



Integración de pasarela de pagos en Ecommerce Blocky

Andrés Elías Carrascal Verona

Informe de práctica para optar al título de Ingeniero de Sistemas

Asesor

José Ignacio López Vélez

Universidad de Antioquia

Facultad de Ingeniería

Ingeniería de Sistemas

Medellín, Antioquia, Colombia

2022

Cita	Andrés Elías Carrascal Verona [1]
Referencia Estilo IEEE (2020)	[1] Andrés Elías Carrascal Verona, “Integración de pasarela de pagos en Ecommerce Blocky”, Trabajo de grado profesional, Ingeniería de Sistemas, Universidad de Antioquia, Medellín, Antioquia, Colombia, 2022.



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

Universidad de Antioquia - www.udea.edu.co

Rector: John Jairo Arboleda Céspedes

Decano/Director: Jesús Francisco Vargas Bonilla

Jefe departamento: Diego José Luis Botía Valderrama

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.

Resumen

El aplicativo Blocky (*Social Marketplace*) necesitaba una integración de una pasarela de pago que permitiera realizar *Split Payment*, es decir, pagos con diferentes empresas, ya que contaba con la pasarela de pagos de Mercado Pago que no permite realizar este tipo de pagos, es por esto que se realizó una investigación de todas las posibles integraciones de pasarelas de pagos que se pudieran utilizar para la plataforma, la pasarela de pagos seleccionada fue ePayco, la cual hace poco tiempo, anexó a sus funcionalidades la opción de realizar *Split Payment*. El proceso de integración se estaba llevando a cabo de manera correcta, no solo con la documentación de la pasarela de pago fue posible avanzar en el desarrollo, sino que también con reuniones personalizadas con ePayco que ayudaron a resolver muchas dudas en el desarrollo, a medida que el desarrollo avanzaba se presentaba al cliente los avances del desarrollo los cuales también ayudaron a solventar algunas dudas en la integración, sin embargo, luego de haber empezado las pruebas de la integración en el ambiente de pruebas, el cliente del aplicativo decidió cambiar las prioridades de desarrollo, por lo que la integración quedó pausada hasta finalizar las prioridades, el resto del tiempo se apoyó en el desarrollo de las nuevas actividades para salir lo más pronto de ellas y así poder continuar con la integración de la pasarela de pagos.

Palabras claves: Pasarela de pagos, *Social Marketplace*, Desarrollo, *Split Payment*.

ABSTRACT

The Blocky application (Social Marketplace) needed an integration of a payment gateway that would allow Split Payment, that is, payments with different companies, since it had the Mercado Pago payment gateway that does not allow this type of payment, that is, For this reason, an investigation was carried out of all the possible integrations of payment gateways that could be used for the platform, the selected payment gateway was ePayco, which recently added the option of making Split Payment to its functionalities. The integration process was being carried out correctly, not only with the documentation of the payment gateway was it possible to advance in the development, but also with personalized meetings with ePayco that helped to resolve many doubts in the development, as that the development progressed, the progress of the development was presented to the client, which also helped to solve some doubts in the integration, however, after having started the integration tests in the test environment, the application client decided to change the priorities of development, so the integration was paused until the priorities were finalized, the rest of the time it was supported by the development of the new activities to get out of them as soon as possible and thus be able to continue with the integration of the payment gateway.

Keywords: Payment gateway, Social Marketplace, Development, Split Payment.

CONTENIDO

1. INTRODUCCIÓN	9
1.1. SITUACIÓN PROBLEMÁTICA	10
1.2. OBJETIVOS	11
1.2.1. Objetivo General.....	11
1.2.2. Objetivos Específicos.....	11
1.3. METODOLOGÍA	11
1.4. ALCANCE	12
2. MARCO TEÓRICO.....	13
3. RESULTADOS Y ANÁLISIS	21
4. CONCLUSIONES Y TRABAJO FUTURO	45
4.1. CONCLUSIONES	45
4.2. TRABAJO FUTURO	45

LISTA DE TABLAS

Tabla 1 Comparación de comisiones de pasarelas de pago.21

Tabla 2 Requisitos funcionales22

Tabla 3 Requisitos no funcionales23

Tabla 4 Ejemplos de pagos23

Tabla 5 Objetivos cumplidos por Sprint41

LISTA DE FIGURAS

Fig. 1.	Angular	14
Fig. 2.	Spring boot	14
Fig. 3.	Cosmo DB y Postgrest.....	15
Fig. 4.	ePayco.....	16
Fig. 5.	Split payment.....	16
Fig. 6.	Scrum.....	17
Fig. 7.	DevOps	17
Fig. 8.	Marketplace	17
Fig. 9.	Social marketplace.....	18
Fig. 10.	Artefacto	18
Fig. 11.	Node	19
Fig. 12.	JavaScript.....	19
Fig. 13.	Ticket	19
Fig. 14.	WebHook	20
Fig. 15.	PCI DSS	20
Fig. 16.	Reverse proxy	20
Fig. 17.	Resumen de la compra, ejemplo 1	24
Fig. 18.	Interfaz compra, ejemplo 1	25
Fig. 19.	Interfaz respuesta, ejemplo 1	25
Fig. 20.	Resumen de la compra, ejemplo 2	26
Fig. 21.	Interfaz compra, ejemplo 2	27
Fig. 22.	Interfaz respuesta, ejemplo 2	27
Fig. 23.	Resumen de la compra, ejemplo 3	28
Fig. 24.	Interfaz compra, ejemplo 3	29

Fig. 25.	Interfaz respuesta, ejemplo 3	29
Fig. 26.	Resumen de la compra, ejemplo 4	30
Fig. 27.	Interfaz compra, ejemplo 4	31
Fig. 28.	Interfaz respuesta, ejemplo 4	32
Fig. 29.	Resumen de la compra, ejemplo 5	33
Fig. 30.	Interfaz compra, ejemplo 5	34
Fig. 31.	Interfaz respuesta, ejemplo 5	35
Fig. 32.	Resumen de la compra, ejemplo 6	36
Fig. 33.	Interfaz compra, ejemplo 6	37
Fig. 34.	Interfaz respuesta, ejemplo 6	38

1. INTRODUCCIÓN

[Blocky](#) es un *social marketplace*, es decir una red social dentro de un centro comercial virtual. La aplicación que permite que las personas puedan crear sus centros comerciales, estos son creados con los diferentes almacenes que se pueden encontrar en la aplicación, a éstos almacenes se les llama bloques y son creados por diferentes marcas que se encuentran registradas en la plataforma; esto no solo ayuda a que las personas puedan construir sus centros comerciales a su gusto, sino que también impulsa el conocimiento de nuevas marcas pequeñas y grandes, que posiblemente no son tan conocidas [1].

La gran diferencia que tiene Blocky contra otras plataformas, es que une la idea de una red social y un marketplace en un solo lugar, de esta forma, ayuda a conectar diferentes usuarios y compartir todos sus gustos de cualquier producto creando historias, realizando publicaciones y hasta creando listas de deseos que ayudan a que otras personas puedan comprar regalos para épocas especiales como cumpleaños, matrimonios, aniversarios, entre otros [1]; todas estas funcionalidades hacen que Blocky sea un proyecto que permite visualizar algo completamente diferente a los que normalmente las personas están acostumbradas a ver.

Dentro de sus funcionalidades también se encuentra la de realizar compras, ya sea como un regalo o como una compra propia; actualmente Blocky, no cuenta con la posibilidad de pagar productos de diferentes marcas en una sola transacción, es decir, debe realizar un pago por cada grupo de productos asociados a la misma marca, lo que genera un poco de inconformidad en la forma de realizar las compras. Es por esto que se ve la necesidad de integrar una pasarela de pago que permita realizar un solo pago por todo el pedido independiente si los productos son de la misma o diferente marca, e internamente divide el pago para cada empresa asociada, lo que se conoce como un *split payment*.

1.1. SITUACIÓN PROBLEMÁTICA

Hoy en día el poder realizar compra de todos los artículos que se quiera dentro un *social marketplace* sin importar de qué empresa sea, es un proceso que para muchas personas es algo normal, pero internamente, para el *social marketplace*, la implementación para el pago de los productos tiene que estar configurado de tal forma que la pasarela de pago permita realizar *split payment*. Es decir, que, al comprar varios productos de diferentes empresas, internamente cada porcentaje de dinero vaya a cada empresa dividiendo el pago total y no a una sola empresa.

Blocky como *social marketplace*, tiene integrado a su forma de pago la pasarela de Mercado Pago, pero esta pasarela no permite realizar *split payments*, por lo que muchas personas no finalizan sus compras, ya que al momento de realizar el pago no se puede continuar sino solo por 1 empresa a la vez.

