ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

Desarrollo de un prototipo de pasarela de pago para la aceptación de tarjeta de crédito y débito

PROYECTO INTEGRADOR

Previo la obtención del Título de:

Ingeniero en Ciencias de la Computación

Presentado por:
Miguel Enrique Rivadeneira Segovia
Ronny Hugo Segura Merchán

GUAYAQUIL - ECUADOR
Año: 2022

DEDICATORIA

El presente proyecto lo dedico a mi familia y a todas las personas que me apoyaron.

Miguel Enrique Rivadeneira Segovia

DEDICATORIA

El presente proyecto lo dedico a mis padres, abuelas y hermanas, por el apoyo y amor que me brindan día a día, por cada mensaje de superación que me han dado para conseguir mis objetivos. Por su apoyo incondicional, esto es dedicado para ustedes con mucho cariño y esfuerzo.

Ronny Hugo Segura Merchán

AGRADECIMIENTOS

Mi más sincero agradecimiento a toda mi familia. También agradezco al Msc. Erick Lavid por todo el apoyo ofrecido en el desarrollo de este proyecto.

Miguel Enrique Rivadeneira Segovia

AGRADECIMIENTOS

Mi más sincero agradecimiento a mi familia y profesores que me guiaron, me apoyaron en no desmayar para llegar a ser un profesional.

En especial agradezco al MSc. Erick Lavid quien dedico su tiempo en responder y brindar solución a los inconvenientes durante el desarrollo de este proyecto.

Ronny Hugo Segura Merchán

DECLARACIÓN EXPRESA

“Los derechos de titularidad y explotación, nos corresponde conforme al reglamento de propiedad intelectual de la institución; Miguel Enrique Rivadeneira Segovia y Ronny Hugo Segura Merchán damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual"

MIGUEL ENRIQUE RIVADENEIRA SEGOVIA

Digitally signed by MIGUEL ENRIQUE RIVADENEIRA SEGOVIA
Date: 2022.09.21 21:12:31 -05'00'

Miguel Enrique Rivadeneira Segovia

Firmado electrónicamente por:
RONNY HUGO SEGURA MERCHAN

Ronny Hugo Segura Merchán

EVALUADORES

Firmado electrónicamente por:
BORIS XAVIER VINTIMILLA BURGOS

Ph.D. Boris Vintimilla Burgos
PROFESOR DE LA MATERIA

Firmado electrónicamente por:
ERICK VICENTE LAVID CEDENO

MSc. Erick Lavid Cedeño
PROFESOR TUTOR

RESUMEN

La empresa CONTABILLY SAS presenta inconvenientes en el proceso de pagos, debido que la herramienta utilizada actualmente le genera costos altos por cada transacción realizada. Dado este problema, la empresa requiere implementar un método de pago en línea fiable y seguro donde sus costos por transacción sean menores, conservando la calidad del servicio. Para cubrir esta necesidad, en este proyecto se desarrolló un prototipo de pasarela de pago utilizando los servicios de *Braintree* en la cual se implementaron mecanismos de seguridad de la información para evitar comprometer los datos sensibles de la tarjeta del usuario. Adicionalmente, se utilizó un mecanismo de verificación de usuario al momento de completar una transacción. Como resultado se obtuvo un prototipo funcional de pasarela de pago, proporcionando una interfaz sencilla e intuitiva, incluyendo las medidas de seguridad para proteger la información proporcionada por los usuarios, manteniendo la calidad del servicio. La principal limitación fue la integración de las puertas de enlace Visa y MasterCard, que no proporcionaron las credenciales de comunicación, dado eso se buscó otra alternativa como Braintree para cumplir con el proceso de pago.

Palabras Clave: *Braintree,* pasarela de pago*,* seguridad de la información, verificación de usuario.

ABSTRACT

The CONTABILLY SAS company has problems with the payment process, because the currently used tool generates high costs for each transaction. Given this problem, the company needs to implement a reliable and secure online payment method where the costs per transaction are lower, while maintaining the quality of service. To cover this need, the project developed a prototype of a payment gateway using Braintree's services in which information security mechanisms were implemented to avoid compromising the user's sensitive card data. Additionally, a user verification mechanism was used at the moment of completing a transaction. As a result, a functional prototype of a payment gateway was obtained, providing a simple and intuitive interface, including security measures to protect the information provided by users, while maintaining the quality of service. The main limitation was the integration of the Visa and MasterCard payment gateways, which did not provide communication credentials, so it looked for an alternative such as Braintree to fulfill the payment process.

Keywords: *Braintree, Payment Gateway, Information Security, User Verification.*

ÍNDICE GENERAL

ABREVIATURAS

AES	*Advanced Encryption Standard*
API	*Application Program Interface*
AVS	*Address Verification System*
CVV	*Card Verification Code*
DSS	*Data Security Standards*
HTTPS	*Hyper Text Transfer Protocol Secure*
IDEA	*International Data Encryption Algorithm*
KISS	*Kiss It Simple, Stupid*
MDES	*MasterCard Digital Enablement Service*
PAN	*Personal Account Number*
PCI	*Payment Card industry*
RC2	*Ron's Code*
RC4	*Rivest Cipher 4*
RC5	*Rivest Cipher 5*
RSA	*Rivest–Shamir–Adleman*
SSL	*Secure Socket Layer*
TIC	Tecnología de Información y Comunicación
VTS	*Visa Token Service*

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

CAPÍTULO 1

1. INTRODUCCIÓN

Las pasarelas de pago representan un sistema que permiten realizar pagos electrónicos entre una tienda *online* o *e-commerce* y entidades bancarias aplicando métodos de seguridad para proteger la información.

En este capítulo se detallará la problemática, proceso y TIC para el desarrollo de una pasarela de pagos. A su vez, se detallará los requisitos de seguridad que se necesitan para restringir el uso malicioso de información.

1.1 Descripción del problema

El comercio electrónico ha sido impulsado a raíz de la pandemia, este servicio permite la compra y venta de productos o servicios a través de medios digitales, de manera eficiente y segura. Debido a esto, muchas empresas han optado por utilizar servicios de pago existentes en el mercado, sin embargo, el uso de estos servicios supone un costo adicional debido al pago de comisiones por uso y por transacción realizada.

Para ofrecer un servicio de pagos eficiente, seguro y flexible, la empresa CONTABILLY SAS adquirió uno de los servicios de la pasarela de pagos *PayPal*. Sin embargo, la utilización de este servicio supone costos, los cuales incluye una comisión fija de 2.44 dólares americanos por transacción efectuada, más el 4.09 % del monto facturado. La empresa ha notado que, aunque el servicio sea eficaz y seguro, las ganancias por venta se ven reducidas.

1.2 Justificación del problema

El comercio electrónico en Ecuador está incrementando rápidamente, debido a que los negocios y empresas se ven en la necesidad de innovar y crear nuevos métodos para comercializar sus productos o servicios [1]. El comercio electrónico se caracteriza por permitir la compra y venta de productos y servicios a través de

medios digitales con métodos de pago que garanticen seguridad a los clientes quienes ingresan información de sus tarjetas de crédito o débito.

En el mes de abril del 2021, en Ecuador, se registraron 6.4 millones de transacciones con tarjetas de débito, lo que representa un crecimiento del 133% con respecto al mismo mes del año 2020. Por otro lado, las transacciones realizadas con tarjetas de crédito ascendieron a 15,3 millones a marzo de 2021, lo que representa un aumento anual del 29% [2]. Esto muestra un aumento significante en el uso de tarjetas de crédito y débito como métodos de pago.

Para que el comercio electrónico sea completamente funcional, este debe contar con un servicio de pasarela de pago que ofrezca varios métodos para completar una compra, como también la seguridad de la información del cliente. Sin embargo, estos servicios tienen costos, los cuales incluyen un pago mensual por su utilización y una comisión por cada transacción ejecutada. Lo que reduce las ganancias de las empresas por cada venta procedida.

La empresa CONTABILLY SAS es propietaria de la aplicación JamaSana, la cual ofrece servicios de venta de comidas *online* por medio de suscripciones usando la pasarela de pago PayPal, aceptando los costos que ello implica. Debido a la alta competitividad en el mercado actual, la empresa ha decidido reducir costos, entre los cuales están los valores de la pasarela de pagos. Como consecuencia, requieren una solución que reemplace la pasarela de pago actual, donde sus costos sean menores sin perder la calidad del servicio.

1.3 Objetivos

1.3.1 Objetivo General

Implementar un prototipo de pasarela de pago que acepte tarjetas de crédito y débito emitidas por Visa y *MasterCard* utilizando servicios de pago *online*, que se integre a la aplicación móvil JamaSana de la empresa CONTABILLY SAS, permitiendo el cobro de los productos que se ofrecen en el aplicativo.

1.3.2 Objetivos Específicos

1. Crear un componente gráfico con la identidad corporativa de la empresa que reciba la información de la tarjeta del cliente.
2. Implementar mecanismos de seguridad para resguardar la integridad de la información del cliente y enviarla de forma segura.
3. Almacenar información de las transacciones realizadas en una base de datos no relacional.

1.4 Módulos del proyecto

El prototipo comprenderá dos módulos a desarrollar, los cuales permitirán brindar un mejor control y administración de la transacción de pago con tarjeta de crédito o débito. Dichos módulos comprenden la integración de un formulario de pago en el aplicativo y un API que se encargara de la gestión con los procesos de pago, tal como se muestra en la Figura 1.1. En los siguientes apartados se detallará cada módulo dentro del prototipo a desarrollar.

