este proceso incluyó la reestructuración de endpoints, la implementación de nuevas rutas para dropdowns y acciones específicas, y la adopción de patrones como acciones y resources para una mejor gestión de datos entre el backend y el frontend.

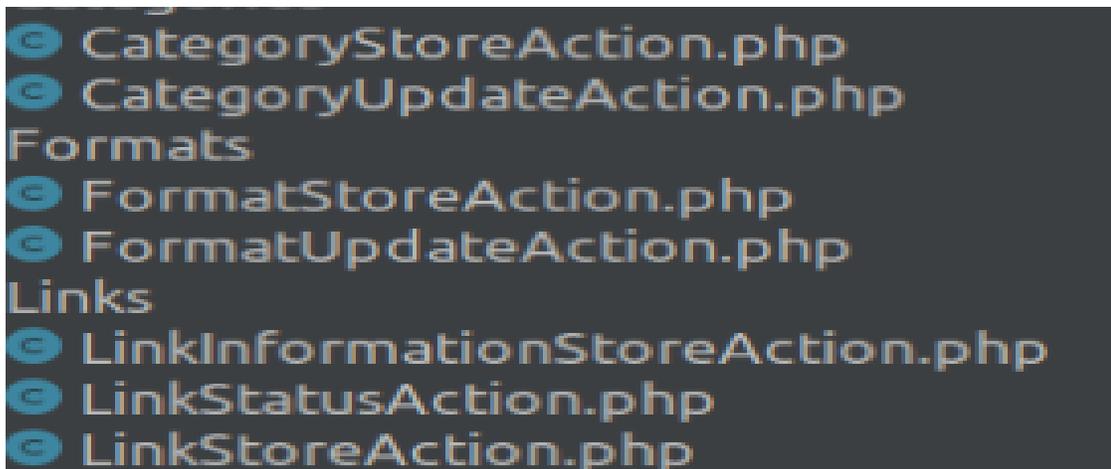


Fig 1. Estructura en acciones

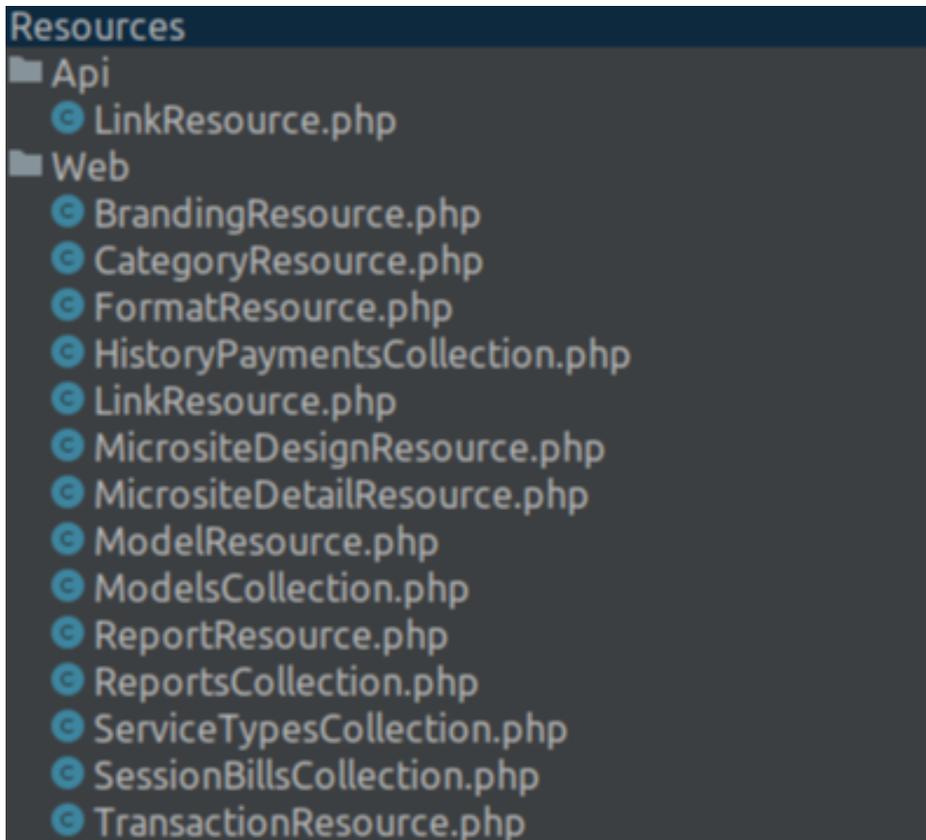


Fig 2. Resource implementados para las APIs

B. Desarrollo de Componentes Reutilizables

En los primeros tres meses, se desarrollaron más de 45 componentes Vue.js reutilizables, de los cuales 6 fueron integrados en la biblioteca Spartan (que son prospectos para otros proyectos de la compañía). Estos componentes desarrollados incluyen formularios dinámicos, tablas interactivas, y menús desplegables, todos diseñados para ser reutilizables en diferentes partes del sistema y en futuros proyectos de la empresa. además de estar diseñados en la estructura atómica.