Dentro de las conversaciones y reuniones entre la empresa desarrolladora y el cliente, se llegó a la conclusión de que hay buscar una pasarela de pagos que permita realizar *split payments*, adicionalmente, que maneje un valor no tan alto de comisión para poder integrarlo al *social marketplace*, dentro de las opciones que se tenían desde un principio, estaba ePayco para la integración, pero no contaba con *split payment* por lo que fue descartado, hace poco tiempo, la pasarela de pagos incorporó esta opción dentro de sus funcionalidades, lo que llevó a realizar su integración que, aparte de permitir integrar sus nuevas funcionalidades, tiene un valor bajo de comisión con respecto a muchas otras pasarelas de pago.

1.2. OBJETIVOS

1.2.1. Objetivo General

Integrar una pasarela de pago que permita dividir un pago en diferentes cuentas asociadas (split payment) en la aplicación Blocky.

1.2.2 Objetivos Específicos

- Investigar sobre la pasarela de pagos de ePayco en la documentación oficial.
- Documentar en Azure DevOps toda la información relevante sobre la pasarela de pago.
- Integrar el SDK de la pasarela de pago a la plataforma de Blocky.
- Diseñar y ejecutar un plan de pruebas para los ambientes de prueba y producción, incorporando todos los posibles escenarios de una compra: productos de una sola marca, compras con productos de múltiples marcas, productos en promoción, uso de códigos de descuento, entre otros.
- Realizar pruebas unitarias para cada uno de los escenarios de un proceso de compra.
- Entregar un prototipo funcional de la pasarela de pago que permita realizar pagos individuales y únicos y que permita comprar con las diferentes formas de pagos disponibles.

1.3. METODOLOGÍA

La metodología propuesta es SCRUM. Blocky cumplirá el rol de Product Owner y será quien defina el Product Backlog o requerimientos a desarrollar dentro de sprints de 15 días hábiles. El Scrum Master es quien velará por el cumplimiento de las asignaciones y el control del tiempo invertido en cada uno de los Sprints.

Contaremos con un Product Backlog de requisitos priorizados que se desarrollarán a través de diferentes sprints, según el tamaño y complejidad, durante la ejecución del proceso, se llevarán a cabo los siguientes eventos:

- Reunión de planeación de cada sprint (Sprint planning)., donde se adquieren los compromisos a cumplir en un sprint, se especifican a mayor detalle las funcionalidades a desarrollar y sus criterios de aceptación (Sprint backlog).

- Reuniones diarias de todo el equipo de trabajo de máximo 30 minutos, durante la ejecución de cada sprint (Scrum Daily).
- Reunión de revisión del avance del producto (Sprint review), una vez desarrolladas las funcionalidades del sprint (Sprint execution).
- Reunión de retroalimentación del trabajo realizado del equipo de trabajo durante la ejecución de cada sprint (Sprint retrospective).

El proyecto incorporará toda la visión DevOps, la cual nos permite mejorar la comunicación entre los equipos (usuarios, desarrollo y operaciones), garantizar la calidad de los procesos de despliegues, obtener retroalimentación inmediatamente de cualquier actividad que se esté desarrollando y automatizar la mayor cantidad de actividades en el desarrollo del producto, agilizando el proceso de entrega de valor al usuario final. Todo esto será gestionado con plataformas como Microsoft Teams para todas las reuniones y Azure DevOps como gestor de historias, tareas, entre otros.

Aparte de la utilización de Scrum, también se realizarán reuniones agendadas con el asesor interno para ir validando el estado del proyecto y de las prácticas académicas en general.

1.4. ALCANCE

Para lograr el alcance del proyecto, se deben pasar por ciertos desarrollos que se ajusten a todos los estándares necesarios para cumplir con el objetivo general del proyecto.

A continuación, se describe cada punto que hace parte del alcance:

- Se debe entregar una documentación de todo lo relacionado a la plataforma de ePayco, esta documentación debe especificar todos los procesos que se implementaron dentro del *social marketplace*.
- Entregar en la plataforma de Azure, toda la información más relevante de la pasarela de pagos, todo lo que comprende: Formas de pagos, cómo realizar pruebas, entre otras.
- Entregar plan de pruebas para que el cliente pueda realizar pruebas tanto en ambiente de pruebas como en ambiente de producción, este plan de pruebas debe explicar de manera clara, cómo hacer el proceso de compra de las diferentes formas de pago que permite realizar ePayco y con diferentes tipos de empresas.

- Entregar pruebas unitarias en cada uno de los artefactos que integre la pasarela de pago y de esta forma prevenir en su mayor posibilidad errores en la integración.
- Entregar un prototipo funcional del aplicativo donde cualquier persona pueda realizar una compra dentro de la plataforma ya sea una compra que incorpore a una sola empresa o a diferentes empresas.

Todos estos objetivos se deben aplicar sobre una sola pasarela de pagos, en este caso sobre ePayco, ya que fue la seleccionada para integrarla en el proyecto.

2. MARCO TEÓRICO

Blocky es un ecommerce *All-In-One* que cuenta con diferentes soluciones tecnológicas para apalancar su negocio, tales como:

- Aplicativos webs para sus usuarios finales y administradores
- Aplicativos móviles nativos, Android y IOS
- Aplicativos de integración con otros sistemas, tales como Vtex y RDStation
- Configuración y despliegue de servicios totalmente en la nube de Azure

La aplicación web, es un sistema que está realizado para ser usado a través de un navegador web, este está basado en un modelo de cliente-servidor que, para su correcto funcionamiento, necesita de los siguientes componentes: Servidores, un navegador, y uno o varias bases de datos que permitan gestionar un conjunto de información por uno o varios usuarios, algunos ejemplos de aplicaciones web existentes son Gmail, Google docs, Facebook, entre otros. Hoy en día se caracterizan por poder utilizarse en múltiples dispositivos, por ejemplo, en tabletas, celulares y computadoras, también algunas permiten usarse de forma offline entre otras funcionalidades más. La calidad del software se mide con el nivel en que este se ajusta a los requisitos no funcionales, estos requisitos se lograron, gracias a la utilización de algunas herramientas las cuales se van a describir a continuación.

FrontEnd: Es considerada la parte del programa o de un dispositivo por medio de la cual los usuarios acceden directamente, es por esto que los usuarios son los que se encargan de la interacción con todas las tecnologías de diseño y desarrollo web que puedan correr dentro de un navegador, estas interacciones se visualizan por medio de imágenes y animaciones que son los

formatos que los desarrollos *FrontEnd* realizan para dar componentes visuales en un sitio web. [2] En este caso, se va a utilizar como tecnología el *Framework* de Angular, que internamente va a ser uso de tecnologías como SCS, Typescript y HTML que internamente darán vida a la interacción con los usuarios de la aplicación y además, también se hará uso de patrones arquitectónicos que ayuden a elaborar un proyecto más organizado y robusto.

Fig. 1. Angular



Imagen tomada de: <https://angular.io/presskit>

BackEnd: Es la capa de acceso a datos de un dispositivo o software, a diferencia del *FrontEnd*, maneja todo lo que no es visible para el usuario, lo que comprende la lógica de la aplicación que maneja todos los datos. Es acá donde los desarrolladores se encargan de que toda la lógica de la aplicación funcione de manera correcta y toda la información que se procese se haga de manera segura y genere un buen desempeño en el aplicativo. [2] En este caso se va a ser uso del *Framework* Spring Boot, el cual se puede usar con el lenguaje Java y también maneja lenguaje SQL, más conocido como JPQL en el *Framework*, también permitirá la utilización de bases de datos H2 para realizar pruebas entre otras funcionalidades más.

Fig. 2. Spring boot



Imagen tomada de: <https://mobile.twitter.com/springboot>

Base de datos: Una base de datos es toda la recopilación organizada de información o datos estructurados, estos datos se almacenan de forma electrónica. Las bases de datos están controladas por un sistema de gestión de base de datos, la mayoría de los datos funcionan estando en estructuras de filas y columnas en una serie de tablas lo que permite que su procesamiento sea más eficaz, entre los procesamientos están la gestión, modificación, actualización, control y organización de los datos, todos estos procesos se pueden realizar por medio de un lenguaje de consulta estructurada (SQL). Dentro de los tipos de bases de datos, las más conocidas son las relacionales y no relacionales, las relacionales son las que proporcionan una forma más eficiente y flexible de acceder a la información estructurada que está organizada en conjuntos de tablas o columnas, las no relacionales permiten manipular datos estructurados y semiestructurados, lo que no obliga a tener que definir cómo se deben componer todos los datos que se vayan a insertar. [3] Para este caso, se va a ser utilización de 2 bases de datos, una es postgresql, la cual es una base de dato relacional y la otra es Cosmos DB, la cual es una base de datos de Azure que es basada en MongoDB y es una base de datos no relacional.