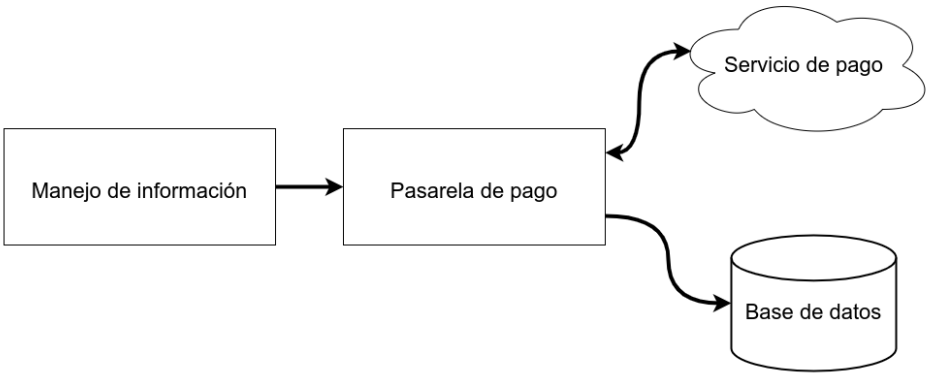


Figura 1.1 Diagrama de bloques del prototipo. [autoría propia]

1.4.1 Módulo 1: Manejo de información

En este módulo se realizará la conexión con el aplicativo JamaSana, se construirá un componente gráfico que contenga un formulario de pago en la cual el cliente pueda ingresar los datos de su tarjeta. Debido a que este módulo manejará datos sensibles, se aplicarán medidas de seguridad de la información como lo es el

cifrado de datos. Por último, se establecerá una conexión segura con el Módulo 2: Pasarela de pago, para asegurar el envío de los datos cifrados del cliente.

1.4.2 Módulo 2: Pasarela de pago

Este módulo comprende la implementación de una capa de comunicación que permita una conexión segura entre el manejo de información y la pasarela de pago, para ello se implementara métodos de seguridad como el protocolo SSL para recibir el token de pago usando un cifrado de llaves para ser enviada al administrador de pago.

Incluirá una base de datos para almacenar el modelo de la API. Esta base de datos servirá para almacenar la transacción de pago proveniente del aplicativo y las transacciones del proceso de pago obtenidas de la comunicación con el p*ayment gateway*. Además, contará de dos servicios que abarcan la lógica y modelo del prototipo a desarrollar, los cuales implican la gestión con el usuario del aplicativo y la administración de pagos:

- La gestión de usuario es la encargada de procesar la información del pago y autorizar al usuario del proceder del mismos.

- La administración de pagos comprenderá la lógica del funcionamiento del proceso de pagos, es la encargada de la comunicación con los servicios del *gateway* para transmitir de forma segura las transacciones. Esta lógica comprende las transacciones de cada proceso de pago, los cuales van desde la autorización hasta la captura y acreditación del detalle de pago a la cuenta del cliente. La pasarela de pago cifra la información antes de proceder a la comprobación final.

1.5 Marco teórico

El comercio electrónico hace uso de TIC por lo que están en constante desarrollo, el uso de estas tecnologías es necesaria para comprar y vender servicios en tiendas

online. El oportuno desarrollo de estas tecnologías, permiten que una tienda *e-commerce* esté en constante crecimiento y llegue a mejorar su alcance sin necesidad de inversiones debido al proceso de pago.

Uno de los aspectos importantes corresponde al proceso de pagos, como se muestra en la Figura 1.2, donde se muestra la ejecución que se sigue al adquirir un producto por internet.

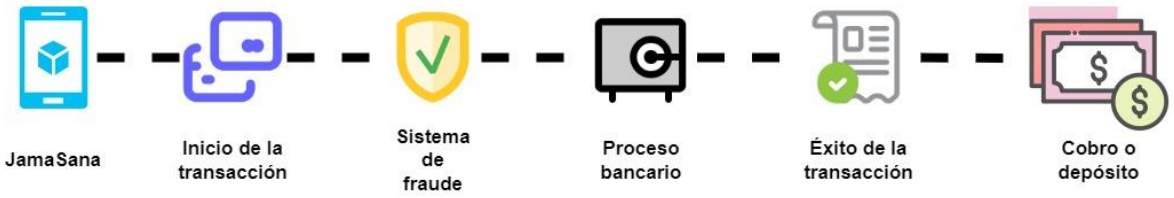


Figura 1.2 Proceso de transacción de pago. [autoría propia]

Para que una tienda *e-commerce* llegue a tener éxito, depende de la herramienta de pago que utilice para adaptarse al modelo de negocio y así lograr que el cliente confié en el comercio proporcionado. Para lograr esto se debe cumplir con las necesidades de seguridad que son requeridas para completar el proceso de pago, necesidades tales como: la privacidad de la información y protección de la información, para ello se debe considerar el uso de mecanismos de encriptación, certificados y protocolos, para lograr una transacción segura.

A continuación, se detallarán los conceptos necesarios para cumplir con el desarrollo del proyecto.

1.5.1 Pasarela de pago o *gateway*

Es una herramienta que efectúa la transmisión de los datos de las compras realizadas en su tienda *e-commerce* de manera segura en el momento del *checkout* **[3]**. Se considera al medio que permite integrar en un solo módulo todos los procesos del flujo de pago, protegiendo los datos mediante las regulaciones de

seguridad necesarias. Además, se encargan de la autorización de un pago sin tarjetas presentes en una transacción.

Dada la importancia de las pasarelas de pago, es necesario seleccionar un proveedor de pasarela de pago que se ajuste al modelo de negocio de la empresa. Los principales proveedores de pasarelas de pago son:

- *Braintree*
- *MasterCard*
- *PayPal*
- *Visa*

1.5.1.1 Braintree

Herramienta tecnológica proporcionada por PayPal que facilita la autorización de pagos en el aplicativo, aceptando deferentes métodos de pagos como tarjetas de crédito y débito. Ofrece un servicio que remplaza la forma de obtener una pasarela de pago [4].

1.5.1.2 MasterCard

Puerta de enlace que maneja funciones de pago para la aceptación de métodos locales y globales de pago, brindando recursos y seguridad en el mundo de pagos digitales [5].

1.5.1.3 PayPal

Herramienta tecnológica de pago que proporciona conveniencia y seguridad en los servicios financieros [6], haciendo los comercios electrónicos más convenientes para que las empresas prosperen en la economía global.

1.5.1.4 Visa

Es una API que ofrece servicios para la transferencia y control de transacciones de fondos. Ofrece administrar e inspira comenzar las tendencias en soluciones

globales, que permitan crear una forma sencilla e impulsar el comercio electrónico [7].

1.5.1.5 Tipos de gateway online

Es un método que permite la conexión directa entre pago y producto. Considerado como uno de los tipos más usado debido a las transacciones sin tarjetas presentes para la compra de un producto, además es la encargada de aceptar o rechazar las compras de los usuarios.

En la actualidad se pueden considerar como tipos de *gateway online* a:

- Tarjetas de crédito/debito: haciendo referencia a *Visa* y *MasterCard*
- *PayPal*
- *e-wallet*

1.5.1.6 Integración de gateway atreves de API

Solución que permite personalizar la totalidad de la interfaz del proceso de pago con la experiencia del usuario, lo que hace aumentar la capacidad de integración en medios digitales [8].

1.5.2 Adquiriente

Se denomina adquiriente a la empresa encargada de implementar y configurar el servicio de pago. Su función es recibir la información del pago para luego transmitirla al *payment gateway* y comunicar la aprobación o rechazo [3].

1.5.3 Seguridad en las pasarelas de pago

Es el método para considerar a la hora de proteger el sistema de ataque que atenten con la integridad y confidencialidad de los datos a la hora de procesar un pago.

Utiliza protocolos que permiten una conexión segura, cifrando los datos entre la comunicación cliente – servidor. Esta estrategia evita que los datos sean interceptados por terceras personas [9].

Dentro de esta estructura se encuentran los siguientes términos.

- **Certificado PCI DSS**: Son normas regulatorias que comprenden estándares de seguridad, denominado *Payment Card Industry Data Security Standards* (PCI DSS), encargados de proteger al consumidor y la empresa de fraudes [10]. Estos estándares no permiten que se guarden detalles de tarjetas.

- **Protocolo SSL**: Por sus siglas *Secure Socket Layer*, es el encargado de la privacidad de la información usando métodos de encriptación y facilitando la autenticación. El SSL se ejecuta en una capa entre los protocolos de aplicación como el *Hyper Text Transfer Protocolo* (HTTP) [9].

- **Protocolo HTTPS:** Este protocolo permite una autenticación digital en aplicaciones, además hace uso de claves públicas que permite la seguridad de los servicios web para poder identificarse [9].

- **Encriptación:** Técnica empleada para cifrar un dato o información que necesita ser intercambiada para que una persona la pueda leer con los métodos de descifrado necesarios. El uso de esta técnica permite asegurar la confidencialidad, integridad y autenticación de la información [9]. Tales como VTS y MDES, tecnologías de seguridad propias de *Visa* y *MasterCard* que reemplazan la información, con un identificador digital único llamado token [11].

- **Algoritmo simétrico:** Son algoritmos que permiten descifrar la información, dentro de estos algoritmos se encuentran algoritmos AES, IDEA, RC2, RC4 y RC5 [9].

- **Algoritmos asimétricos:** Son algoritmos basados en llaves públicas y privadas generadas por pareja, tales ejemplos son: RSA, *Diffie-Hellman* [9].

1.5.4 Tokenización

Es considerado como un identificador, permite remplazar los detalles confidenciales de un mensaje por un código alfanumérico único para no comprometer la información. Esto se usa con el propósito de proteger la información de cuentas bancarias o detalles de tarjetas de crédito que son manejados por procesos de pagos [12].

Se pueden tokenizar de varias formas sea por criptografía matemática, como un hash o como un numero aleatorio generado.

En una transacción de pagos donde se usa tarjetas de crédito, el contenido del token está comprendido entre los cuatro últimos dígitos del número de la tarjeta y los detalles de la información del dueño de la tarjeta con los datos de la transición.

1.5.5 Sistemas antifraude

Es un sistema encargado de analizar los patrones y comportamientos del usuario con el fin de identificar actos de fraude [3]. Este proceso consiste en recopilar la información del usuario y comparar los patrones para poder aprobar o no el pago del producto.