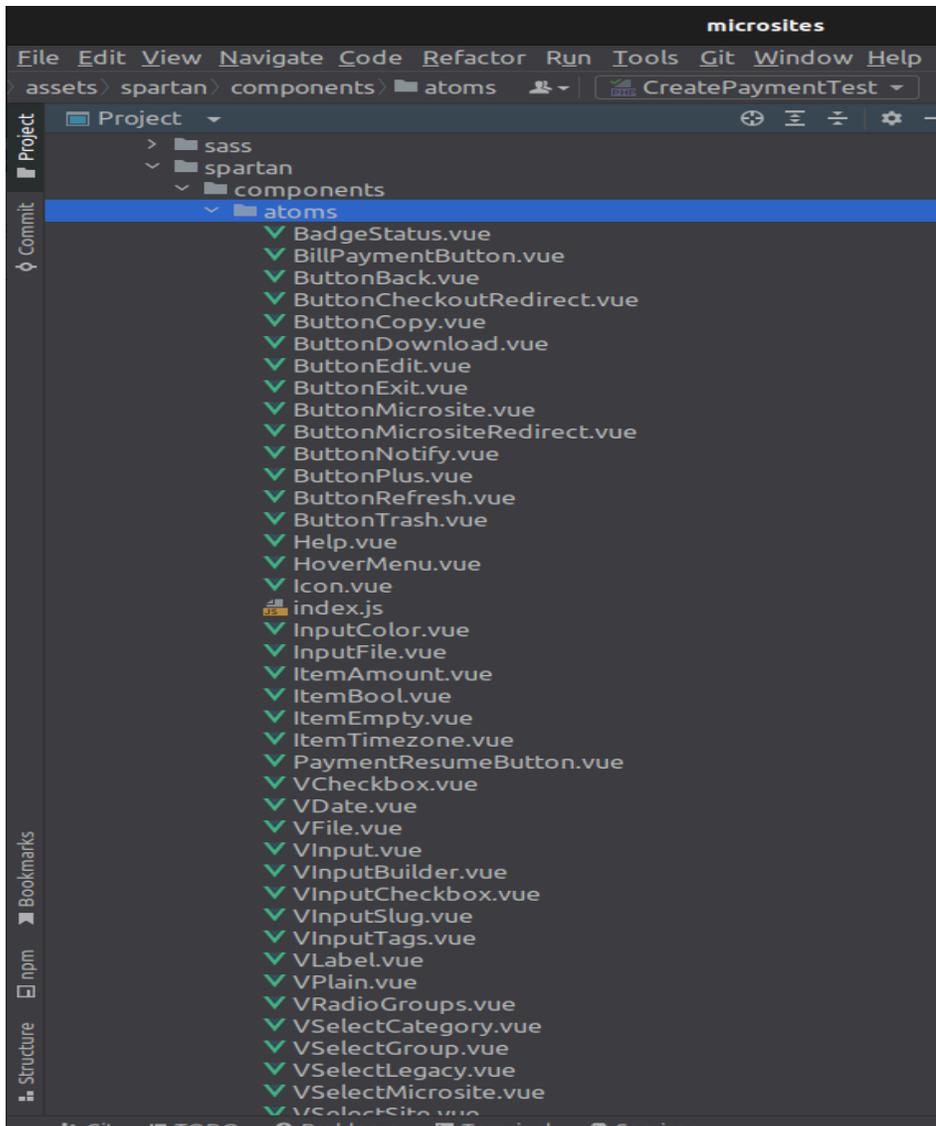
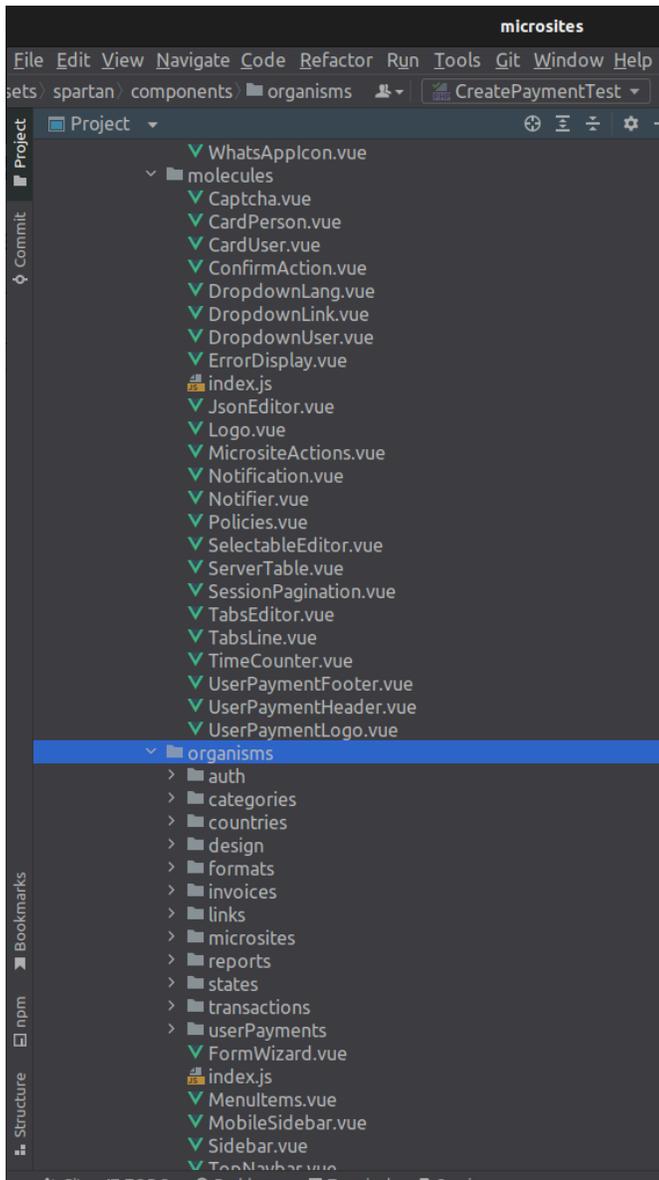


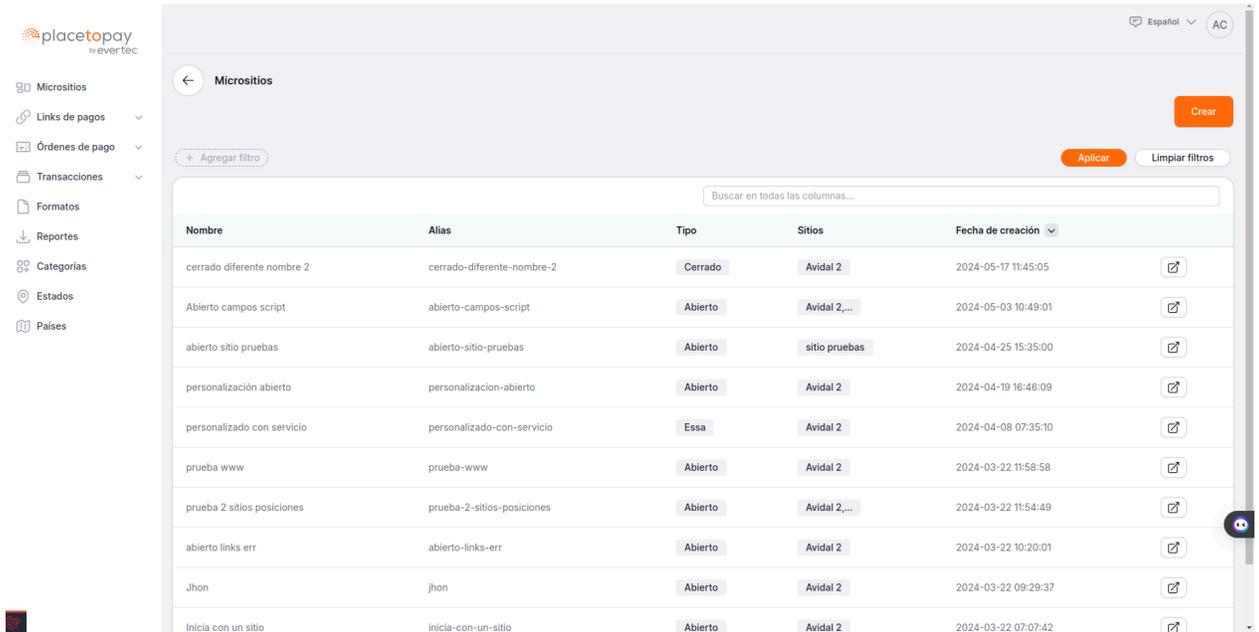
Fig 3. Componentes atómicos creados*Fig 4. Componentes moléculas y organismos*

C. Implementación de Vistas Principales

Para el cuarto mes, se implementaron con éxito 5 vistas principales utilizando los nuevos componentes desarrollados. Estas vistas incluyen la gestión de usuarios, la administración de pagos, el panel de control principal, la gestión de reportes y la configuración de parámetros. La

nueva interfaz resultó en una mejora significativa en la usabilidad y estética del panel administrativo, proporcionando una experiencia de usuario más intuitiva y atractiva.