Fig. 3. Cosmo DB y Postgrest



Imágenes tomadas de: https://www.flaticon.com/free-icon/postgre_5968342, <https://sqlplayer.net/azure-cosmos-db/>

Pasarela de pago: Una pasarela de pago es un portal que permite conectar una cuenta bancaria con un procesador de pagos, este proceso se da por medio de forma virtual usando servicios de pagos web y APIS que permiten transmitir la información de las diferentes transacciones realizadas o de forma personal por medio de una terminal de pago, entre algunas de las pasarelas de pagos más conocidas se tiene Paypal, Stripe, Square, entre otras. [4] Actualmente la plataforma utiliza la pasarela de mercado pago, la idea es migrar a una pasarela que permita realizar *Split Payment*, en este caso ePayco.

Fig. 4. ePayco



Imagen tomada de: <https://epayco.com/recursos-graficos/>

Split Payment: Los pagos individuales son los que permiten personalizar la división de pagos, capturas o reembolsos independientemente de las cuentas que se tenga en la plataforma, muchos *marketplaces* personalizan esta división de pagos para adaptar los modelos de negocios. [5] En este caso, la plataforma de Blocky tiene la necesidad de realizar compras con diferentes cuentas, en este caso, marcas, ya que mercado pago no permite realizar *split payment*, por lo que se va a realizar la integración por medio de la pasarela de pagos de ePayco.

Fig. 5. Split payment



Imagen tomada de: <https://www.vectorstock.com/royalty-free-vector/split-payment-framed-icon-vector-18212965>

Scrum: Scrum son todos esos procesos que ayudan a obtener resultados positivos dentro de un proyecto, basado en la implementación de buenas prácticas que apoyan a que el trabajo sea más colaborativo y de los mejores resultados posibles. Dentro de algunos términos más conocidos dentro de *Scrum* se tienen: *Product owner* que hace referencia al cliente y es el encargado de priorizar los objetivos creando un balance entre el valor que aporta y el costo tiene dentro del proyecto, *scrum master* quien es el encargado de que el equipo pueda mantener el foco para cumplir con los objetivos, *product backlog* que son un listado donde se encuentran las diferentes actividades que se pretenden realizar durante el desarrollo del proyecto, entre otros. [6] En este caso se va a utilizar dentro del proyecto buscando obtener un proceso más ágil, colaborativo.

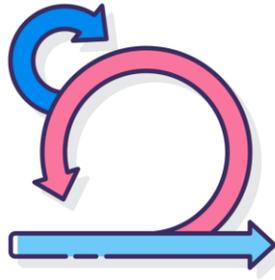
Fig. 6. Scrum

Imagen tomada de: https://www.flaticon.com/free-icon/scrum_2620863

Azure DevOps: Azure DevOps son un conjunto de herramientas y servicios que permiten administrar el ciclo de vida de un proyecto, dentro de sus principales ventajas está que soporta cualquier lenguaje de programación. [7] Todo el ciclo de vida del proyecto se va a administrar con esta plataforma que actualmente, es muy usada por los amplios servicios que ofrece.

Fig. 7. DevOps

Imagen tomada de: <https://blog.sandro-pereira.com/2020/03/06/microsoft-integration-and-azure-stencils-pack-for-visio-new-version-available-v6-1-0/>

Marketplace: La palabra marketplace viene de la unión entre las palabras *market* y *place*, las cuales significan mercado y lugar, por lo que se puede considerar como un lugar para hacer compras de manera virtual, una especie de vitrina virtual en la que los clientes pueden acceder [8].

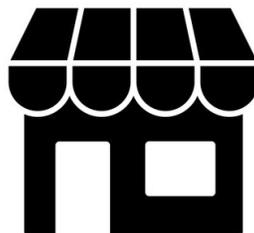
Fig. 8. Marketplace

Imagen tomada de: <https://stock.adobe.com/co/search?k=marketplace+icon>

Social Marketplace: El *social marketplace* son las diferentes plataformas de empresas que le ofrecen un servicio a los usuarios de sus plataformas, en otras palabras, es una red empresarial que es posible gracias a una variedad de soportes digitales [9].

Fig. 9. Social marketplace



Imagen tomada de: <https://www.istockphoto.com/es/vector/icono-web-de-banner-de-comercio-social-para-comercio-electrónico-y-marketing-en-gm1172148385-325029873>

Artefacto de software: Un artefacto es un producto tangible resultante del proceso de desarrollo de software. Algunos artefactos como los casos de uso, diagrama de clases u otros modelos UML ayudan a la descripción de la función, la arquitectura o el diseño del software. [10].

Fig. 10. Artefacto

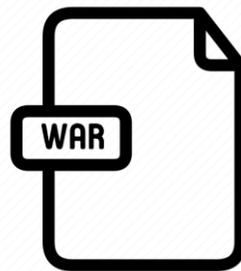


Imagen tomada de: https://www.iconfinder.com/icons/4564453/war_war_extension_war_file_icon

Node: Node.js®, Node.js, es un entorno en tiempo de ejecución multiplataforma para la capa del servidor (en el lado del servidor) basado en JavaScript. [11].

Fig. 11. Node

Imagen tomada de: https://www.flaticon.es/icono-gratis/nodejs_919825

JS: JavaScript es un lenguaje de programación o de secuencias de comandos que te permite implementar funciones complejas en páginas web, cada vez que una página web hace algo más que sentarse allí y mostrar información estática para que la veas, muestra oportunas actualizaciones de contenido, mapas interactivos, animación de Gráficos 2D/3D, desplazamiento de máquinas reproductoras de vídeo, etc [12].

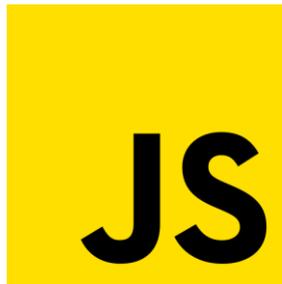
Fig. 12. JavaScript

Imagen tomada de: https://www.flaticon.es/icono-gratis/js_5968292

Ticket: Un ticket de soporte técnico es una forma de hacernos llegar por escrito una consulta o de reportar un problema relacionado con cualquiera de nuestros servicios [13].

Fig. 13. Ticket

Imagen tomada de: https://www.flaticon.es/icono-gratis/ticket-de-soporte_1786662

Webhook: Un webhook es, por decirlo de la manera más sencilla, un sistema de comunicación automático entre apps. Lo que hacen es aportar una solución sencilla para el intercambio de datos entre aplicaciones web, o incluso entre nuestro sitio web y esas aplicaciones [14].

Fig. 14. WebHook



Imagen tomada de: <https://www.svix.com/resources/webhook-icon/>

PCI DSS: Es un conjunto de controles de seguridad diseñados para garantizar que todas las empresas que recopilan y procesan información de tarjetas de crédito mantengan un entorno seguro [15].

Fig. 15. PCI DSS



Imagen tomada de: <https://qualitytelecom.es/que-es-certificacion-pci-dss-nivel-1/>

Reverse proxy: En términos generales, un servidor proxy es una interfaz de comunicación en una red que se hace cargo de las peticiones y las transmite en calidad de representante a un ordenador de destino. [16].

Fig. 16. Reverse proxy

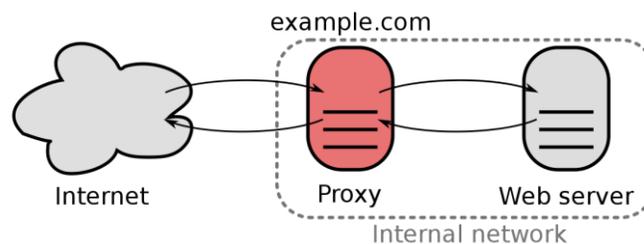


Imagen tomada de: https://en.wikipedia.org/wiki/Reverse_proxy

3. RESULTADOS Y ANÁLISIS

La propuesta del trabajo le apunta al objetivo de aumentar la calidad del producto actual del proyecto, es por esto que el trabajo se enfocó en el soporte de *split payment*, debido a que era uno de los requerimientos más importantes que se tenían en el proyecto, pues normalmente al realizar una compra de uno o varios productos, cualquier persona quisiera comprar la cantidad que pueda o que esté disponible sin tener en cuenta si un producto es de una empresa específica, el problema es que con la pasarela que el proyecto tiene ahora, si se compran productos de diferentes empresas, tienen que ser compras diferentes. Después de evaluar las diferentes herramientas de desarrollo, se encontró que la que mejor se ajustaba al proyecto era la de ePayco, ya que aparte de ser colombiana, maneja la funcionalidad de split payment, adicionalmente a esto, el soporte que brinda la pasarela de pago, es muy bueno, pues hasta reuniones personalizadas se pueden hacer, otro punto adicional que impulsó la selección de esta pasarela de pago, es que maneja un buen porcentaje de comisión. Por todas estas características que tiene ePayco, es que se seleccionó como la pasarela a implementar en el proyecto. A continuación, se mostrará una tabla que permitirá visualizar la ventaja que tiene ePayco con respecto a otras 4 pasarelas de pago más utilizadas en Colombia en cuanto a compatibilidad de split payment, seguridad y soporte.