Una medida que emiten las *payment Gateway* es la comprobación de AVS/CVV en tiempo real y su autenticación *3D Secure*, con el fin de proteger el negocio de acciones fraudulentas [10].

1.5.6 Banderas de tarjeta de crédito o débito

Se considera bandera a la empresa que emite las tarjetas de crédito o débito y responsables del modelo de negocio, “Son quienes definen los estándares por los cuales los adquirientes deben procesar las transacciones realizadas por ese medio de pago (cada marca tiene su propia regla)” [3].

1.5.7 Pagos con tarjetas de crédito y débito

Son considerados el método más común de pago dado su uso generalizado para poder comprar *online*. La mayoría de las tiendas de comercio electrónico aceptan este medio de pago.

Al comprar online, los datos más generales solicitados son: nombre del titular de la tarjeta, fecha de vencimiento de la tarjeta y código CVV [13].

La principal diferencia entre una tarjeta de débito y una tarjeta de crédito es la forma de pago [14]. Es decir, con tarjeta de débito se limita el cobro según los fondos, mientras que con las tarjetas de crédito no hay restricciones y es posible que se prolongue el cobro hasta el siguiente mes.

1.5.8 Bancos emisor y banco receptor

El banco emisor es la entidad que emitió la tarjeta del usuario que precederá a pagar, encargada proporciona la información de la tarjeta de crédito o débito y el banco receptor es la entidad encargada de recibir el pago.

CAPÍTULO 2

2. METODOLOGÍA

En este capítulo se presentarán los requerimientos funcionales y no funcionales del sistema, las herramientas que se usarán para su implementación, los procesos que conforman la solución con el fin de cumplir con las etapas descritas en la Proceso de transacción de pago. y se describirán las pantallas principales del prototipo de baja calidad.

2.1 Análisis y diseño del sistema

Para el diseño del sistema fue necesario establecer y conocer, de forma clara, las necesidades del cliente y sus expectativas sobre el producto final. Para lograr este objetivo, se realizaron varias reuniones con el cliente, de las cuales se obtuvo información sobre las funciones principales del sistema y cómo estará compuesto. A continuación, se presentarán las funcionalidades generales del sistema y también las específicas para cada módulo de desarrollo propuesto.

2.1.1 Cifrado de información

Debido a que se manejará información sensible del usuario en el sistema, se procederá con la integración de medidas de seguridad de la información en ambos módulos. Siguiendo el estándar sugerido por al PCI DSS se utilizará un algoritmo de cifrado asimétrico RSA con una longitud de 4096-bits. Se utilizará la herramienta *OpenSSL* para la creación del par de llaves pública y privada que servirán para realizar el cifrado y descifrado de la información respectivamente.

Para usar este algoritmo, la llave pública será almacenada en el entorno de desarrollo del Módulo 1: Manejo de información, y se utilizará siempre que el cliente realice una nueva transacción. Por otro lado, la llave privada será almacenada en el entorno de desarrollo del Módulo 2: Pasarela de pago, y se utilizará para descifrar información cada vez que el cliente realice la petición para iniciar una nueva transacción de pago.

2.1.2 Diseño del módulo 1: Manejo de información

Este módulo se encargará de manejar la información recibida por el aplicativo y la información de la tarjeta del cliente de forma segura. Se incluirá una verificación con el Módulo 2: Pasarela de pago, para comprobar que la aplicación este aprobada para el uso del sistema. La Figura 2.1 muestra en detalle los procesos internos propuestos para este módulo y sus respectivas conexiones.

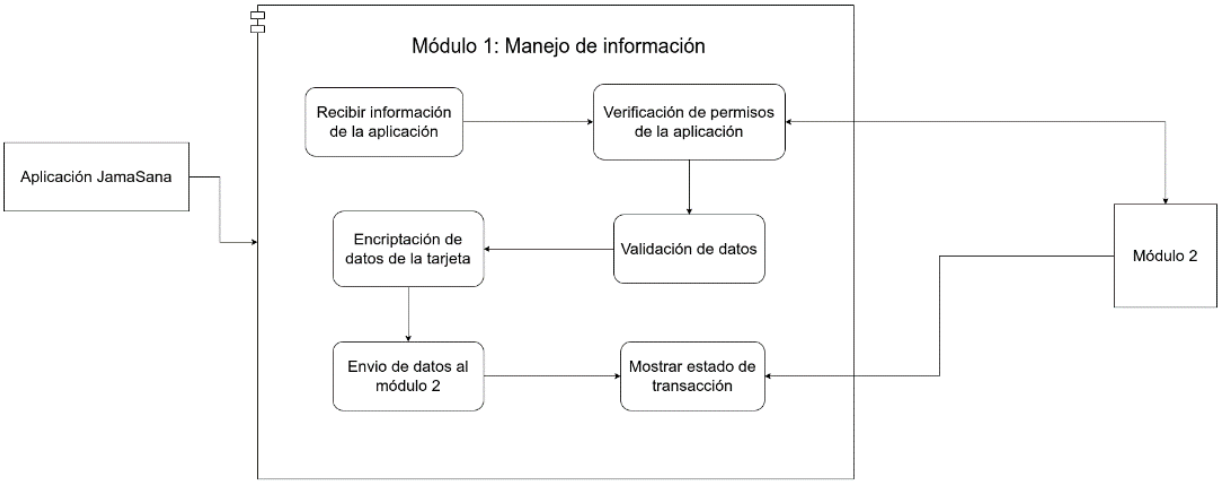


Figura 2.1 Diagrama del módulo de manejo de información. [autoría propia]

Debido a que este módulo contendrá un formulario, se realizarán validaciones para cada campo. Las reglas para las validaciones son las siguientes:

- **Nombre del titular:** Solo se debe ingresar nombre y apellido.
- **Fecha de expiración:** La fecha no debe ser anterior a la fecha en la cual se realiza la transacción.
- **Código CVV:** El código debe contener únicamente 3 cifras.
- **Número de tarjeta:** Para validar este campo se utilizó el algoritmo de *Luhn*[1], que consiste en una verificación simple de un número de tarjeta.

[1] https://www.geeksforgeeks.org/luhn-algorithm

2.1.2.1 Requerimientos funcionales

A continuación, se presentan los requerimientos para el desarrollo del Módulo 1: Manejo de información.

- El sistema recibirá la siguiente información por parte del aplicativo:
  - Monto por pagar
  - Nombre del cliente
  - Nombre de la aplicación
- El sistema contará con un formulario para ingresar los datos de la tarjeta del cliente, el cual recibirá la siguiente información:
  - Nombre del titular de la tarjeta
  - Número de tarjeta
  - Fecha de expiración
  - Código CVV
  - Correo electrónico
- El sistema aceptará únicamente tarjetas emitidas por *Visa* y *MasterCard*.
- La información de la tarjeta del cliente será cifrada utilizando mecanismos de seguridad de la información.
- El sistema deberá pedir al cliente el código de verificación para completar la transacción.
- La información ingresada en el formulario se enviará vía HTTPS al Módulo 2: Pasarela de pago.

2.1.2.2 Prototipo

Se procedió a crear un prototipo de bajo nivel desarrollado en *Figma app*[2], el cual fue presentado al cliente para su respectivo análisis. A continuación, se presentarán las dos pantallas importantes, las demás pantallas se pueden observar en el APÉNDICE B y la aceptación por el cliente en el APÉNDICE C.

[2] About Figma, the collaborative interface design tool.

La pantalla principal cuenta con el logo del sistema y un formulario acompañado con entradas de texto para que el usuario ingrese los detalles de la tarjeta, además usuario podrá escoger el tipo de tarjeta con la que desea pagar.

El formulario de pago cuenta con dos temas: modo claro y modo oscuro, los cuales se presentan en la Figura 2.2 y Figura 2.3

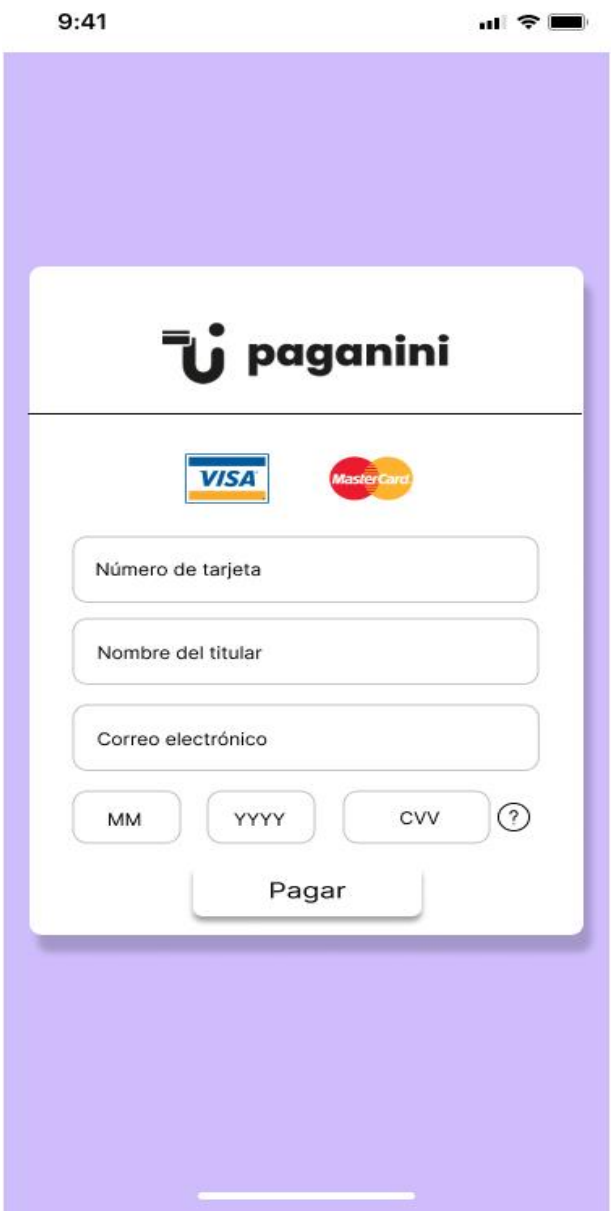


Figura 2.2 Pantalla de pago modo claro. [autoría propia]

Figura 2.3 Pantalla de pago modo oscuro. [autoría propia]

2.1.3 Diseño del módulo 2: Pasarela de pago

Dentro de este módulo se realizarán todas las validaciones y verificaciones antes de ejecutar los procesos envueltos en una transacción financiera *online*. Este módulo se encargará de las conexiones con el servicio de pago y de almacenar el

estado de las transacciones en función del proceso ejecutado. La Figura 2.4 muestra los procesos y conexiones correspondientes a este módulo.