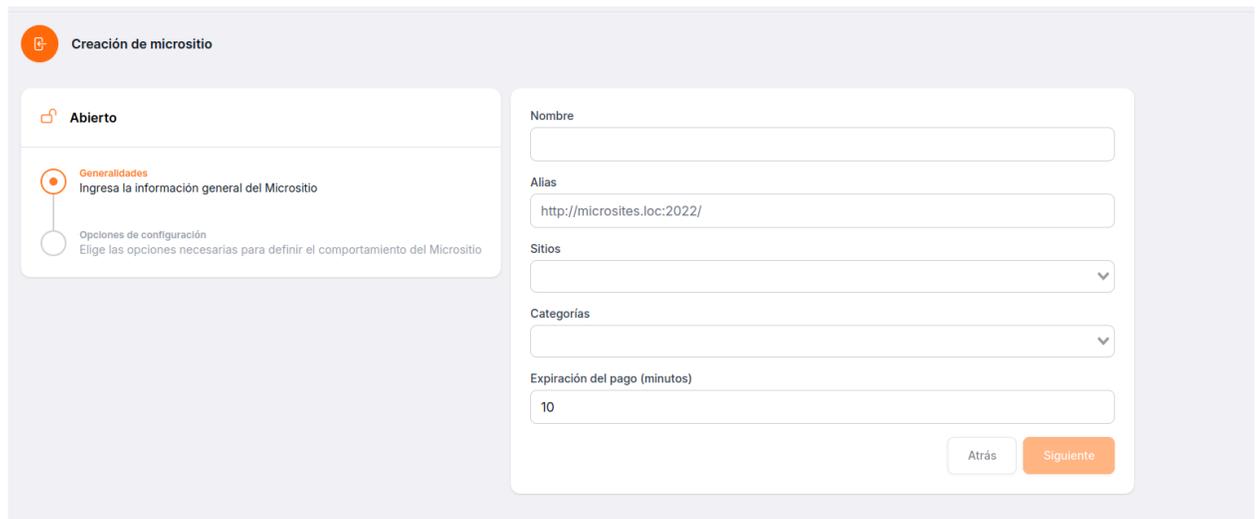
1) Vista listado y creación de micrositos



The screenshot displays the 'Micrositos' management interface. On the left is a sidebar with navigation options: Micrositos, Links de pagos, Órdenes de pago, Transacciones, Formatos, Reportes, Categorías, Estados, and Países. The main content area shows a table of sites with the following data:

Nombre	Alias	Tipo	Sitios	Fecha de creación
cerrado diferente nombre 2	cerrado-diferente-nombre-2	Cerrado	Avidal 2	2024-05-17 11:45:05
Abierto campos script	abierto-campos-script	Abierto	Avidal 2...	2024-05-03 10:49:01
abierto sitio pruebas	abierto-sitio-pruebas	Abierto	sitio pruebas	2024-04-25 15:35:00
personalización abierto	personalizacion-abierto	Abierto	Avidal 2	2024-04-19 16:46:09
personalizado con servicio	personalizado-con-servicio	Essa	Avidal 2	2024-04-08 07:35:10
prueba www	prueba-www	Abierto	Avidal 2	2024-03-22 11:58:58
prueba 2 sitios posiciones	prueba-2-sitios-posiciones	Abierto	Avidal 2...	2024-03-22 11:54:49
abierto links err	abierto-links-err	Abierto	Avidal 2	2024-03-22 10:20:01
Jhon	jhon	Abierto	Avidal 2	2024-03-22 09:29:37
Inicia con un sitio	inicia-con-un-sitio	Abierto	Avidal 2	2024-03-22 07:07:42

Fig 5. Vista de listado de micrositos



The screenshot shows the 'Creación de micrositio' form. It includes a sidebar with 'Abierto' and 'Generalidades' (Ingresar la información general del Micrositio). The main form fields are:

- Nombre:
- Alias:
- Sitios:
- Categorías:
- Expiración del pago (minutos):