Tabla 1 Comparación de comisiones de pasarelas de pago.

	Wompi	ePayco	Mercado Pago	PayU	PayPal
Soporte	Asesoría en línea, Correo, Preguntar frecuentes y foro	Base de conocimiento, tickets de soporte, reuniones con soporte, llamadas telefónicas y asesoría en línea	Base de conocimiento y generación de ticket	Base de conocimiento, chat en línea y generación de tickets	Generación de mensajes de texto, preguntas a comunidad, centro de resolución y llamadas
Split payment	X	✓	X	✓	✓

PCI DSS (Seguridad)	✓	✓	✓	✓	✓
Comisión	Depende del tiempo de desembolso: : Diario: 2.85% + \$800 + IVA Semanal: 2.75% + \$700 + IVA Mensual: 2.65% + \$700 + IVA	2 tipo de cobros por transacción: Normal: 2.99% + \$900 + IVA Con cuenta Davivienda o Daviplata: 2.68% + \$900	Depende del tiempo de desembolso: Instantáneo: 3,29% + \$800 + IVA 7 días: 2,99% + \$800 + IVA 15 días: 2,79% + \$800 + IVA	3,49% + 800 pesos + IVA por transacción	5.4% + 0.30 USD por transacción y si se quiere traer ese dinero a Colombia se debe hacer a través de Nequi que cobra el 5% + IVA, pero nunca más de 10 dólares

Estudio de requisitos: A continuación, se describe por medio de una tabla cuáles fueron los requisitos y de qué tipo era cada uno.

Tabla 2 Requisitos funcionales

Requisitos funcionales
El sistema debe permitir aplicar cupón y descuento con pagos únicos y divididos.
El sistema debe permitir gestionar las compras de forma separada por empresa.
El sistema debe permitir gestionar las PQRS de manera separada.
El sistema debe mostrar el resultado de la transacción.
El sistema debe notificar por correo las compras realizadas a cada empresa
El sistema debe notificar por correo al comprador de los productos que pagó
El sistema debe permitir gestionar el pedido por diferentes empresas

Tabla 3 Requisitos no funcionales

Requisitos no funcionales
El sistema debe permitir realizar <i>split payments</i> .
El sistema debe permitir realizar pagos únicos.
El sistema debe permitir usarse en ambiente productivo y de pruebas.
Se debe entregar una documentación de la pasarela de pago

Los resultados que se obtuvieron al final fueron satisfactorios, pues, aunque al inicio la integración se estaba tornando un poco complicada, se lograron realizar compras en el ambiente de pruebas que reflejaron que el proceso había quedado bien desarrollado. A continuación, se muestra una tabla que contiene información de las pruebas que se realizaron en las compras.

Estas pruebas fueron realizadas teniendo en cuenta la utilización de 1 o más empresas en la compra, en las compras que se realizaron se tuvo en cuenta si se aplicaba el cupón o no y también si el producto estaba en promoción. Un ejemplo sería el #1 de la tabla, donde la compra se realizó con 1 empresa sin aplicar cupón ni descuento y el resultado fue exitoso.

Tabla 4 Ejemplos de pagos

# Ejemplo	# Empresas	Aplicó cupón	Aplicó descuento	Resultados
1	1	X	X	✓
2	2	X	X	✓
3	1	X	✓	✓
4	1	✓	✓	✓
5	2	X	✓	✓
6	2	✓	✓	✓

A continuación, se van a mostrar imágenes donde se pueden apreciar los resultados de cada uno de los ejemplos comentados anteriormente.

Ejemplo #1:

- **Resumen de la compra:** En este ejemplo se puede apreciar la compra de un artículo, tiene un costo de \$60.000 y con el valor del costo de envío da un total de \$72.000.

Fig. 17. Resumen de la compra, ejemplo 1

Resumen del pedido

Bombillas

 [Volver a la bolsa de compras](#)

Artículos 1

	Bombillo Led Redondo Ahorrador 15w Capsula Alta Potencia AC125623621 + \$60.000	Forma : Cápsula Tecnología de iluminación : LED
---	---	---

Cupones

Costo envío	\$12.000
Pago total	\$72.000

- **Interfaz de compra:** En esta imagen se puede apreciar la redirección a la ventana de ePayco para terminar la compra, en la parte derecha se puede ver el total de la compra y el nombre de los artículos, en la parte izquierda se puede visualizar las diferentes formas de pago.

Fig. 18. Interfaz compra, ejemplo 1

- **Interfaz de respuesta:** En esta imagen se puede visualizar el resultado de la compra, donde se aprecia el costo de envío, el costo del artículo, el valor total de la compra, el estado de la compra e información del usuario que realizó la compra.

Fig. 19. Interfaz respuesta, ejemplo 1

Detalles del pedido		
Número del pedido		
#109610437		
Detalles del pago		
Nombre	Tipo y número de documento	Número de orden
Andres E Carrascal Verona	CC 109885922	#2509
Fecha y hora de la transacción	Medio de pago	Estado de transacción
2022-10-20 12:22 AM	BANCO DE PRUEBAS	Aceptada
 Bombillo Led Redondo Ahorrador 15w Capsula Alta Potencia		
\$60.000		
1 Unidad		
Valor	Descuento total promoción	Costo envío
\$72.000	\$0	\$12.000

Ejemplo #2:

- **Resumen de la compra:** En este ejemplo se puede apreciar la compra de 2 artículos, tienen un costo de \$37.000 los 2 y con el valor del costo de envío da un total de \$59.000.

Fig. 20. Resumen de la compra, ejemplo 2

Resumen del pedido

Bombillas , Mundo Otaku

 [Volver a la bolsa de compras](#)

Artículos 2

	Bombillo Vintage Pera Filamento 6w Tipo Edison Luz Cálida AC3532626263221 + \$25.000	Forma : Globo Tecnología de iluminación : LED
	Medias Tobilleras Los 7 Pecados Capiales AC1516161221	Talla : Única Color : 

Cupones

Costo envío **\$22.000**

Pago total **\$59.000**

- **Interfaz de compra:** En esta imagen se puede apreciar la redirección a la ventana de ePayco para terminar la compra, en la parte derecha se puede ver el total de la compra y el nombre de los artículos, en la parte izquierda se puede visualizar las diferentes formas de pago.

Fig. 21. Interfaz compra, ejemplo 2

Transacción en modo pruebas

blocky Pago seguro por ePayco ES / EN

1 Seleccione el medio de pago 2 Complete la información 3 Confirmación

→ Ingrese su correo electrónico para iniciar

alguien@dominio.com Continuar

→ Seleccione su medio de pago preferido

Productos Davivienda Pague Fácil, Rápido y Seguro

Tarjeta de Crédito y Débito Pague con tu tarjeta hasta 36 cuotas VISA

PSE (Cuentas de ahorro y corriente) Pagos seguros en línea con su banco Ver bancos soportados

DaviPlata

SU COMPRA

Bombillo Vintage Pera Filamento 6w Tipo Edison Luz Cálida, Medias Tobilleras Los 7 Pecados Capitales

Total \$59,000.00 COP

¿PREGUNTAS AL COMERCIO?

Comercio BLOCKY
Llámenos 3138342394
Escribanos info@blocky.shop
https://blocky.shop/

PAGO SEGURO

- **Interfaz de respuesta:** En esta imagen se puede visualizar el resultado de la compra, donde se aprecia el costo de envío, el costo de los artículos, el valor total de la compra, el estado de la compra e información del usuario que realizó la compra.