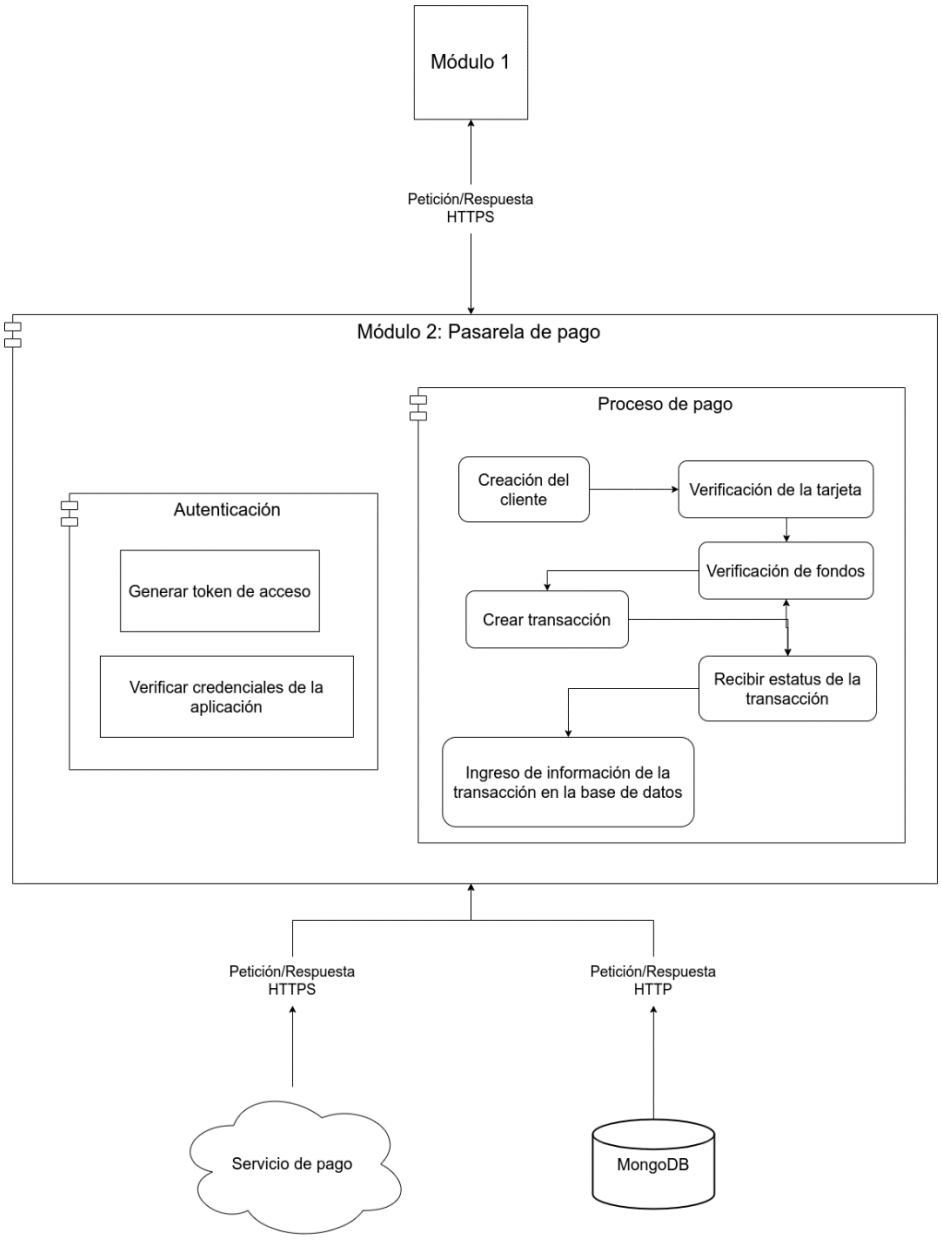


Figura 2.4 Diagrama del módulo pasarela de pago. [autoría propia]

2.1.3.1 Requerimientos funcionales

Para el desarrollo del Módulo 2: Pasarela de pago, se deben tomar en cuenta los siguientes requerimientos:

- El sistema contará con una tabla de códigos de respuesta y su respectiva descripción que indicaran el estado de la transacción. Los códigos se detallan en el APÉNDICE A.
- El sistema utilizará las herramientas del servicio de pago para realizar la validación de la información de la tarjeta del cliente. En caso de que sea inválida, se devolverá una respuesta indicando el error.
- El sistema contará con un mecanismo de autenticación y autorización del aplicativo.
- El sistema contará con un sistema de verificación de usuario utilizando el envío de mensajes de texto al cliente del aplicativo.
- Se utilizará las herramientas del servicio de pago para verificar que las tarjetas cuenten con fondos suficientes para completar la transacción. En caso contrario, el sistema responderá con el código de error correspondiente.
- Luego de completar las verificaciones y la transacción con el banco receptor, la aplicación guardará la siguiente información en la base de datos:
    - Nombre del cliente
    - Monto por pagar
    - Nombre de la aplicación
    - Estatus de la transacción (éxito o error)
- Cuando se complete todo el proceso de la transacción el sistema enviará una respuesta con el código de éxito. En caso de que ocurra un error en algún proceso independiente que detenga el flujo de la transacción, se almacenarán los mismos datos descritos anteriormente con un estatus de error.

2.1.3.2 Propuestas de diseño de seguridad para el sistema

Proceso de autenticación y autorización

Para el proceso de autenticación se utilizará el *framework* de autenticación básica de HTTP, mediante el cual se recibirá la información de las credenciales

del aplicativo por medio de una petición HTTPS de tipo POST. Esta petición deberá contener las credenciales del cliente unidas por el carácter ":" y codificadas en *base64* utilizando la palabra clave *Basic* para indicar que se realizará la autenticación. Las credenciales y la palabra clave mencionada anteriormente se recibirán en la cabecera de *Autorization*.

Luego de verificar la autenticidad del aplicativo, se utilizará el *framework* de autorización *OAuth 2.0*, el cual devolverá una respuesta con la información de acceso al sistema. Esta información corresponde a:

- ***Token* de acceso:** *Token* con el cual el aplicativo podrá conectarse al sistema.
- **Tipo de *token*:** Tipo del *token* utilizado en el *framework*. El tipo de *token* es *Bearer*.
- **Expiración:** Tiempo expresado en segundos que define el tiempo de validez del *token* de acceso.

Proceso de verificación de usuario

Este proceso consistirá en la verificación del usuario del aplicativo JamaSana. Se implementará una verificación por medio de mensaje de texto en el cual el usuario recibirá un mensaje con un código de verificación el cual será ingresado en el Módulo 1: Manejo de información. Esta verificación servirá para confirmar el proceso de transacción y realizar el respectivo cobro.

Este proceso busca simular los servicios provistos por los sistemas antifraude que se pueden utilizar desde los servicios de pago de *Visa* y *MasterCard*. De esta manera, se buscar agregar una capa más de seguridad y confiabilidad en el sistema para el usuario.

2.1.3.3 Modelo físico

El modelo físico que se muestra en la Figura 2.5 Diagrama físico NoSQL., corresponde al diseño de la base de datos no relacional. Se eligió este tipo de

base de datos por las características del sistema y la capacidad de ser rápido y fácil de trabajar.

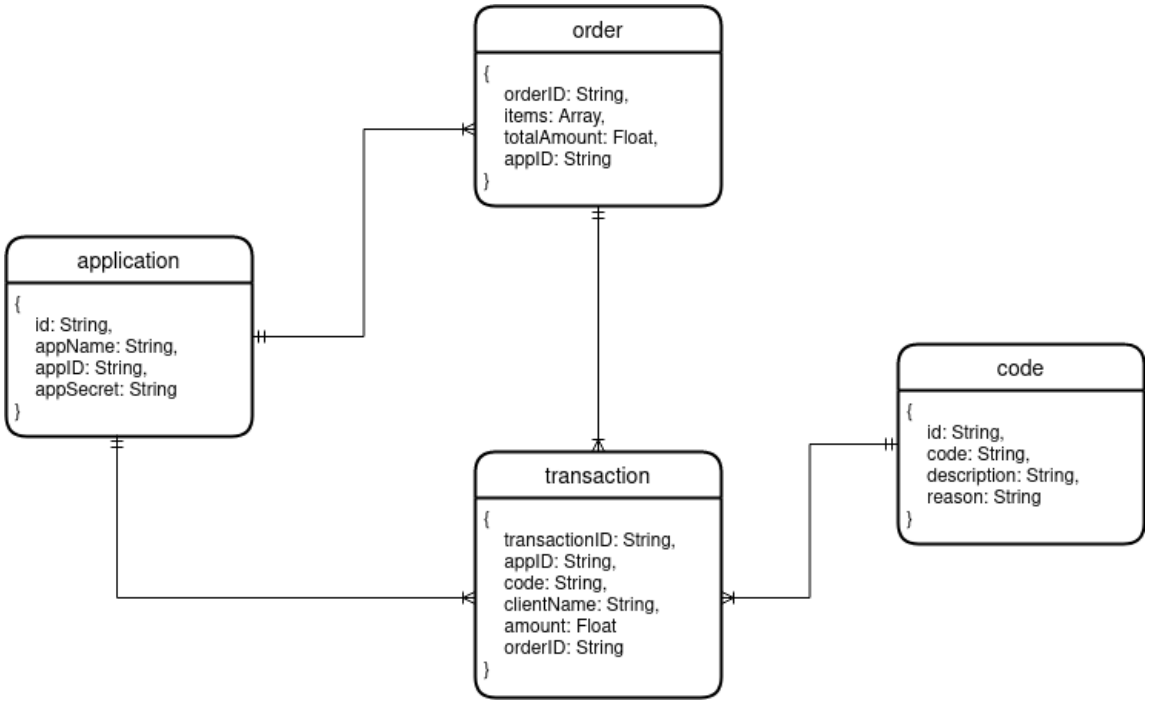


Figura 2.5 Diagrama físico NoSQL. [autoría propia]