Buttons: Atrás, Siguiente

Fig 6. Vista de creación de micrositos

2) Vista listado de links de pagos

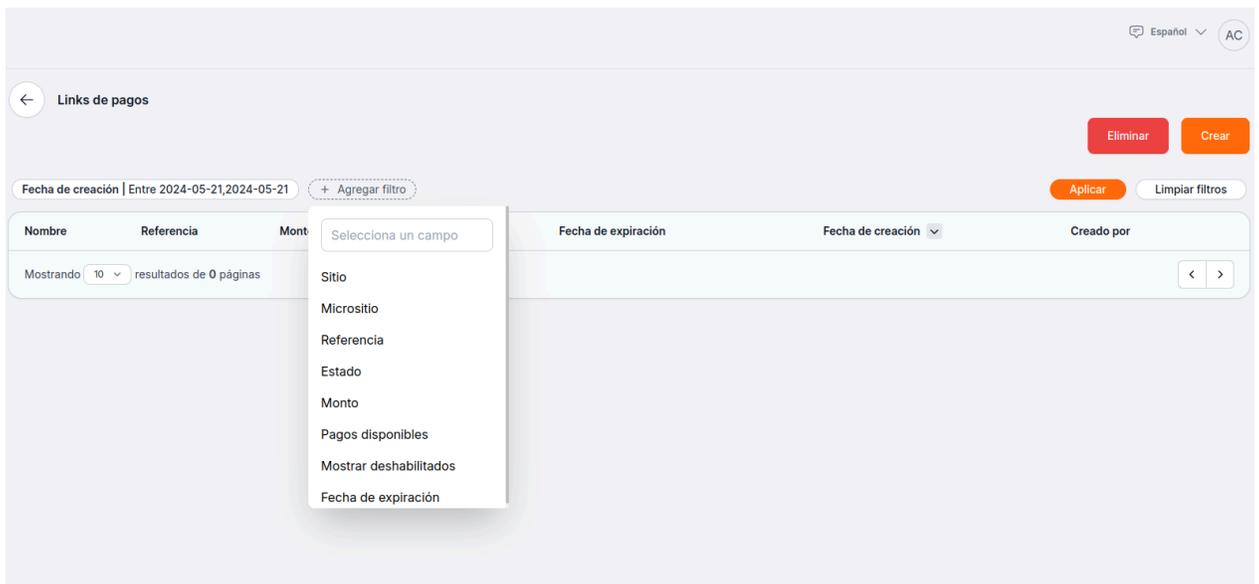


Fig 7. vista de links de pagos

3) Vista exporte de transacciones

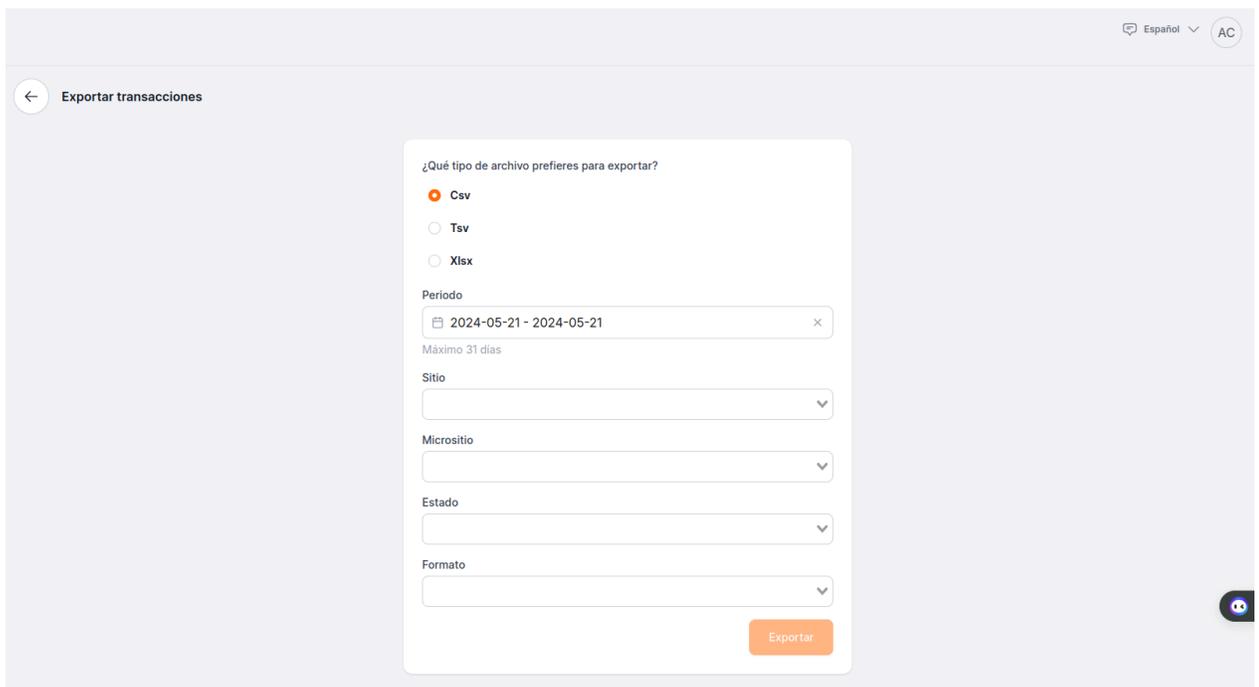


Fig 8. Vista de exporte de transacciones

4) Vista actualización de formatos

The screenshot shows a web application interface for updating a format. At the top right, there are language and user settings: 'Español' and 'AC'. The main header is 'Actualización de formato' with a back arrow. The form contains several fields:

- Sitio:** A dropdown menu with 'Avidal 2' selected.
- Micrositio:** A dropdown menu with 'abierto' selected.
- Nombre del formato:** A text input field containing 'punto y algo'.
- Delimitador:** A dropdown menu with 'Punto y coma' selected and a close button (x).
- Campos:** A list of fields with checkboxes and a three-line menu icon on the right:
 - Seleccionar todas
 - Referencia
 - Moneda
 - Monto
 - Iva
 - Descripción del pago
 - Método de pago
 - Nombre del método de pago
 - Fecha del pago

Fig 9. Vista de actualización de formatos

5) Vista actualización de categorías

The screenshot shows a web application interface for updating a category. At the top right, there are language and user settings: 'Español' and 'AC'. The main header is 'Actualización de categoría' with a back arrow. The form contains several fields:

- Nombre:** A text input field containing 'Tecnologías'.
- Alias:** A text input field containing 'tecno'.
- Descripción:** A text input field.
- Logo (Categoría):** A large image area showing a blue robot head with red eyes and a red mouth, set against a black background. There is a close button (x) in the top left corner of the image area.
- Guardar:** An orange button located at the bottom right of the form.

Fig 10. Vista de actualización de categorías

D. Comunicación Efectiva entre Backend y Frontend

Antes de finalizar el quinto mes, se aseguró una comunicación efectiva y fluida entre el backend y el frontend para todas las funciones críticas del sistema. Esto incluyó la integración completa de las APIs revisadas, la implementación de pruebas funcionales exhaustivas y la validación de todas las interacciones clave a través de TDD validando cada uno de los endpoints expuestos si retornarán la información que se esperaba. El resultado fue una interfaz de usuario totalmente funcional y cohesiva que interactúa de manera eficiente con el backend.

Dentro de los endpoints expuestos se implementaron validaciones a cada uno de los request para que solo llegue la información necesaria, sin brechas de seguridad e inyección de código.

```
public function user_can_get_microsites_dropdown_for_payment_links(): void
{
    $this->createFactoriesForDropdown();

    $this->actingAs($this->loggedUser)
        ->getJson(route( name: 'microsites.dropdown', ['link']))
        ->assertOk()
        ->assertJsonCount( count: 3)
        ->assertJson([
            [
                'name' => 'Closed',
                'alias' => 'ms-closed',
                'service_type' => ServiceType::REST,
            ],
            [
                'name' => 'Default',
                'alias' => 'ms-default',
                'service_type' => ServiceType::DEFAULT,
            ],
            [
                'name' => 'Open',
                'alias' => 'ms-open',
                'service_type' => ServiceType::OPEN,
            ],
        ]);
}
```

Fig 11. implementación de test y aserción del resource

E. Cumplimiento del Objetivo General

La migración del panel administrativo de micrositiOS a Vue.js se completó con éxito, logrando una mejora significativa en la eficiencia del proceso de gestión y en la experiencia de usuario. La nueva interfaz basada en Vue.js es más rápida, reactiva e intuitiva, lo que facilita la gestión de tareas cotidianas y mejora la satisfacción del usuario. Además, la integración con la biblioteca Spartan y la creación de componentes reutilizables fortalecieron el ecosistema de desarrollo de Evertec, estableciendo una base sólida para futuros proyectos.

1) Módulo administrativo diseño antiguo. (algunas vistas)

placetopay
evertec

Idioma ▼ alejandro ciro ▼

RECURSOS

- Órdenes de pago
- Transacciones
- Reportes
- MicrositiOS
- QR

CONFIGURACIÓN

- Categorías
- Estados
- Países

MicrositiOS Crear nuevo

Mostrar 10 registros

Buscar:

Nombre	Alias	Modalidad	Acción
abierto	abierto	Abierto	
Abierto 2 sitios sin guardar campos	abierto-2-sitios-sin-guardar-cam	Abierto	
Abierto campos script	abierto-campos-script	Abierto	
abierto links err	abierto-links-err	Abierto	
abierto orden en campos dev	abierto-orden-en-campos-dev	Abierto	
abierto orden en campos personalizados	abierto-orden-en-campos-personal	Abierto	
abierto recurrencia traducción	abierto-recurrencia-traduccion	Abierto	
abierto sitio pruebas	abierto-sitio-pruebas	Abierto	
abierto varios sitios	abierto-varios-sitios	Abierto	
Alejandro	jairo-testing	Abierto	

Mostrando registros del 1 al 10 de un total de 71 registros

Anterior 1 2 3 4 5 ... 8 Siguiente

Fig 12. listado de micrositiOS antiguo

The screenshot shows a form titled "Creación de micrositio" with a "Cerrar" button in the top right corner. The form is organized into several sections:

- Sitios:** A dropdown menu.
- Nombre:** A text input field.
- Alias:** A text input field.
- Categorías:** A dropdown menu.
- Expiración del pago (minutos):** A text input field with the value "10".
- Encabezado:** A large text area with the placeholder "Arrastre y suelte aquí los archivos...".
- Logo (Categoría):** A large text area with the placeholder "Arrastre y suelte aquí los archivos...".
- Pie de página:** A large text area with the placeholder "Arrastre y suelte aquí los archivos...".
- Checkboxes:** A list of options on the right side:
 - Generar referencias automáticamente
 - Permite múltiples pagos con la misma referencia
 - Habilitar recurrencia
 - Enviar comprador como pagador
 - Solamente disponible usando Link de Pagos
 - Ocultar en el listado público de búsqueda
 - Deshabilitar el micrositio
 - Redireccionar usando Lightbox
 - Permite iniciar sesión por URL (GET)
 - Seleccionar el color del micrositio

At the bottom left, there are "Cancelar" and "Guardar" buttons.