Fig. 22. Interfaz respuesta, ejemplo 2

Detalles del pedido

Número del pedido
#109610515

Detalles del pago

Nombre Andres E Carrascal Verona	Tipo y número de documento CC 109885922	Número de orden #2510
Fecha y hora de la transacción 2022-10-20 12:25 AM	Medio de pago BANCO DE PRUEBAS	Estado de transacción Aceptada

Bombillo Vintage Pera Filamento 6w Tipo Edison Luz Cálida
 \$25,000
 1 Unidad

Medias Tobilleras Los 7 Pecados Capitales
 \$12,000
 1 Unidad

Valor \$59,000	Descuento total promoción \$0	Costo envío \$22,000
--------------------------	---	--------------------------------

Ejemplo #3:

- **Resumen de la compra:** En este ejemplo se puede apreciar la compra de un artículo con descuento, el costo con descuento es de \$22.000 y con el valor del costo de envío da un total de \$72.000.

Fig. 23. Resumen de la compra, ejemplo 3

Resumen del pedido

Bombillas

 [Volver a la bolsa de compras](#)

Artículos 1

	Bombillo Vintage Pera Filamento 6w Tipo Edison Luz Cálida AC3532626263221 + \$22.000 -\$25.000	Forma : Globo Tecnología de iluminación : LED
---	---	---

Cupones

Costo envío \$12.000

Pago total **\$34.000**

- **Interfaz de compra:** En esta imagen se puede apreciar la redirección a la ventana de ePayco para terminar la compra, en la parte derecha se puede ver el total de la compra y el nombre de los artículos, en la parte izquierda se puede visualizar las diferentes formas de pago.

Fig. 24. Interfaz compra, ejemplo 3

- **Interfaz de respuesta:** En esta imagen se puede visualizar el resultado de la compra, donde se aprecia el costo de envío, el costo del artículo con su respectivo descuento, el valor total de la compra, el estado de la compra e información del usuario que realizó la compra.

Fig. 25. Interfaz respuesta, ejemplo 3

Ejemplo #4:

- **Resumen de la compra:** En este ejemplo se puede apreciar la compra de un artículo que tiene descuento y se le aplicó un cupón, el costo con descuento es de \$47.520 y con un 12% de descuento con el cupón utilizado, da un total de \$41.818, con el valor del costo de envío da un total de \$51.818.

Fig. 26. Resumen de la compra, ejemplo 4

The screenshot shows an order summary for 'Mundo Otaku'. It includes a link to return to the shopping bag, a list of items with a price breakdown, a coupon application section, and a final total including shipping costs.

Resumen del pedido	
Mundo Otaku	
Volver a la bolsa de compras	
Artículos	1
	Camiseta Inuyasha AC2362623623 + \$47.520 -\$54.000 - \$5.702,4
Talla : M	Color : 
Cupones	
KJVQE4E6	Aplicar
Cupón aplicado exitosamente	
Descuento 12%, máximo monto a aplicar: \$1.000.000 ✖	
Costo envío	\$10.000
Pago total	\$51.818

- **Interfaz de compra:** En esta imagen se puede apreciar la redirección a la ventana de ePayco para terminar la compra, en la parte derecha se puede ver el total de la compra y el nombre de los artículos, en la parte izquierda se puede visualizar las diferentes formas de pago.

Fig. 27. Interfaz compra, ejemplo 4

Transacción en modo pruebas

blocky Pago seguro por ePayco
ES / EN

1 Seleccione el medio de pago 2 Complete la información 3 Confirmación

→ Ingrese su correo electrónico para iniciar

Continuar

→ Seleccione su medio de pago preferido

Productos Davivienda
Pague Fácil, Rápido y Seguro

Tarjeta de Crédito y Débito
Pague con tu tarjeta hasta 36 cuotas

PSE (Cuentas de ahorro y corriente)
Pagos seguros en línea con su banco

DaviPlata

SU COMPRA

Camiseta Inuyasha

Total \$51,817.60 COP

¿PREGUNTAS AL COMERCIO?

Comercio BLOCKY
Llámenos 3138342394
Escribanos info@blocky.shop
<https://blocky.shop/>

PAGO SEGURO

SecureTrust

- **Interfaz de respuesta:** En esta imagen se puede visualizar el resultado de la compra, donde se aprecia el costo de envío, el costo del artículo con su respectivo descuento y el cupón que se utilizó en la compra, el valor total de la compra, el estado de la compra e información del usuario que realizó la compra.

Fig. 28. Interfaz respuesta, ejemplo 4

Detalles del pedido

Número del pedido
#109610612

Detalles del pago

Nombre Andres E Carrascal Verona	Tipo y número de documento CC 109885922	Número de órden #2512
Fecha y hora de la transacción 2022-10-20 12:31 AM	Medio de pago BANCO DE PRUEBAS	Estado de transacción Acceptada
 Camiseta Inuyasha \$41.818 \$48.298 1 Unidad		
Valor \$51.817,6	Descuento total promoción \$6.480	Costo envío \$10.000
Cupón Código: KJVQE4E6		
Descripción: Descuento del 12%, máximo descuento aplicado de \$1.000.000		

Ejemplo #5:

- **Resumen de la compra:** En este ejemplo se puede apreciar la compra de 2 artículos con descuento, el costo de los 2 artículos con descuento es de \$69.520 y con el valor del costo de envío da un total de \$91.520.

Fig. 29. Resumen de la compra, ejemplo 5

The screenshot shows a shopping cart summary for 'Mundo Otaku, Bombillas'. It lists two items: a yellow light fixture and a vintage-style Edison-style LED bulb. The total cost of the items is \$69.520, with a discount of \$54.000. The shipping cost is \$22.000, resulting in a total payment of \$91.520. There is a coupon field and an 'Aplicar' button.

Resumen del pedido	
Mundo Otaku , Bombillas	
Volver a la bolsa de compras	
Artículos	2
	+ \$47.520 -\$54.000 Color :
	Bombillo Vintage Pera Filamento 6w Tipo Edison Luz Cálida AC3532626263221 + \$22.000 -\$25.000 Forma : Globo Tecnología de iluminación : LED
Cupones	
<input type="text" value="INGRESE UN CUPÓN"/> <input type="button" value="Aplicar"/>	
Costo envío	\$22.000
Pago total	\$91.520

- **Interfaz de compra:** En esta imagen se puede apreciar la redirección a la ventana de ePayco para terminar la compra, en la parte derecha se puede ver el total de la compra y el nombre de los artículos, en la parte izquierda se puede visualizar las diferentes formas de pago.

Fig. 30. Interfaz compra, ejemplo 5

Transacción en modo pruebas

blocky Pago seguro por ePayco ES / EN

1 Seleccione el medio de pago 2 Complete la información 3 Confirmación

→ Ingrese su correo electrónico para iniciar

alguien@dominio.com Continuar

→ Seleccione su medio de pago preferido

Productos Davivienda
Pague Fácil, Rápido y Seguro

Tarjeta de Crédito y Débito
Pague con tu tarjeta hasta 36 cuotas

PSE (Cuentas de ahorro y corriente)
Pagos seguros en línea con su banco

DaviPlata

SU COMPRA

Camiseta Inuyasha, Bombillo Vintage Pera Filamento 6w Tipo Edison Luz Cálida

Total \$91,520.00 COP

¿PREGUNTAS AL COMERCIO?

Comercio BLOCKY
Llámenos 3138342394
Escribanos info@blocky.shop
<https://blocky.shop/>

PAGO SEGURO

- **Interfaz de respuesta:** En esta imagen se puede visualizar el resultado de la compra, donde se aprecia el costo de envío, el costo de los artículos con su respectivo descuento de cada artículo, el valor total de la compra, el estado de la compra e información del usuario que realizó la compra.

Fig. 31. Interfaz respuesta, ejemplo 5

Detalles del pedido

Número del pedido
#109610710

Detalles del pago

Nombre Andres E Carrascal Verona	Tipo y número de documento CC 109885922	Número de órden #2513
Fecha y hora de la transacción 2022-10-20 12:36 AM	Medio de pago BANCO DE PRUEBAS	Estado de transacción Aceptada
 Camiseta Inuyasha \$47.520 \$54.000 1 Unidad		
 Bombillo Vintage Pera Filamento 6w Tipo Edison Luz Cálida \$22.000 \$25.000 1 Unidad		
Valor \$91.520	Descuento total promoción \$9.480	Costo envío \$22.000

[Descargar](#)

Ejemplo #6:

- **Resumen de la compra:** En este ejemplo se puede apreciar la compra de 2 artículo que tiene descuento y se le aplicó un cupón, el costo de los 2 artículos con descuento es de \$69.520 y con un 15% de descuento con el cupón utilizado en los 2 artículos, da un total de \$59.092, con el valor del costo de envío da un total de \$81.092.