Las colecciones mostradas en el modelo físico de la Diagrama físico NoSQL. permitirán obtener consultas de documentos en función de algún identificador, para gestionar cada transacción que se realice. Las cuales se detallan como:

- La colección *Order*, tendrá un ID, un array con los ítems adquiridos por el cliente, el precio total de la orden y el ID de la aplicación a la que pertenece la orden.
- La colección *Application*, tendrá un ID, el nombre de la aplicación y un secreto que corresponde a las credenciales de acceso al sistema.
- La colección *Transaction* tendrá un ID, el ID del aplicativo a la que pertenece la transacción, el código de estado de la transacción, el nombre del cliente, el monto total y el ID de la orden a la que corresponde dicha transacción.

- La colección *Code* tendrá un ID, el código de estado, un ID que corresponda a un objeto de la colección Código, una descripción y la razón del código.

2.1.4 Alcance y limitaciones del proyecto

2.1.4.1 Alcance del proyecto

El presente proyecto consiste en desarrollar un prototipo funcional de una pasarela de pagos para la empresa CONTABILLY SAS. Este prototipo solo aceptará el uso de tarjetas de crédito y débito de Visa o *MasterCard*.

Este prototipo cuenta con un *frontend* y un *backend*. El *frontend* contendrá un formulario para el ingreso de la información de la tarjeta del cliente y funcionará únicamente para aplicaciones móviles que hayan sido desarrolladas bajo el lenguaje de programación *Dart* con el *framework Flutter*, en este caso será integrado a la aplicación JamaSana perteneciente a la empresa.

La seguridad dentro de las pasarelas de pago es importante y necesaria para manejar la información del cliente de forma segura, en este prototipo se manejará únicamente la seguridad de la información al momento de adquirirla y al enviarla por medio de internet.

2.1.4.2 Limitaciones del proyecto

- Acceso a la API del Banco receptor escogido por el cliente.
- Cumplimiento de las normas PCI referentes a seguridad en infraestructura.
- Acceso a los servicios y las API de las redes de tarjetas *Visa* y *MasterCard*.

2.1.5 Diagramas de solución del proyecto

2.1.5.1 Diagrama de secuencia

En el siguiente diagrama que se presenta en la Figura 2.6, se detalla el flujo de acciones que se realizan al intercambiar información entre los distintos actores y

objetos que implican el sistema, siguiendo las acciones detalladas a continuación:

- Para acceder al servicio, el aplicativo deberá tener un token de acceso, el cual permite mostrar el formulario dentro del aplicativo.
- Si el aplicativo no posee las credenciales, el sistema presentara una pantalla de error.
- Después de que el sistema presente el formulario, el usuario ingresará los datos de la tarjeta y el sistema procederá a comunicarse con el servicio de pago para iniciar la transacción de pago.
- Si el usuario ingresa datos erróneos de la tarjeta como numero incorrecto, fecha expirada o código invalido, el servicio de pago retornará una respuesta de error y se le presentará en el aplicativo un mensaje con el error correspondiente.
- Si los datos son correctos el servicio de pago retornará una respuesta de éxito, el sistema realizará una petición para autorizar el pago.
- Para reducir riesgos de fraude, el sistema enviara un código mediante canal SMS al cliente del aplicativo quien realiza el pago.
- Si el usuario no ingresa el código que se le envió, el sistema procederá a almacenar la acción de compra y responderá con un mensaje de transacción cancelada.
- Si el usuario ingresa un código incorrecto, se guardará la acción de compra y el sistema responderá con un mensaje de transacción rechazada.
- Si el código es válido el sistema procederá a realizar una petición al servicio de pago para realizar la captura del monto, si la tarjeta no tiene fondos suficientes, el servicio de pago responderá con una respuesta de error al sistema y el sistema presentará un mensaje de transacción rechazada.
- Si la tarjeta tiene fondos, el servicio de pago procederá a iniciar la transacción y acreditar el monto en la cuenta del cliente. El servicio de

pago retornará una respuesta de éxito el cual servirá para presentar una notificación de orden aceptada en el aplicativo.

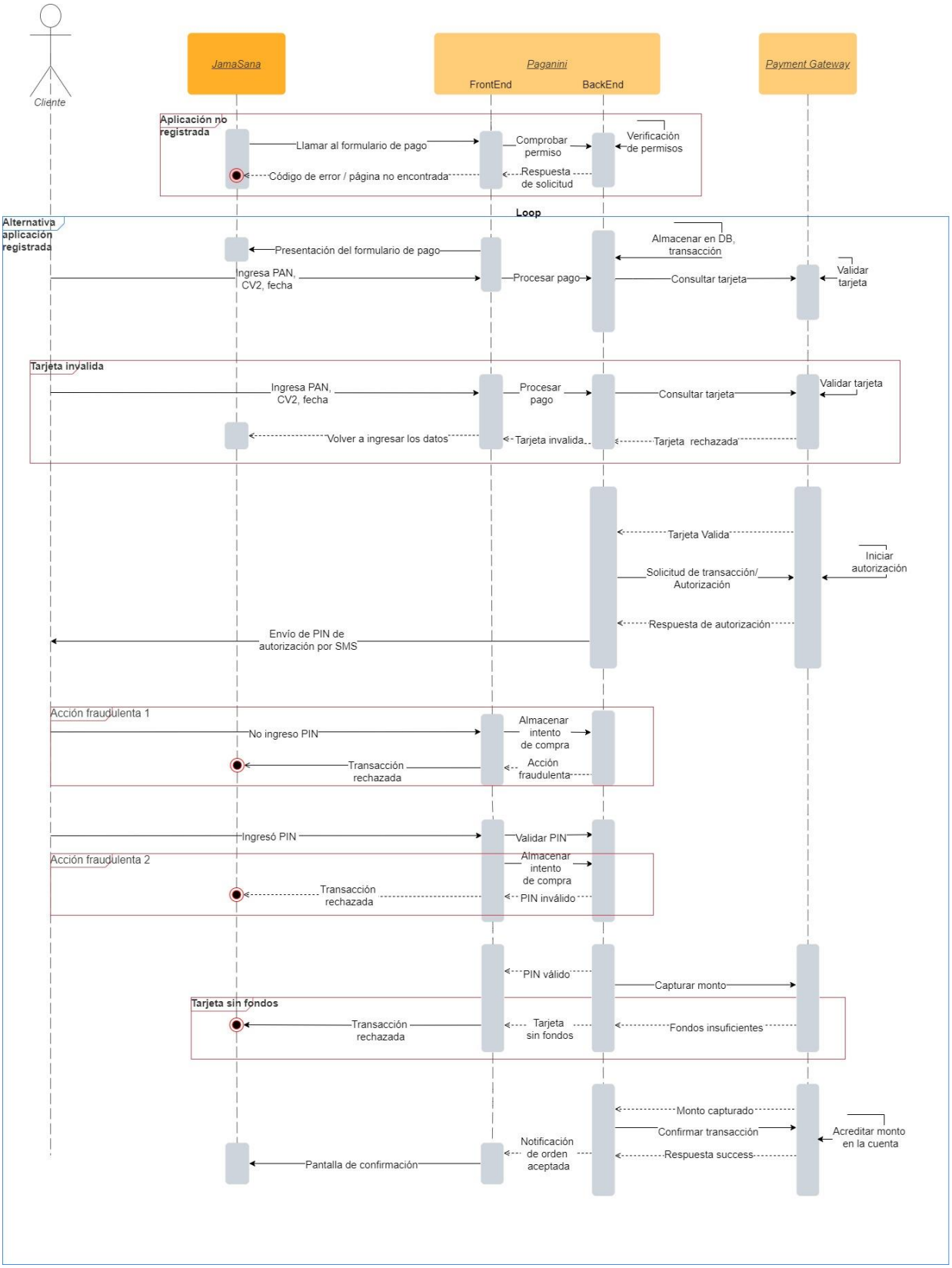


Figura 2.6 Diagrama de secuencia de la solución. [autoría propia]

2.1.5.2 Diagrama de despliegue

El diagrama presentado en la Diagrama de despliegue.Figura 2.7 presenta la arquitectura de *software* y *hardware* que se utilizará para el desarrollo del prototipo funcional. Se muestra el tipo de *hardware* y *software* sobre el cual el sistema trabajará sin inconvenientes. En este diagrama se pueden notas las siguientes características:

- El *frontend* se ejecutará sobre una aplicación móvil en los sistemas operativos *Android* e *IOs*.
- El *backend* estará alojado en la nube, y se ejecutará sobre un sistema con Ubuntu como sistema operativo.
- El *backend* realizará peticiones al servicio de pago y al servicio de MongoDB.