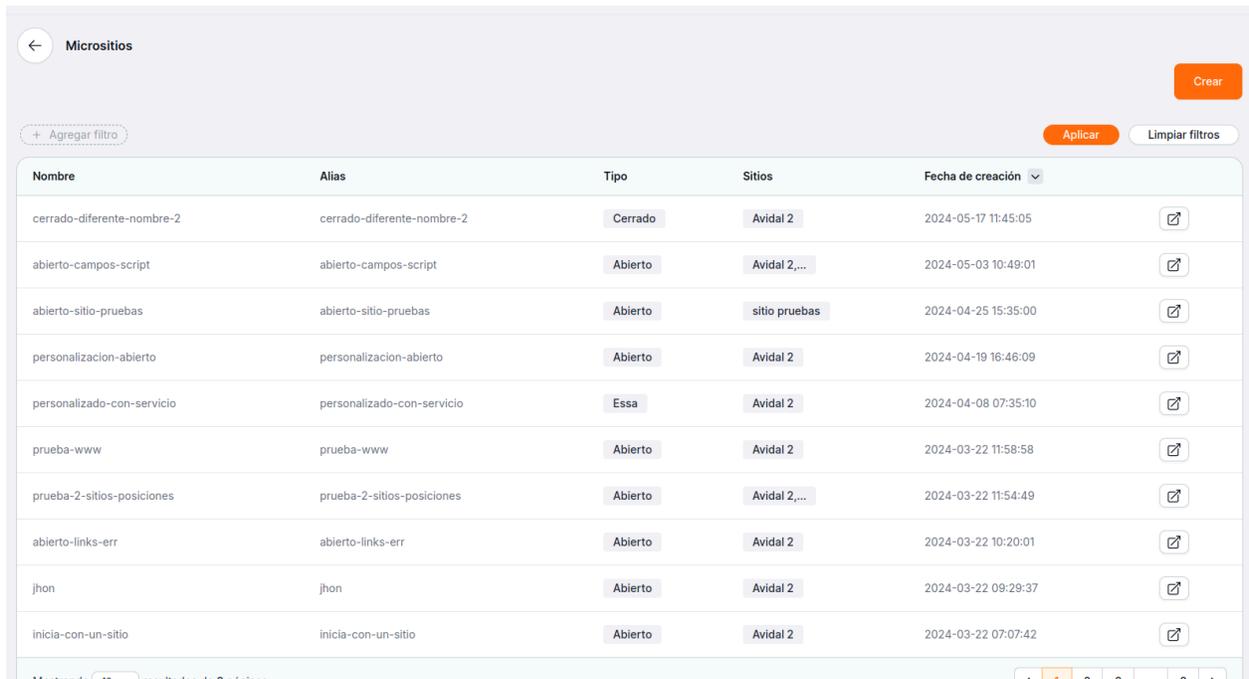
Fig 13. creación de micrositios antiguo

The screenshot shows a web application interface for "placetopay" with the user "alejandro ciro". The main content area is titled "Consultar transacciones" and includes the following elements:

- Navigation:** "Consultar transacciones" (active), "Exportar transacciones", and "Formatos".
- Filters:**
 - Fecha inicio:** 2024-05-21
 - Fecha final:** 2024-05-21
 - Estado:** A dropdown menu.
 - Referencia:** A text input field.
 - Documento:** A text input field.
 - Email:** A text input field.
 - Sitio:** A dropdown menu.
 - Micrositio:** A dropdown menu.
- Action:** "Aplicar filtro" button.
- Left Sidebar:**
 - RECURSOS:** Ordenes de pago, Transacciones, Reportes, Micrositios, QR.
 - CONFIGURACIÓN:** Categorías, Estados, Países.

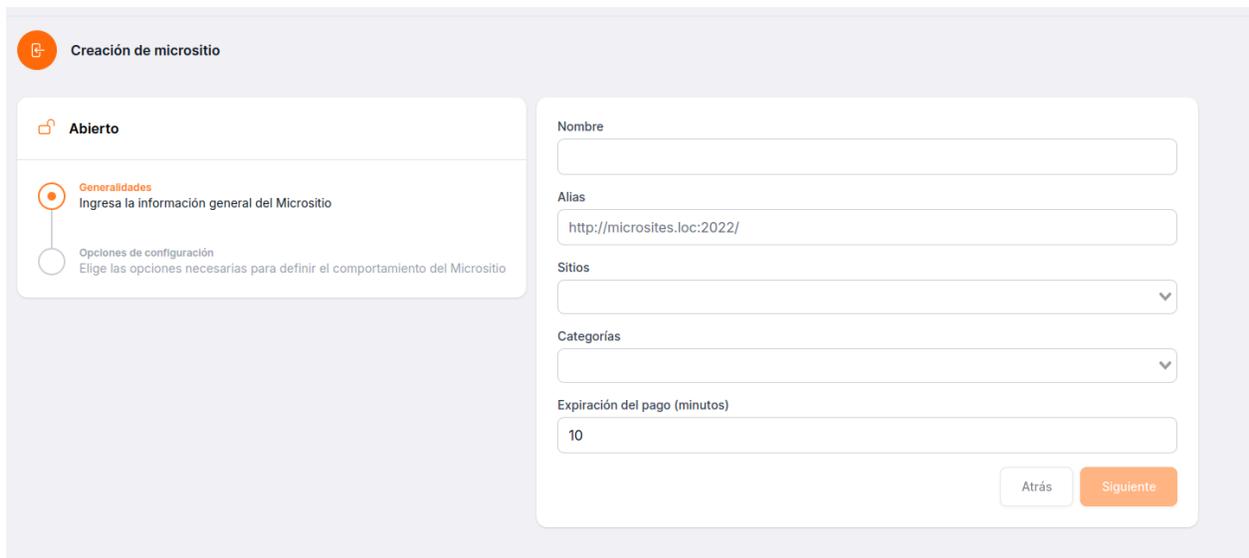
Fig 14. Exporte de transacciones antiguo

2) Módulo administrativo diseño nuevo en VUE. (algunas vistas)



Nombre	Alias	Tipo	Sitios	Fecha de creación
cerrado-diferente-nombre-2	cerrado-diferente-nombre-2	Cerrado	Avidal 2	2024-05-17 11:45:05
abierto-campos-script	abierto-campos-script	Abierto	Avidal 2,...	2024-05-03 10:49:01
abierto-sitio-pruebas	abierto-sitio-pruebas	Abierto	sitio pruebas	2024-04-25 15:35:00
personalizacion-abierto	personalizacion-abierto	Abierto	Avidal 2	2024-04-19 16:46:09
personalizado-con-servicio	personalizado-con-servicio	Essa	Avidal 2	2024-04-08 07:35:10
prueba-www	prueba-www	Abierto	Avidal 2	2024-03-22 11:58:58
prueba-2-sitios-posiciones	prueba-2-sitios-posiciones	Abierto	Avidal 2,...	2024-03-22 11:54:49
abierto-links-err	abierto-links-err	Abierto	Avidal 2	2024-03-22 10:20:01
jhon	jhon	Abierto	Avidal 2	2024-03-22 09:29:37
inicia-con-un-sitio	inicia-con-un-sitio	Abierto	Avidal 2	2024-03-22 07:07:42

Fig 15. Listado de micrositios nuevo



Creación de micrositio

Abierto

Generalidades
Ingresa la información general del Micrositio

Opciones de configuración
Elige las opciones necesarias para definir el comportamiento del Micrositio

Nombre

Alias

Sitios

Categorías

Expiración del pago (minutos)