Fig. 32. Resumen de la compra, ejemplo 6

The screenshot shows an order summary for 'Bombillas, Mundo Otaku'. It lists two items: a light bulb and a yellow Inuyasha t-shirt. A coupon 'CUPONGENERAL' is applied, resulting in a 15% discount. The total cost including shipping is \$81.092.

Resumen del pedido	
Bombillas , Mundo Otaku	
Volver a la bolsa de compras	
Artículos	2
	+ \$22.000 → \$25.000 - \$3.300
	Camiseta Inuyasha AC2362623623 Talla : M Color :
	+ \$47.520 → \$54.000 - \$7.128
Cupones	
CUPONGENERAL Aplicar	
Cupón aplicado exitosamente	
Descuento 15%, máximo monto a aplicar: \$100.000 ✖	
Costo envío	\$22.000
Pago total	\$81.092

- **Interfaz de compra:** En esta imagen se puede apreciar la redirección a la ventana de ePayco para terminar la compra, en la parte derecha se puede ver el total de la compra y el nombre de los artículos, en la parte izquierda se puede visualizar las diferentes formas de pago.

Fig. 33. Interfaz compra, ejemplo 6

Transacción en modo pruebas

blocky Pago seguro por ePayco ES / EN

1 Seleccione el medio de pago 2 Complete la información 3 Confirmación

→ Ingrese su correo electrónico para iniciar

alguien@dominio.com Continuar

→ Seleccione su medio de pago preferido

Productos Davivienda
Pague Fácil, Rápido y Seguro

Tarjeta de Crédito y Débito
Pague con tu tarjeta hasta 36 cuotas

PSE (Cuentas de ahorro y corriente)
Pagos seguros en línea con su banco

DaviPlata

SU COMPRA

Bombillo Vintage Pera Filamento 6w Tipo Edison Luz Cálida,
Camiseta Inuyasha

Total \$81,092.00 COP

¿PREGUNTAS AL COMERCIO?

Comercio BLOCKY
Llámenos 3138342394
Escríbanos info@blocky.shop
<https://blocky.shop/>

PAGO SEGURO

- **Interfaz de respuesta:** En esta imagen se puede visualizar el resultado de la compra, donde se aprecia el costo de envío, el costo de los artículos con el respectivo descuento de cada uno y el cupón utilizado en la compra, el valor total de la compra, el estado de la compra e información del usuario comprador.

Fig. 34. Interfaz respuesta, ejemplo 6

Detalles del pedido

Número del pedido
#109610804

Detalles del pago

Nombre Andres E Carrascal Verona	Tipo y número de documento CC 109885922	Número de orden #2515
Fecha y hora de la transacción 2022-10-20 12:41 AM	Medio de pago BANCO DE PRUEBAS	Estado de transacción Aceptada

	Bombillo Vintage Pera Filamento 6w Tipo Edison Luz Cálida \$18.700 \$21.700 1 Unidad	
	Camiseta Inuyasha \$40.392 \$46.872 1 Unidad	

Valor \$81.092	Descuento total promoción \$9.480	Costo envío \$22.000
--------------------------	---	--------------------------------

Cupón
Código: CUPONGENERAL
Descripción: Descuento del 15%, máximo descuento aplicado de \$100.000

Una vez validado que las compras se aplicaran de manera correcta, el siguiente proceso fue crear la ventana de resultado de pago, la cual se implementó y arrojaba los resultados correctos en las diferentes formas de pagos que se realizaron, hasta este punto, todo el desarrollo se estaba llevando de manera satisfactoria, hasta que el cliente cambió las prioridades y todo el resto de desarrollo quedó pausado.

Con el problema que se tenía dentro de la plataforma, se presentaron varios problemas, pero, aun así, con el apoyo del cliente, el equipo de desarrollo y el apoyo del personal de ePayco, se lograron resolver todos los problemas y dudas que se presentaron durante la integración de la pasarela. A continuación, se van a presentar los diferentes puntos que describen cómo fue el proceso para obtener los resultados deseados.

Especificación de los requisitos con historias de usuario: A continuación, teniendo en cuenta los requisitos comentados anteriormente, se especifican las diferentes historias de usuario y cada actor.

- **H1:** Yo como usuario, quiero poder realizar compras de artículos en varias empresas.
- **H2:** Yo como usuario, quiero poder realizar compras de artículos en varias empresas utilizando cupones para al menos un artículo.
- **H3:** Yo como usuario, quiero poder realizar compras de artículos en varias empresas que tengan promoción.
- **H4:** Yo como usuario, quiero poder realizar compras de artículos de una sola empresa.
- **H5:** Yo como usuario, quiero poder realizar compras de un solo artículo.
- **H6:** Yo como administrador del portal, quiero poder probar compras en modo pruebas y en modo productivo.
- **H7:** Yo como administrador del portal, quiero poder gestionar los pedidos por empresa.
- **H8:** Yo como administrador del portal, quiero poder gestionar las PQR por empresas.
- **H9:** Yo como administrador, quiero poder visualizar la documentación de la pasarela de pago.
- **H10:** Yo como usuario, quiero poder visualizar el resultado de mi pago.
- **H11:** Yo como usuario, quiero poder visualizar por medio de correo electrónico las compras realizadas.

- **H12:** Yo como empresa vendedora, quiero que me llegue un correo por cada compra que realicen de mis productos.
- **H13:** Yo como empresa vendedora, quiero poder gestionar mis pedidos.
- **H14:** Yo como empresa vendedora, quiero poder gestionar las PQR que generen los compradores.

Aplicación de la metodología: Hay que recordar que la metodología empleada en este caso fue SCRUM, y fue gracias a esta, que se pudo avanzar de manera coordinada y organizada todos los desarrollos para la implementación de la pasarela de pago, a continuación, se describe cómo se fue desarrollando cada actividad respecto a la metodología.

- En la parte investigativa, se realizó una lectura sobre toda la documentación que ofrece ePayco, se fueron sacando los puntos más importantes en cuanto a los *split payments*, ya que este es el desarrollo principal a implementar, todo esto sin dejar de lado el resto de documentación, ya que por medio de esta es que se puede llegar a los *split payments*. En esta parte inicial también hubo una reunión por parte de ePayco donde explicaban de manera más detallada cómo es el flujo para utilizar *split payment* lo que ayudó bastante en la integración, pues todas las dudas que en ese entonces surgían eran solucionadas por el equipo técnico de ePayco, es gracias a todo esto que la parte investigativa se cumplió exitosamente. A continuación, se deja una imagen donde se puede visualizar la configuración realizada en ePayco para pagos divididos y la guía utilizada.

Nota: Por temas de confidencialidad, alguna información no será visible.

- De la documentación que se generó a base de los investigado, se hizo una pequeña wiki con la información principal recolectada de todo lo leído, además se documentó también las credenciales de las cuentas que se crearon tanto para prueba como para producción, finalmente se guardó el video de la reunión que se hizo con ePayco ya que ahí se explicó todo el flujo para una compra y sería de gran ayuda para cuando se esté realizando toda la implementación, por todo lo dicho anteriormente, el punto de documentación se realizó exitosamente.
- En la integración de la pasarela de pago, se presentaron problemas al inicio, pues no había literalmente una librería para Angular, pero sí para Node, por lo que tocaba manejar un archivo JS para poder utilizar la librería, lo que también generó problemas por las

credenciales de la cuenta de ePayco las cuales, luego de generar un *Ticket* a ePayco se logró configurar de manera exitosa, otro problema que se presentó en la integración, fue al generar la primera compra, lastimosamente en Google hay poca información sobre la librería lo que hace el proceso un poco complicado cuando hay dudas, lo positivo es que el personal de ePayco es muy puntual en cuanto a la respuesta de *Tickets* y reuniones personalizadas, luego de realizar varias pruebas, se soluciona el error y finalmente, el último problema presentado fue al generar una compra con 2 empresas, que luego de una reunión con el equipo de ePayco, se pudo resolver logrando integrar satisfactoriamente la pasarela de pagos. Luego de todo este proceso, se realizaron varios pagos usando productos con descuento y con cupones de diferentes maneras y al final, después de varias correcciones, el flujo quedó listo, lo que permitió finalizar esta parte exitosamente.

- En la parte de análisis y ejecución de pruebas para los ambientes de prueba y producción, solo quedaron registros de las pruebas realizadas en el ambiente de pruebas, ya que, por prioridades, el desarrollo quedó pausado y no permitió finalizar todo el proceso. Es importante aclarar que estas pruebas fueron realizadas usando combinaciones de promociones y cupones, tal como se especifica en la *Tabla 2. Ejemplos de pagos*. Teniendo en cuenta lo anteriormente mencionado, se puede decir que este ítem se cumplió parcialmente.
- En la realización de pruebas en cada escenario, no fue posible implementar del todo las pruebas debido a que el desarrollo fue pausado por prioridades, por lo tanto, este ítem no se cumplió.
- En la entrega de prototipo funcional, tampoco se pudo llevar a cabo por la pausa del desarrollo por parte del cliente.