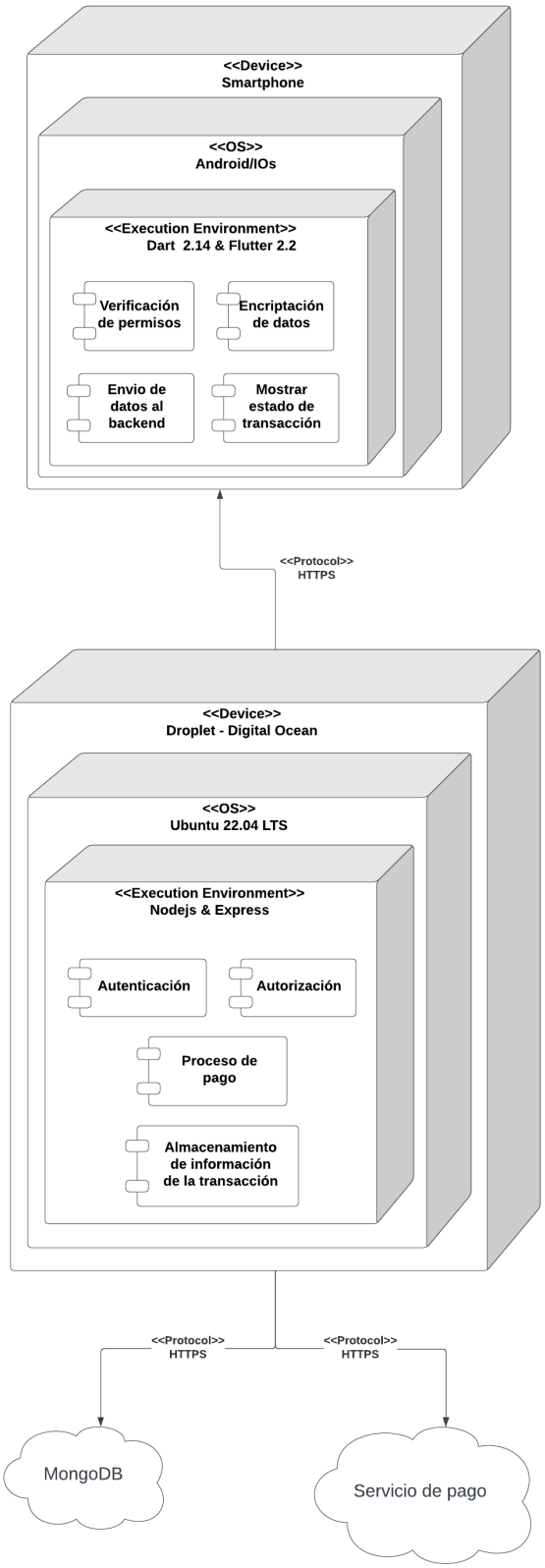


Figura 2.7 Diagrama de despliegue. [autoría propia]

2.1.6 Riesgos y plan de acción

2.1.6.1 Riesgos

- Recepción de información incorrecta por parte del aplicativo
- Falla del sistema en medio de una transacción.
- Fallo en la obtención de credenciales de la puerta de enlace *Visa* y *MasterCard*.

2.1.6.2 Plan de acción

Dadas las limitaciones y riesgos presentados durante el proyecto, se presentará el siguiente plan de acción que se tomará como medida sustitutiva para el cumplimiento y desarrollo del objetivo.

- Dado el fallo recepción de información incorrecta por parte del aplicativo, se procedería a realizar validaciones internas para mapear y seleccionar datos correctos para ser usados como parámetros en las funciones que impliquen el proceso de pago.

- Dado si existe un fallo en medio de una transacción, se procedería a crear un token de sesión con tiempo útil para prevenir cualquier acción de compra incompleta.

- Dado el fallo en la obtención de las credenciales de las puertas de enlace, se procedería a buscar y determinar los costos por transacciones que generan las principales pasarelas y procesadores de pago, tales son *Visa*, *MasterCard* y *PayPal*. En las tablas: Tabla A.2, Tabla A.3 y Tabla A.4 en la sección APÉNDICE A, se detallan los principales valores que cobran estas entidades por procesar un pago en sus servicios.

- Esto permitirá realizar una comparativa de costos por transacciones de las pasarelas de pago investigadas según el monto de transacción mensual y anual esperada por el cliente, tal y como se muestra en la Tabla

2.1. Con el fin de determinar una mejor alternativa al uso de herramientas de pago y sustituir el fallo antes mencionado.

Tabla 2.1 Precios y tarifas de los servicios de pago. [autoría propia]

	Monto de transacción	Total PayPal	Total *Braintree*	Total *MasterCard*	Total *Visa*
Mensual	$360,00	$17,16	$13,41	$5,91	$7,09
Anual	$4.320,00	$179,13	$155,58	$62,94	$82,33

2.1.7 Beneficios de la solución

- Reducción en los costos, no se pagarán comisiones por las transacciones procesadas en el aplicativo JamaSana.
- Control total sobre las transacciones realizadas en el aplicativo.
- Personalización de los métodos de pago que serán aceptados en el sistema.

2.2 Software

Las tecnologías para usar en el desarrollo del proyecto son:

Frontend:

- Lenguaje de programación *Dart*.
- *Framework Flutter*.

Backend:

- Lenguaje de programación *JavaScript*.
- *Framework Express*.

2.3 Plan de desarrollo

La implementación del proyecto consistirá en varias actividades que se describen en la Figura 2.8 Plan de actividades del proyecto. y Figura 2.9, donde se detalla la fecha de inicio y fin. En la Plan de actividades del proyecto. , se detalla el inicio del plan de actividades, donde se comenzó con el levantamiento de requerimientos.

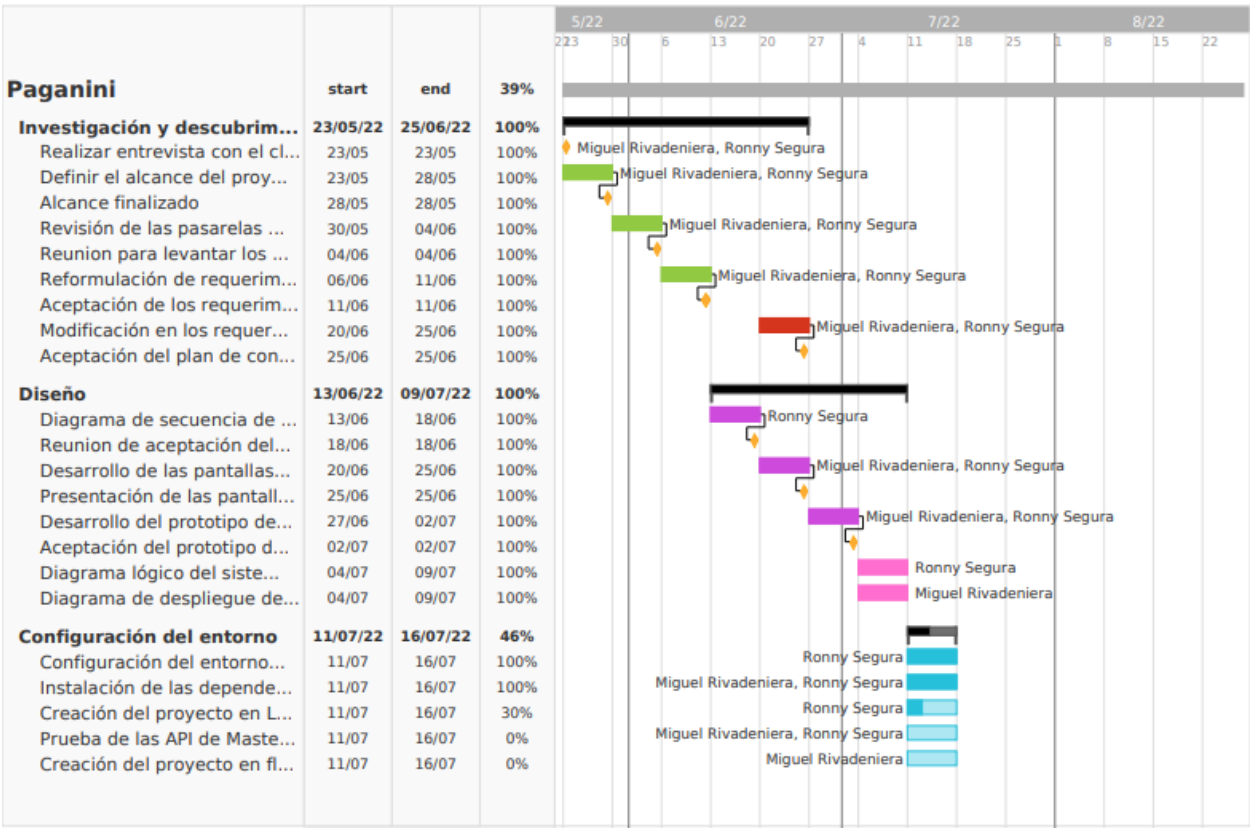


Figura 2.8 Plan de actividades del proyecto. [autoría propia]

Para el desarrollo del proyecto, se planificó la solución en 4 *sprints*, como se observa en la Figura 2.9, donde el *sprint* 1 y 3 tendrán una duración de semana y media cada uno, el *sprint* 2 durara dos semanas y el *sprint* 4 una semana de duración.