Atrás **Siguiente**

Fig 16. Creación de micrositios nuevo (paso 1)

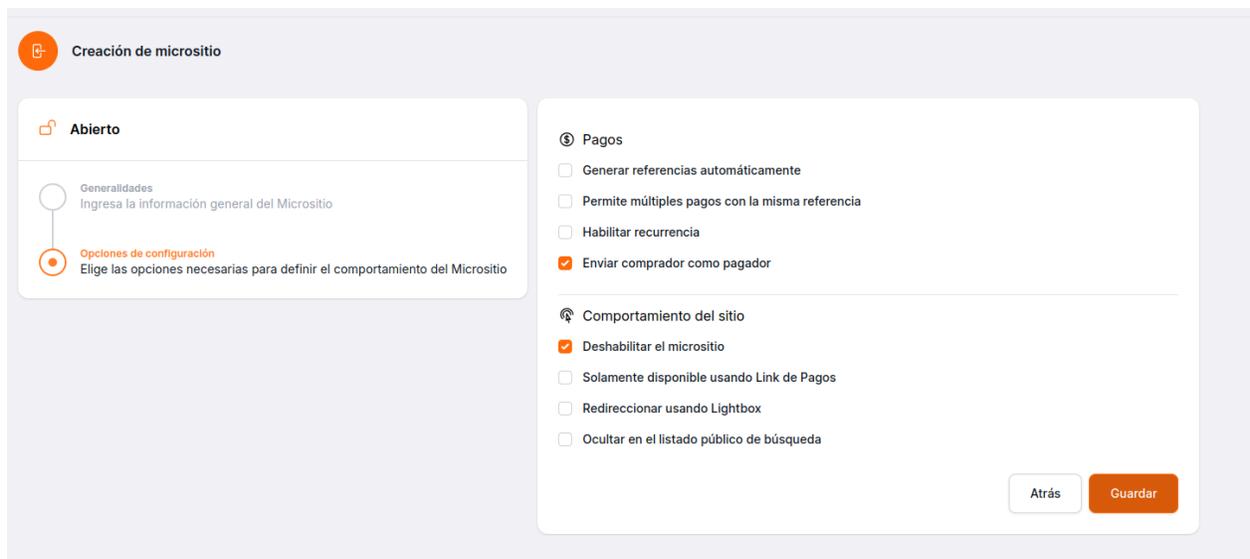


Fig 17. Creación de microsítios nuevo (paso 2)

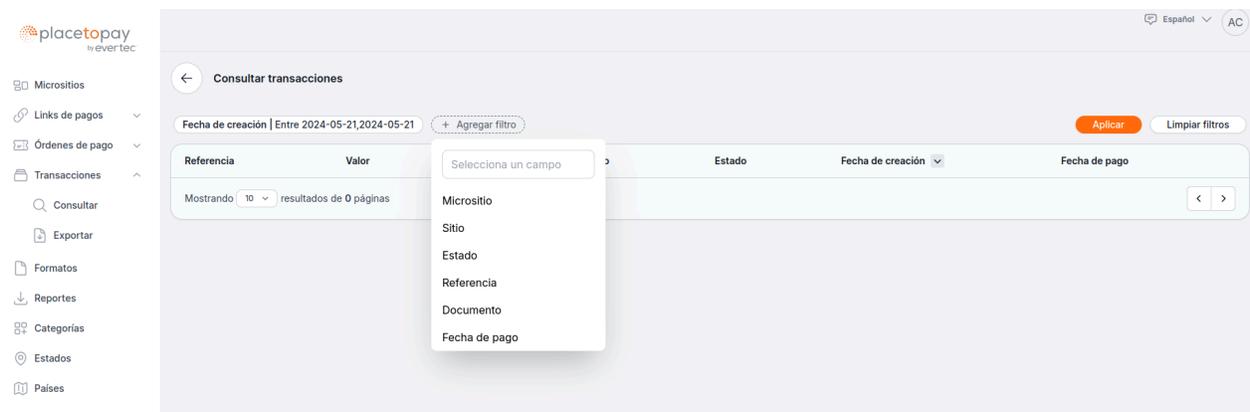


Fig 18. Exporte de transacciones nuevo

VIII. RESULTADOS ADICIONALES

A. Implementación de Single Page Application (SPA)

Dentro del proceso de migración a Vue.js, se implementó Vue Router para transformar el módulo administrativo en una Single Page Application (SPA). Esta implementación permitió que cada componente se renderizara dinámicamente sin necesidad de recargar la página completa, mejorando significativamente la velocidad y la fluidez de la interacción del usuario con el sistema. La SPA resultante ofrece una experiencia de usuario más coherente y rápida, eliminando tiempos de espera y proporcionando transiciones suaves entre diferentes secciones del panel administrativo. Además, la navegación interna es más intuitiva y reactiva, contribuyendo a una mejor usabilidad general.

B. Integración de Vue Router

La integración de Vue Router fue fundamental para lograr la estructura de SPA. Vue Router permitió definir rutas específicas para cada vista y componente dentro del panel administrativo, facilitando una navegación estructurada y lógica. Esta herramienta no solo habilitó la navegación sin recarga de página, sino que también permitió gestionar parámetros dinámicos en las URL, lo que mejoró la gestión de estados y la accesibilidad del sistema. La configuración de rutas protegidas y la implementación de guardias de navegación garantizaron que los usuarios solo pudieran acceder a secciones autorizadas del sistema, mejorando la seguridad y el control de acceso.

Validación de Permisos Consultando al Backend

Se implementó un sistema robusto de validación de permisos que consulta al backend para verificar los derechos de acceso de los usuarios en tiempo real. Cada solicitud realizada por el frontend verifica los permisos asociados al usuario actual, garantizando que solo se muestren y permitan acciones autorizadas. Esta validación dinámica asegura que las políticas de seguridad de Evertec se apliquen estrictamente, protegiendo la integridad del sistema y los datos sensibles. La integración de este mecanismo de control de acceso mejoró significativamente la seguridad del módulo administrativo, evitando accesos no autorizados y asegurando el cumplimiento de los roles y permisos definidos.

Aumento de Coverage en la Aplicación

Durante el desarrollo, se realizó un esfuerzo considerable para aumentar el coverage de pruebas en la aplicación, validando exhaustivamente las funcionalidades del backend. Esto incluyó la creación de pruebas unitarias, de integración y funcionales que garantizaron que cada componente y endpoint del sistema funcionara correctamente. El enfoque en Test-Driven Development (TDD) no solo mejoró la calidad del código, sino que también facilitó la detección y corrección temprana de errores, resultando en un sistema más estable y confiable. El incremento en el coverage de pruebas aseguró que el sistema pudiera manejar diversas situaciones y cargas de trabajo, mejorando su resiliencia y rendimiento general.