Nota: En los anexos se encuentran pruebas de algunos de los procedimientos mencionados anteriormente.

Desarrollo por Sprint: Es importante tener en cuenta, que el proceso de integración de la pasarela de pago, también fue acompañado de otros desarrollos, aunque en el mayor porcentaje del tiempo, se estuvo desarrollando todo lo relacionado a la pasarela de pago. A continuación, se van a especificar lo realizado en cada sprint en cuanto al desarrollo de la pasarela de pago.

Tabla 5 *Objetivos cumplidos por Sprint*

Sprint	Desarrollo	Entregado	Cumplió
1	Lectura sobre la pasarela de pago	Si	Si
1	Documentación de la pasarela de pago	Si	Si
2	Análisis de arquitectura y validación de cambios en las plataformas	Si	Si
2	Integración de la pasarela de pago en Angular	Si	Si
3	Configuración para pagos únicos y divididos	Si	Si
3	Configuración de pagos con cupones	Si	Si
4	Configuración de pagos con promociones	Si	Si
4	Configuración de ventana de resultado	Si	Si
5	Refactorización de módulos necesarios para la integración	Si	Si
6	Cambio de lógica al procesar split payments	No	No
6	Cambio de visualización de información de pedidos en Datacenter	No	No
7	Cambio en la lógica de gestión de PQR por parte de los usuarios	No	No
8	Cambio en la lógica de gestión de PQR por parte de la empresa	No	No
9	Entrega del prototipo funcional en ambiente de pruebas	No	No
10	Entrega del prototipo funcional en ambiente de producción	No	No

Es necesario tener en cuenta que muchos de estos desarrollos se hicieron muy largos por uno que otro inconveniente a la hora de desarrollarlos y que todas de las actividades anteriormente nombradas, se subdividieron en varias tareas. El resto de resultados esperados en cada Sprint no se llevaron a cabo por el cambio de prioridades por parte del cliente.

Prototipo de la aplicación: El siguiente prototipo, se implementó para el ambiente de pruebas, inicialmente se puede apreciar que los usuarios ingresar a la plataforma, la cual se visualiza por medio de un *Reverse Proxy*, seguidamente el *Front End* de Blocky solicita la creación de la orden al *Back End* para así poder ser redireccionado a ePayco para realizar la compra. Finalmente, ePayco

por medio de un *WebHook* notifica al *Back End* el resultado de la compra y redirecciona a la página de Blocky para mostrar el resultado de la transacción, la información de la compra es consultada el *Back End* del aplicativo para hacerla visible al usuario.

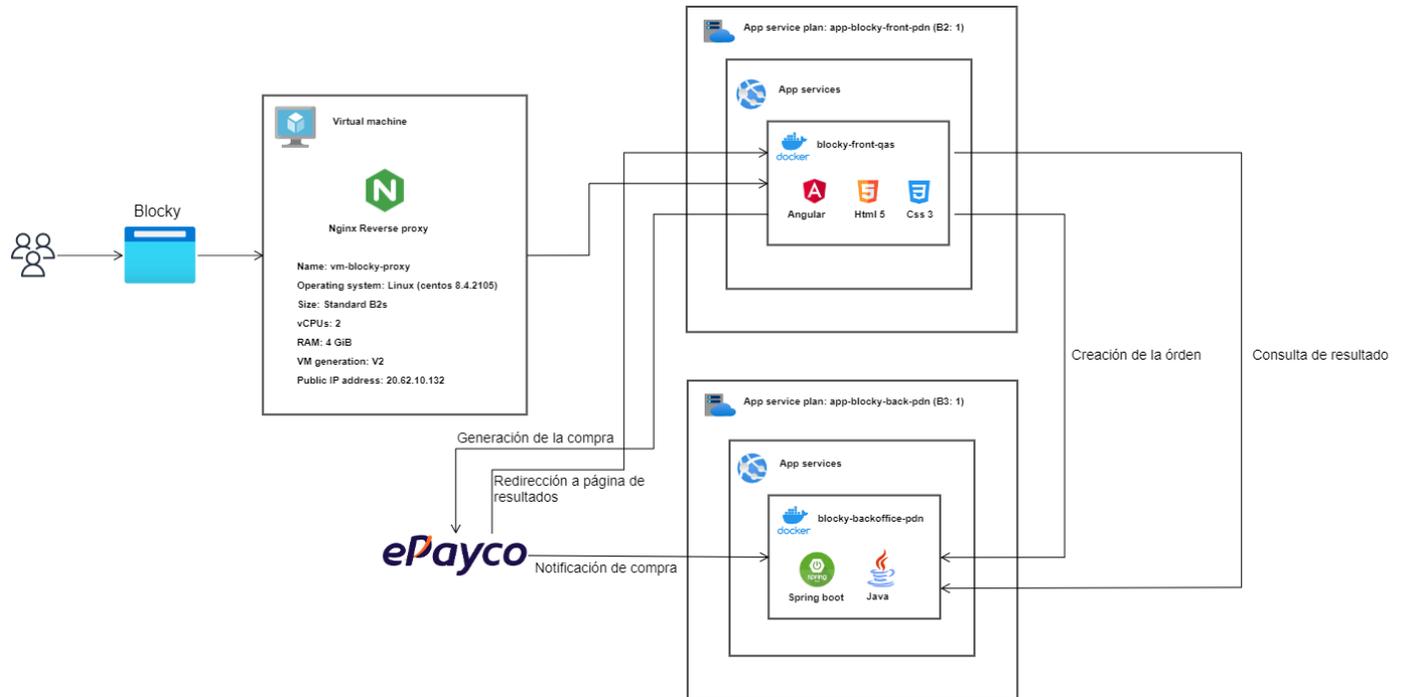


Imagen 34. Prototipo

Adicionalmente se muestra un diagrama de flujo para entender cómo funcionan los diferentes estados que se pueden presentar en el proceso de una compra.