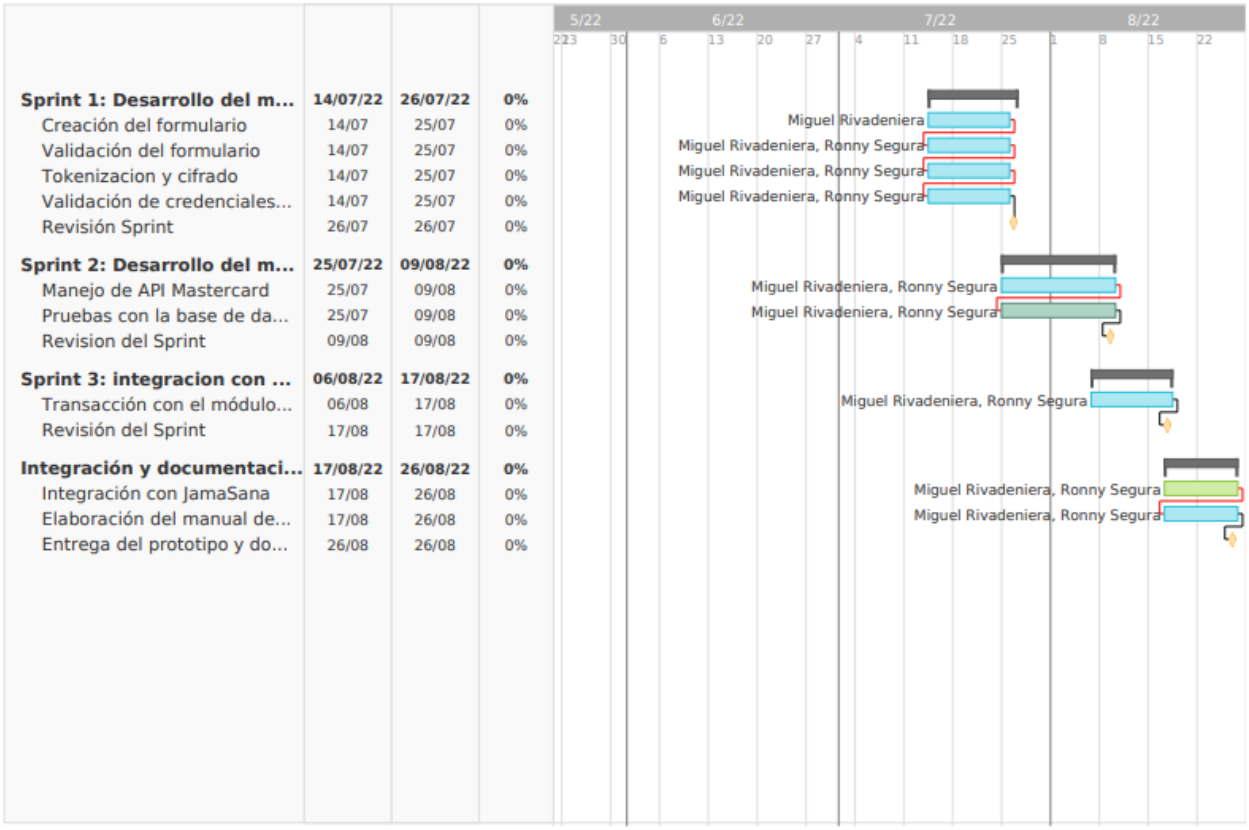


Figura 2.9 Actividades a desarrollar. [autoría propia]

CAPÍTULO 3

3. RESULTADOS Y ANALISIS

En este capítulo se presentará detalladamente el proceso de desarrollo de los módulos propuestos para el prototipo, siguiendo los requerimientos y diagramas presentados en el capítulo anterior. Se realizará un análisis para cada módulo y se discutirán los problemas encontrados a lo largo del desarrollo y sus respectivas soluciones. En el APÉNDICE D, se detalla el manual de implementación del prototipo.

3.1 Desarrollo del módulo 1: Manejo de información

Durante el desarrollo de este módulo, se realizaron varias reuniones con el cliente en las cuales se modificó el diseño final del formulario de pago y la información que el aplicativo debe entregar al módulo. En las siguientes subsecciones se detallarán estos cambios.

3.1.1 Conexión con el aplicativo

Se requirió información adicional del usuario y del aplicativo para realizar la comunicación requerida con el Módulo 2: Pasarela de pago. Esta información corresponde a las credenciales del aplicativo entregadas al momento del registro y a la siguiente información del cliente:

- Correo electrónico
- Número de teléfono
- Función de redireccionamiento
- Tema: El tema puede ser claro u oscuro
- Orden: Los ítems que el usuario desea adquirir

3.1.2 Diseño del formulario

Se retiró el campo correo electrónico debido a que esta información se obtiene directamente desde el aplicativo. Adicionalmente, se cambió el método de detección del tipo de tarjeta de una forma manual a una automática. Y, por último, se unificó el campo de la fecha de expiración. La Figura 3.1 muestra el diseño final del formulario.

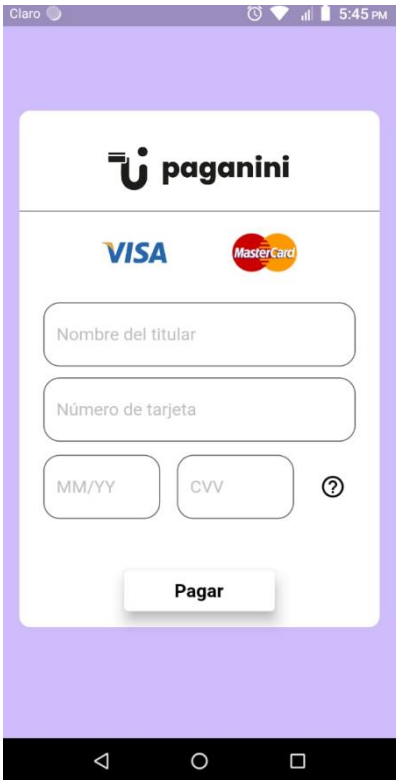


Figura 3.1 Diseño final del formulario de pago. [autoría propia]

Adicionalmente, se agregaron nuevas pantallas para presentar diferentes tipos de errores tales como:

- Número de tarjeta inválida, Figura 3.15.
- Código CVV inválido, Figura 3.16.
- Fraude detectado, Figura 3.17.
- Código de verificación incorrecto, Figura 3.19.

3.2 Desarrollo del módulo 2: Pasarela de pago

Debido a los problemas encontrados en la propuesta inicial del prototipo, se procedió con la implementación del plan de acción; es decir, para el desarrollo de este módulo se utilizaron los servicios de pago provistos por *Braintree*. A continuación, se presentan los cambios encontrados en este módulo.

3.2.1 Verificación del usuario

La verificación se logró usando la librería de *Twilio* y el número del teléfono del usuario, más no del dueño de la tarjeta. Luego de realizar la primera petición por parte del Módulo 1: Manejo de información, se procede a enviar un mensaje de texto con un código de 6 cifras generado aleatoriamente que tiene un tiempo de vida de 90 segundos. Luego, es enviado al cliente usando la API de la librería como se muestra en la Figura 3.2.

```
const verificationCode = await generateVerificationCode(req.hostname);
const message = await twilioClient.messages.create({
    body: `El código de verificación para JamaSana es: ${verificationCode}`,
    from: TWILIO_NUMBER,
    to: body.clientNumber
});
```

Figura 3.2 Envío del mensaje usando *Twilio*. [autoría propia]

3.2.2 Proceso de pago

El proceso de pago se encuentra divido en dos secciones. En primer lugar, se realiza el proceso de validación de tarjeta del cliente y luego; para terminar, se realiza la verificación del cliente y el proceso de transacción.

La primera parte del proceso se inició creando un *customer* utilizando los servicios de *Braintree*. Acto seguido, se procedió a crear un método de pago al *customer* creado, utilizando la información de la tarjeta del cliente.

Luego, se realizó la comprobación de que este proceso haya terminado con éxito, en caso de que haya terminado con errores, se procedió a identificar el error obtenido y se almacenó la transacción con el código de error correspondiente. La Figura 3.3, muestra el código de este proceso.

```
// NUEVO CUSTOMER VACIO
newCustomer = await gateway.customer.create({});

// ERRORES: TARJETA INVALIDA, CVV INVALIDO, FRAUDE DETECTADO
// Crear el metodo de pago para el customer
const newCard = await gateway.creditCard.create({
    customerId: newCustomer.customer.id,
    ...JSON.parse(decrypt(body.cardData.data)),
    options: {
        failOnDuplicatePaymentMethod: true,
        makeDefault: true,
        verifyCard: true
    }
});
```

Figura 3.3 Creación del *customer* con método de pago. [autoría propia]

En la segunda parte del proceso se procedió a comprobar que el código enviado por el usuario sea el correcto. Si los códigos no coinciden, se actualiza la transacción creada en el primer proceso con el código de error correspondiente. Si el código es correcto, se crea la transacción tipo *sale*, usando los servicios de *Braintree* el cual culmina el proceso de pago. En la Figura 3.4 se muestra el código de este proceso.

```
const code = decrypt(body.verificationCode);
const codeFromRedis = await getVerificationCode(req.hostname);
if (code != codeFromRedis) {
    const response = await getCode('0050');
    await updateTransactionStatusByID(transaction.transactionID, '0050');
    await gateway.customer.delete(body.paymentID);
    console.error("Error en la verificacion del codigo enviado al customer");
    return res.status(400).json(response);
}

// Crear la nueva transaccion en braintree
const newSale = await gateway.transaction.sale({
    customerId: body.paymentID,
    amount: transaction.amount,
});
```

Figura 3.4 Verificación del código y transacción. [autoría propia]

3.3 Plan de pruebas

Para comprobar los resultados esperados del sistema se elaboró un plan de pruebas el cual se muestra completamente en el APÉNDICE E, Tabla E.1. El plan de pruebas cuenta con las siguientes categorías:

- Acceso al sistema
- Validación
- Verificación
- Transacción

3.4 Análisis y resultados de las pruebas de integración con el aplicativo

En esta sección se presentarán los resultados del plan de pruebas, incluyendo una descripción general de cada proceso.