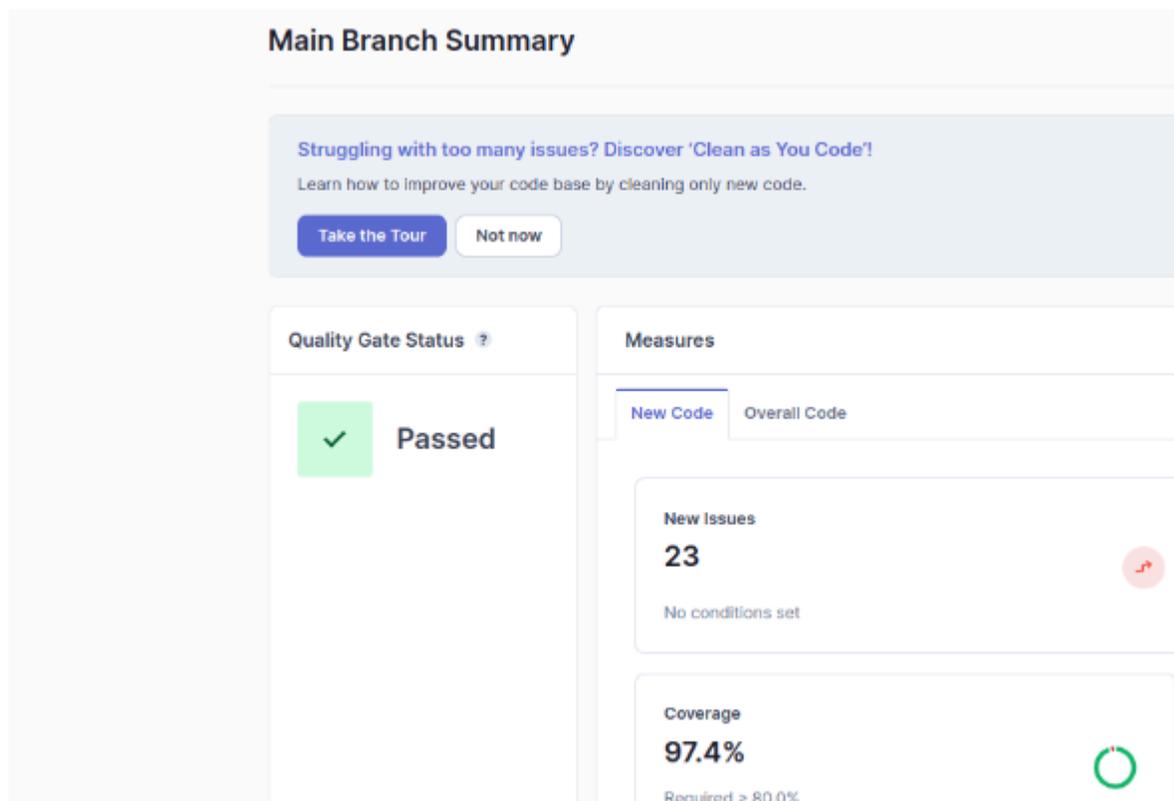


Fig 19. Coverage del aplicativo micrositios

Eliminación de Código Obsoleto

Otro resultado adicional importante fue la eliminación de código obsoleto y redundante. Durante el proceso de migración, se realizó una revisión minuciosa del código existente, identificando y eliminando fragmentos de código que ya no eran necesarios o que podían ser

reemplazados por soluciones más eficientes y modernas. Este esfuerzo no solo limpió la base de código, reduciendo su complejidad y mejorando su mantenibilidad, sino que también mejoró el rendimiento general del sistema al eliminar procesos y dependencias innecesarias. La base de código resultante es más limpia, fácil de entender y mantener, lo que facilita futuras actualizaciones y desarrollos.

Actualización de Dependencias

La actualización de dependencias fue un paso crucial para asegurar que la infraestructura del proyecto estuviera al día con las últimas mejoras y correcciones de seguridad. Durante el proyecto, se revisaron y actualizaron todas las dependencias a sus versiones más recientes. Esto incluyó bibliotecas de frontend, herramientas de desarrollo y paquetes de backend. La actualización de dependencias no solo mejoró la seguridad del sistema al aplicar parches de vulnerabilidades conocidas, sino que también introdujo mejoras de rendimiento y nuevas funcionalidades que beneficiaron el desarrollo y la operación del panel administrativo.

Implementación de Filtros

La implementación de filtros avanzados en el panel administrativo fue otro resultado significativo. Se desarrollaron filtros personalizables que permitieron a los usuarios refinar y segmentar la información presentada en diversas vistas del sistema. Estos filtros mejoraron la capacidad de los usuarios para buscar y gestionar datos, proporcionando herramientas intuitivas para la filtración por diferentes criterios como fechas, estados, categorías, entre otros. La adición de estos filtros no solo optimizó la usabilidad del sistema, sino que también aumentó la eficiencia operativa al permitir a los usuarios encontrar rápidamente la información relevante.

IX. DISCUSIÓN

La migración del panel administrativo de microsítios a Vue.js en Evertec no solo se alinea con la estrategia de transformación digital de la empresa, sino que también representa un avance significativo en términos de eficiencia operativa y experiencia de usuario. Este proyecto abordó

de manera efectiva los desafíos inherentes a la infraestructura existente y mejoró considerablemente la funcionalidad del sistema.

Interpretación Crítica de los Resultados:

1. Mejora en la Experiencia de Usuario (UX):

La adopción de Vue.js y la creación de una Single Page Application (SPA) transformaron la interfaz de usuario, haciéndola más intuitiva y reactiva. Los usuarios experimentan ahora transiciones más suaves y tiempos de carga significativamente reducidos. La implementación de Vue Router permitió definir rutas específicas para cada componente, facilitando una navegación lógica y fluida. Esta mejora en la UX no solo optimiza la interacción del usuario con la plataforma, sino que también contribuye a una mayor satisfacción y eficiencia en la gestión de tareas cotidianas.

2. Validación de Permisos y Seguridad:

La implementación de un sistema robusto de validación de permisos, que consulta al backend para verificar los derechos de acceso en tiempo real, ha reforzado significativamente la seguridad del sistema. Esta validación dinámica asegura que solo los usuarios autorizados puedan realizar ciertas acciones, protegiendo la integridad de los datos y el sistema. Esta mejora es crítica en un entorno de pagos electrónicos donde la seguridad es primordial.

3. Aumento de Coverage de Pruebas:

El enfoque en Test-Driven Development (TDD) y la creación de pruebas unitarias, de integración y funcionales aseguraron que cada componente y endpoint del sistema funcionara correctamente. Este riguroso proceso de pruebas mejoró la estabilidad y confiabilidad del sistema, facilitando la detección y corrección temprana de errores y aumentando la resiliencia del sistema ante diversas cargas de trabajo.

4. Eliminación de Código Obsoleto y Actualización de Dependencias:

La eliminación de código redundante y la actualización de dependencias a sus versiones más recientes mejoraron el rendimiento y la mantenibilidad del sistema. Estas acciones no solo

simplificaron la base de código, sino que también prepararon la infraestructura para futuras mejoras, asegurando que el sistema esté alineado con las mejores prácticas y estándares actuales.

5. Implementación de Filtros Avanzados:

La adición de filtros personalizables mejoró la usabilidad del sistema, permitiendo a los usuarios refinar y segmentar la información de manera eficiente. Esto no solo optimiza la gestión de datos, sino que también mejora la eficiencia operativa al permitir a los usuarios encontrar rápidamente la información relevante.

Evaluación General:

La migración a Vue.js, la integración de la biblioteca Spartan y la adopción de metodologías ágiles y TDD han establecido una base sólida para el desarrollo futuro en Evertec. Este proyecto ha demostrado ser un modelo de buenas prácticas en la migración tecnológica, ofreciendo una experiencia de usuario mejorada y una infraestructura más robusta y flexible.