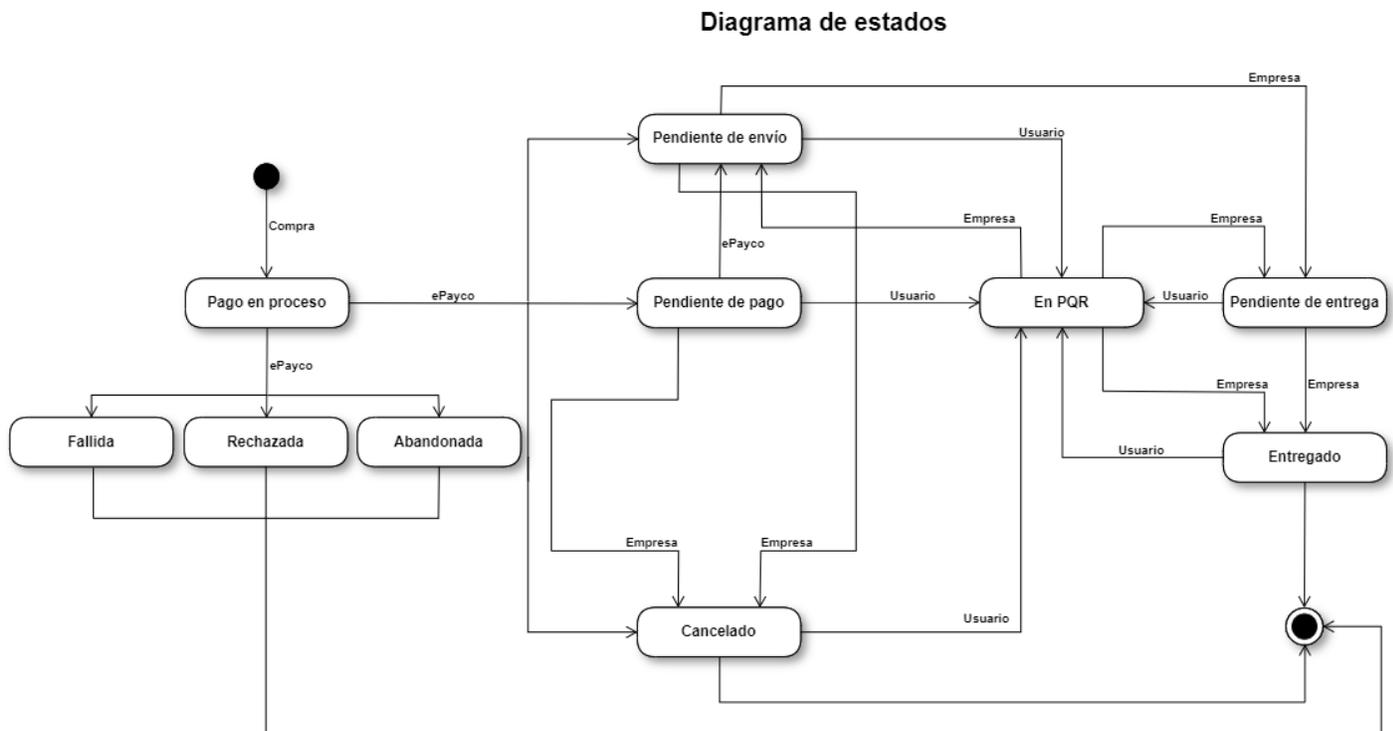


Imagen 35. Diagrama de estados

Finalmente, los resultados obtenidos dejaron una gran experiencia sobre las pasarelas de pago, comprender su importancia dentro de los *social marketplace* y cómo es la implementación que hay detrás de muchas compras que se realizan por medio de la internet, a pesar de no haberse podido terminar todo el desarrollo, los resultados hasta donde se dejó la implementación también fue muy satisfactoria para el cliente, pues se siente a gusto con todo lo sé que ha logrado hasta el momento. A pesar de que el desarrollo no se ha terminado, con lo visualizado en la utilización del ambiente de pruebas se logra apreciar una gran ventaja y facilidad de compra a comparación de los pagos que no permiten *split payment*.

4. CONCLUSIONES Y TRABAJO FUTURO

4.1. CONCLUSIONES

En conclusión, todo el desarrollo realizado antes de que cambiaran la prioridad del desarrollo se realizó de manera exitosa, el paso más difícil y complicado fue la refactorización del código para que se pudiera utilizar la funcionalidad de *split payment*, otro punto de los más complicados fue hacer que funcionaran las 2 primeras compras, pues para que se lograra hacer de manera exitosa, se tuvieron que envía varios tickets y hasta realizar una reunión con el equipo de ePayco. La investigación y documentación se logró de manera exitosa, se logró integrar de manera correcta el SDK a Angular y se logró diseñar y ejecutar un plan de pruebas para el ambiente de pruebas, todo lo anteriormente mencionado, ayudó mucho en el entendimiento del desarrollo de software y cómo funcionan los pagos en muchas plataformas hoy en día.

4.2. TRABAJO FUTURO

Los trabajo futuros se centran en poder finalizar toda la integración de la pasarela de pagos, para poder finalizar la integración hay que realizar todas las pruebas unitarias que sean necesarias para el correcto funcionamiento del aplicativo, hay que realizar un cambio en las vistas y servicios del datacenter del aplicativo, ya que este también está acoplado a la lógica para compras de una sola empresa, también hay que validar todo el proceso de notificaciones por *webhooks* para el caso en que sean compras divididas, adicionalmente toda la lógica de gestión de pedidos y de PQRS del usuario toca modificarlas, pues también está acoplada a pagos de una única empresa y finalmente y no menos importante, realizar todo este proceso para que quede totalmente funcional la implementación pero para un ambiente productivo y así poder realizar la entrega de todo el desarrollo e implementación de la pasarela de pagos.

REFERENCIAS

- [1] Blocky, «<https://blocky.shop/social/information/we-are-different>,» 30 10 2022. [En línea]. Available: <https://blocky.shop/social/information/we-are-different>. [Último acceso: 30 10 2022].
- [2] Platzi, «Platzi,» 30 10 2022. [En línea]. Available: <https://platzi.com/blog/que-es-frontend-y-backend/>. [Último acceso: 30 10 2022].
- [3] O. |. C. A. a. C. Platform, «Oracle | Cloud Applications and Cloud Platform,» 30 10 2022. [En línea]. Available: <https://www.oracle.com/co/database/what-is-database/>. [Último acceso: 30 10 2022].
- [4] T. Hostinger, «Tutoriales Hostinger,» 30 10 2022. [En línea]. Available: <https://www.hostinger.co/tutoriales/pasarela-de-pago>. [Último acceso: 30 10 2022].
- [5] WANNME, «WANNME,» 30 10 2022. [En línea]. Available: <https://www.wannme.com/que-es-y-como-funciona-el-split-payment-wannme/#:~:text=Split%20Payments%2C%20perfecto%20para%20el%20sector%20B2B&text=La%20empresa%20recibe%20el%20pago,previamente%20acordadas%20por%20las%20partes..> [Último acceso: 30 10 2022].
- [6] P. Ágiles, «Proyectos Ágiles,» 30 10 2022. [En línea]. Available: <https://proyectosagiles.org/que-es-scrum/>. [Último acceso: 30 10 2022].
- [7] Openwebinars, «Openwebinars,» 30 10 2022. [En línea]. Available: <https://openwebinars.net/blog/que-es-azure-devops/> . [Último acceso: 30 10 2022].
- [8] Hotmart, «Hotmart,» 21 09 2022. [En línea]. Available: <https://hotmart.com/es/blog/que-es-marketplace#:~:text=La%20palabra%20marketplace%20viene%20de,de%20diversas%20marcas%20o%20empresas..> [Último acceso: 30 10 2022].
- [9] InformaBTL, «InformaBTL,» 26 8 2016. [En línea]. Available: <https://www.informabtl.com/que-es-el-social-marketplace/> . [Último acceso: 30 10 2022].
- [10] Academic, «Academic,» 30 10 2022. [En línea]. Available: <https://es-academic.com/dic.nsf/eswiki/109351>. [Último acceso: 30 10 2022].
- [11] ITDO, «ITDO,» 27 7 2021. [En línea]. Available: <https://www.itdo.com/blog/que-es-node-js-y-para-que-sirve/> . [Último acceso: 30 10 2022].
- [12] M. W. Docs, «MDN Web Docs,» 3 10 2022. [En línea]. Available: https://developer.mozilla.org/es/docs/Learn/JavaScript/First_steps/What_is_JavaScript . [Último acceso: 30 10 2022].

-
- [13] LINUBE, «LINUBE,» 30 10 2022. [En línea]. Available: <https://linube.com/ayuda/articulo/195/que-es-un-ticket-de-soporte> . [Último acceso: 30 10 2022].
- [14] MDirector, «MDirector,» 2 3 2020. [En línea]. Available: <https://www.mdirector.com/blog/que-es-un-webhook/> . [Último acceso: 30 10 2022].
- [15] H. Blog, «Hackmetrix Blog,» 15 7 2021. [En línea]. Available: <https://blog.hackmetrix.com/que-es-pci-dss/>. [Último acceso: 30 10 2022].
- [16] I. D. Guide, «IONOS Digital Guide,» 30 10 2022. [En línea]. Available: <https://www.ionos.es/digitalguide/servidores/know-how/que-es-un-servidor-proxy-inverso/>. [Último acceso: 30 10 2022].
- [17] GCFGlobal.org, «GCFGlobal.org,» 30 10 2022. [En línea]. Available: <https://edu.gcfglobal.org/es/informatica-basica/que-son-las-aplicaciones-web/1/>. [Último acceso: 30 10 2022].
- [18] C. C. S. |. M. Azure, «Cloud Computing Services | Microsoft Azure,» 30 10 2022. [En línea]. Available: <https://azure.microsoft.com/es-es/resources/cloud-computing-dictionary/what-is-devops/>. [Último acceso: 30 10 2022].
- [19] MongoDB, «MongoDB,» 30 10 2022. [En línea]. Available: https://www.mongodb.com/cloud/atlas/lp/try2?utm_source=google&utm_campaign=gs_americanas_colombia_search_core_brand_atlas_desktop&utm_term=mongodb%20database&utm_medium=cpc_paid_search&utm_ad=e&utm_ad_campaign_id=12212624317&adgroup=115749712463&gclid=EAla. [Último acceso: 30 10 2022].
- [20] P. T. w. m. a. o. s. database, «PostgreSQL: The world's most advanced open source database,» 30 10 2022. [En línea]. Available: <https://www.postgresql.org/about/>. [Último acceso: 30 10 2022].
- [21] P. y. m. |. A. d. w. y. móvil, «Programación y más | Aprende desarrollo web y móvil,» 30 10 2022. [En línea]. Available: Recuperado de: <https://programacionymas.com/blog/scrum-product-backlog> . [Último acceso: 30 10 2022].