X. CONCLUSIONES

El proyecto de migración del panel administrativo de microsítios a Vue.js ha logrado cumplir con los objetivos planteados, tanto específicos como generales, y ha tenido un impacto significativo en la operatividad y experiencia de usuario de Evertec. A continuación, se detallan las conclusiones clave:

1. Cumplimiento de los Objetivos Específicos:

Modificación de APIs: Se revisaron y actualizaron aproximadamente el 85% de las APIs existentes, asegurando la compatibilidad con Vue.js y mejorando la gestión de datos entre el backend y el frontend.

Desarrollo de Componentes Reutilizables: Se desarrollaron más de 45 componentes Vue.js reutilizables, de los cuales 6 fueron integrados en la biblioteca Spartan. Estos componentes mejoran la eficiencia y consistencia del desarrollo futuro.

Implementación de Vistas Principales: Se implementaron 5 vistas principales utilizando los nuevos componentes, mejorando significativamente la usabilidad y estética del panel administrativo.

Comunicación Efectiva entre Backend y Frontend: Se aseguró una comunicación fluida y efectiva entre el backend y el frontend para todas las funciones críticas, validando cada endpoint y garantizando la integridad del sistema.

2. Mejora de la Experiencia de Usuario y Eficiencia Operativa:

La nueva interfaz basada en Vue.js es más rápida, reactiva e intuitiva, facilitando la gestión de tareas cotidianas y mejorando la satisfacción del usuario. La SPA resultante proporciona una experiencia de usuario coherente y fluida, eliminando tiempos de espera y mejorando la eficiencia operativa.

3. Fortalecimiento del Ecosistema de Desarrollo:

La integración con la biblioteca Spartan y la creación de componentes reutilizables han fortalecido el ecosistema de desarrollo de Evertec, estableciendo una base sólida para futuros proyectos. La adopción de metodologías ágiles y TDD ha mejorado la calidad del código y la capacidad del equipo para adaptarse a cambios y mejorar continuamente.

XI. RECOMENDACIONES

1. Monitoreo y Mantenimiento Continuo:

Implementar un sistema de monitoreo continuo para identificar y resolver problemas de manera proactiva. Mantener una base de código limpia y actualizada es crucial para asegurar el rendimiento y la seguridad del sistema.

2. Capacitación y Formación del Equipo:

Proporcionar capacitación continua al equipo de desarrollo en nuevas tecnologías y prácticas emergentes. Esto incluye el aprendizaje de nuevas herramientas y frameworks que puedan beneficiar futuros proyectos.

3. Evaluación y Actualización Regular de Dependencias:

Evaluar y actualizar regularmente las dependencias para aprovechar las mejoras y correcciones de seguridad. Esto ayudará a mantener el sistema seguro y optimizado.

4. Expansión de la Biblioteca de Componentes:

Continuar desarrollando y añadiendo nuevos componentes reutilizables a la biblioteca Spartan. Esto fortalecerá el ecosistema de desarrollo de Evertec, facilitando futuros proyectos y asegurando la coherencia en el diseño y funcionalidad de las aplicaciones.

5. Implementación de Pruebas Automatizadas:

Aumentar la cobertura de pruebas automatizadas, incluyendo pruebas de regresión, para identificar errores y garantizar que las nuevas funcionalidades no afecten negativamente al sistema existente.

6. Feedback y Mejora Continua:

Establecer mecanismos para recopilar feedback de los usuarios de manera continua, permitiendo identificar áreas de mejora y realizar ajustes necesarios. Este enfoque centrado en el usuario asegurará que el sistema evolucione de acuerdo con las necesidades y expectativas de los usuarios.

Implementar estas recomendaciones ayudará a Evertec a mantener y mejorar la calidad de sus soluciones tecnológicas, asegurando que la empresa continúe a la vanguardia de la innovación y la excelencia en el servicio al cliente.

REFERENCIAS

- [1] J. L. Arango, “Enfermedades respiratorias del recién nacido,” en *Fundamentos de pediatría: generalidades y neonatología*, J. A. Correa, J. F. Gómez, y R. Posada, Eds. Fondo Editorial CIB, 2000, pp. 463–467.
- [2] “Vue.js”. Accedido el 30 de mayo de 2024. [En línea]. Disponible: <https://vuejs.org/>
- [3] “La Sinfonía Tecnológica: Entendiendo la Relación Entre APIs y Backend”. Uxcale. Accedido el 30 de mayo de 2024. [En línea]. Disponible: [https://www.uxcale.com/la-sinfonia-tecnologica-entendiendo-la-relacion-entre-apis-y-backend/#:~:text=Las%20APIs%20y%20el%20backend,en%20aplicaciones%20modernas%2C%20transformando%20industrias.&text=El%20desarrollo%20de%20software%20moderno,de%20Aplicaciones\)%20y%20el%20backend](https://www.uxcale.com/la-sinfonia-tecnologica-entendiendo-la-relacion-entre-apis-y-backend/#:~:text=Las%20APIs%20y%20el%20backend,en%20aplicaciones%20modernas%2C%20transformando%20industrias.&text=El%20desarrollo%20de%20software%20moderno,de%20Aplicaciones)%20y%20el%20backend)
- [4] “¿Qué son metodologías ágiles?”. IEBSchool. Accedido el 30 de mayo de 2024. [En línea]. Disponible: <https://www.iebschool.com/blog/que-son-metodologias-agiles-agile-scrum/>
- [5] “TDD: Test Driven Development”. Software Crafters. Accedido el 30 de mayo de 2024. [En línea]. Disponible: <https://softwarecrafters.io/javascript/tdd-test-driven-development>
- [6] “Vuejs: Diferentes Formas de Reutilizar Código”. El Abismo de Null. Accedido el 30 de mayo de 2024. [En línea]. Disponible: <https://elabismodenull.wordpress.com/2017/08/29/vuejs-diferentes-formas-de-reutilizar-codigo/>
- [7] “Experiencia de Usuario (UX)”. HubSpot. Accedido el 30 de mayo de 2024. [En línea]. Disponible: <https://blog.hubspot.es/website/experiencia-de-usuario-ux#:~:text=La%20experiencia%20de%20usuario%2C%20tambi%C3%A9n,uso%2C%20accesibilidad%20y%20la%20conveniencia>
- [8] “Interfaz de Usuario”. Rock Content. Accedido el 30 de mayo de 2024. [En línea]. Disponible: <https://rockcontent.com/es/blog/interfaz-de-usuario/#:~:text=La%20interfaz%20de%20usuario%20o,y%20softwares%20de%20diversos%20dispositivos>
- [9] “¿Qué es el Desarrollo Front-end?”. KeepCoding. Accedido el 30 de mayo de 2024. [En línea]. Disponible: <https://keepcoding.io/blog/que-es-el-desarrollo-front-end/>
- [10] “¿Qué es un Framework?”. Cesuma. Accedido el 30 de mayo de 2024. [En línea]. Disponible: <https://www.cesuma.mx/blog/que-es-el-framework.html#:~:text=Un%20framework%20es%20un%20conjunto%20de%20reglas%20y%20convenciones%20que,utilizar%20como%20punto%20de%20partida>
- [11] “¿Qué es Figma?”. Pixartprinting. Accedido el 30 de mayo de 2024. [En línea]. Disponible: <https://www.pixartprinting.es/blog/figma-que-es/>

- [12] “Patrones de Diseño”. Refactoring Guru. Accedido el 30 de mayo de 2024. [En línea]. Disponible: <https://refactoring.guru/es/design-patterns>