3.4.1 Configuración del entorno para pruebas

Para empezar a realizar las pruebas, se necesitaron archivos de configuración para ambos módulos. En primer lugar, se creó el par de llaves privada y pública para el uso del cifrado y descifrado asimétrico utilizando la herramienta *OpenSSL* como se muestra en la Figura 3.5.

```
# Creación de llave privada
openssl genrsa -out jamasana-private-key.pem 4096

# Creación de la llave pública
openssl rsa -in jamasana-private-key.pem -outform PEM -pubout -out jamsana-public-key.pem
```

Figura 3.5 Creación de llaves. [autoría propia]

En el aplicativo se creó un archivo *.env* el cual contiene las credenciales necesarias para realizar la conexión con el Módulo 2: Pasarela de pago. Un ejemplo de este archivo se muestra en la Figura 3.6.

```
APP_ID="" # ID del aplicativo entregado al momento del registro

APP_SECRET="" # Secreto del aplicativo entregado al momento del registro
```

Figura 3.6 Credenciales del aplicativo. [autoría propia]

Luego, se procedió a crear un archivo llamado *.env* en la ruta raíz del Módulo 2: Pasarela de pago, ingresando la información para realizar la conexión con los servicios externos utilizados. Un ejemplo de este archivo se muestra en la Figura 3.7.

```
URI='' # Url de la base de datos de mongodb
PORT=3000
SECRET='' # Cadena de caracteres secreta para realizar la firma de los tokens de acceso
EXPIRE_TIME=900 # Tiempo de vida del token de acceso expresado en segundos.
PRIV_KEY='Ruta a la llave privada en formato pem'

BT_ENVIRONMENT='Sandbox' # Entorno de pruebas
BT_MERCHANT_ID='Recuperar de la cuenta de Braintree'
BT_PUBLIC_KEY='Recuperar de la cuenta de Braintree'
BT_PRIVATE_KEY='Recuperar de la cuenta de Braintree'

ACCOUNT_SID_TWILIO='Recuperar de la cuenta de Twilio'
AUTH_TOKEN_TWILIO='Recuperar de la cuenta de Twilio'
NUMBER_TWILIO='Recuperar de la cuenta de Twilio'

MSG_EXP=90 # Tiempo de vida del código de verificación expresado en segundos
```

Figura 3.7 Credenciales para la conexión con el servicio. [autoría propia]

3.4.2 Acceso al sistema

Para esta prueba se realizó una petición al Módulo 2: Pasarela de pago, para obtener acceso al formulario de pago. La petición fue de tipo *POST* y contenía las credenciales del aplicativo en la cabecera *autorization* como se muestra en la Figura 3.8.

```
- PETICIÓN TIPO: POST

- Hedader: { Authorization:
"Y2VjMGE0YzQ1NzE5MzBhOWM4ZTkxOTZkNDZlNTNlYWQ6ZWJiN2RiMGRhODgzND
kzYjhjMmUxNjY2ZWNlOTAzZWJiOGRlYzUwOTQ5NTI5ZGI5Y2U2OGIxYzJkNjlmY
2VmZg==" }

- Body: { grant_type: "client_credentials" }
```

Figura 3.8 Petición correcta para solicitar acceso. [autoría propia]

Como resultado de esta petición, se obtuvo la respuesta del servidor mostrada en la Figura 3.9 que indica que el aplicativo obtuvo acceso al sistema, y por parte del Módulo 1: Manejo de información, se muestra el formulario de pago mostrado en la Figura 3.9 y Figura 3.10.

```
{
  "access_token":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6ImNlYzBhNGM0NTc
xOTMwYTljOGU5MTk2ZDQ2ZTUzZWFkIiwic2VjcmV0IjoiZWJiN2RiMGRhODgz
NDkzYjhjMmUxNjY2ZWNlOTAzZWJiOGRlYzUwOTQ5NTI5ZGI5Y2U2OGIxYzJkN
jlmY2VmZiIsImlhdCI6MTY2MTQ1Njg2OCwiZXhwIjoxNjYxNDU3NzY4fQ.nmJ
rHz25HNccWCEvEgbU2y8G0k9eGdOOEgG3w7e2N0U",
  "token_type": "Bearer",
  "expires_in": 900
}
```


Figura 3.9 Respuesta del *backend*. [autoría propia]

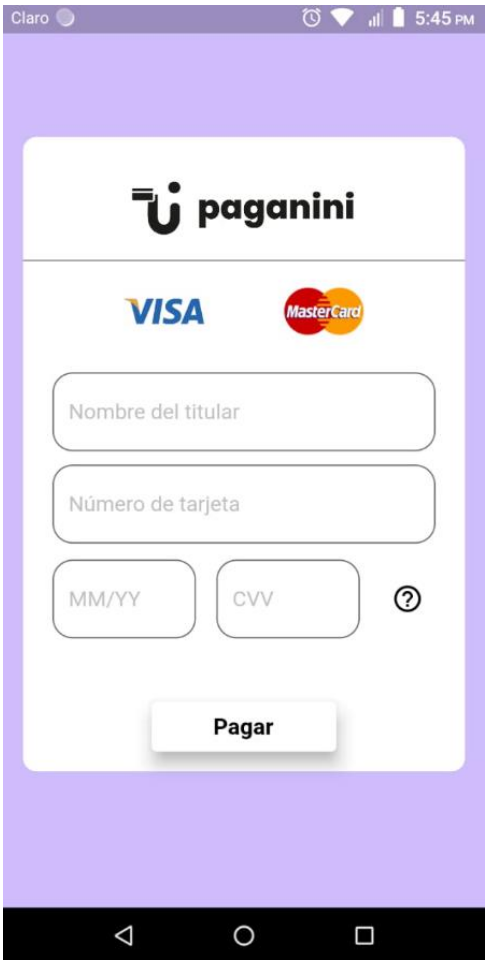


Figura 3.10 Formulario de pago. [autoría propia]

Se realizó una petición con errores en las credenciales del aplicativo, con la cual se obtuvo como respuesta el objeto mostrado en la Figura 3.11 que indica que el

proceso de autenticación y autorización falló. En el Módulo 1: Manejo de información, se presenta la pantalla que se muestra en la Figura 3.12. Este mismo error se obtuvo enviando una petición con error de sintaxis.

```
{
  "error": true,
  "reason": "AUTHORIZATION_FAILED",
  "description": "Error en la autenticación por falta de
cabecera de autorización o credenciales inválidas"
}
```

Figura 3.11 Mensaje de error. [autoría propia]

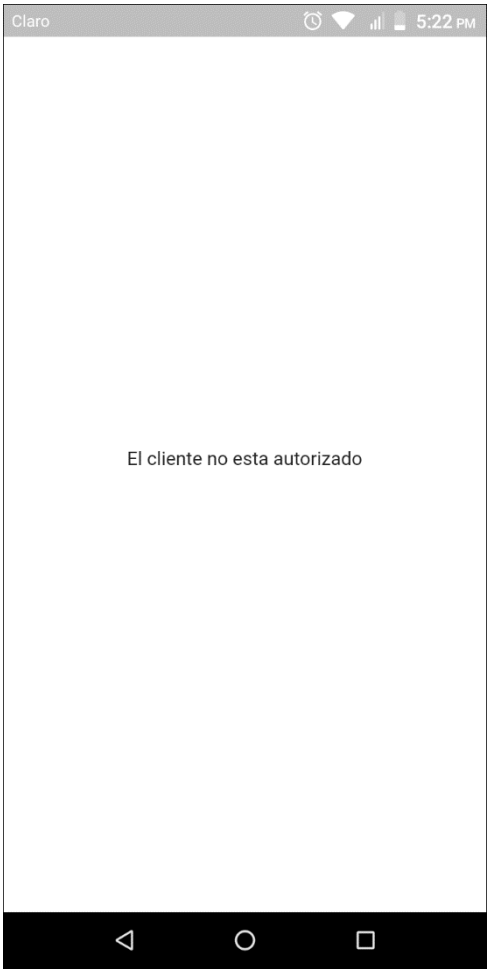


Figura 3.12 Pantalla de Cliente no autorizado. [autoría propia]

3.4.3 Validación

Los datos utilizados para esta prueba se encuentran descritos en el APÉNDICE E – Datos de prueba.

En primer lugar, se realizó la prueba utilizando los datos válidos. Como resultado, se obtuvo la respuesta mostrada en la Figura 3.13 por parte del servidor que corresponde al ID de la transacción procesada y al ID del método de pago, la cual sirve para la confirmación de la transacción luego de la verificación.

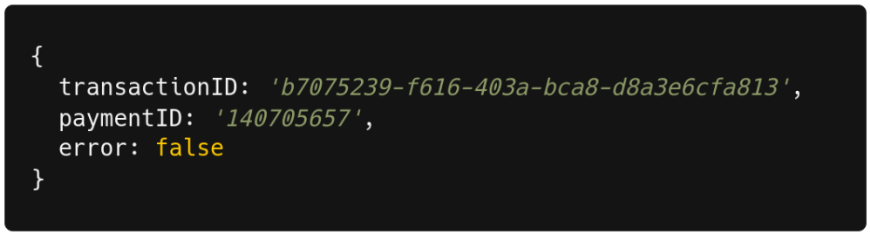


Figura 3.13 Confirmación exitosa. [autoría propia]

Por otro lado, en el Módulo 1: Manejo de información, se presentó la pantalla que corresponde a la verificación del usuario tal como se muestra en la Figura 3.14, indicando que el proceso de verificación culminó de forma correcta.

Figura 3.14 Pantalla código de verificación. [autoría propia]

Para verificar los resultados de validación de datos inválidos de las tarjetas se realizaron pruebas correspondientes a los siguientes escenarios: número de tarjeta inválida, código CVV inválido y fraude detectado.

Número de tarjeta inválido

Como resultado de este escenario, se obtuvo el mensaje mostrado en la Figura 3.15. Indicando que el número de tarjeta ingresado fue inválido.

Figura 3.15 Mensaje tarjeta invalida. [autoría propia]

Código CVV inválido

Como resultado de este escenario, se obtuvo el mensaje mostrado en la Figura 3.16, confirmando de esta forma que la validación del código de seguridad funciona.

Figura 3.16 Mensaje de error por código CVV invalido. [autoría propia]

Fraude detectado

Por último, como resultado de un proceso de detección de fraude al momento de crear la transacción, se obtuvo le mensaje mostrado en la Figura 3.17.

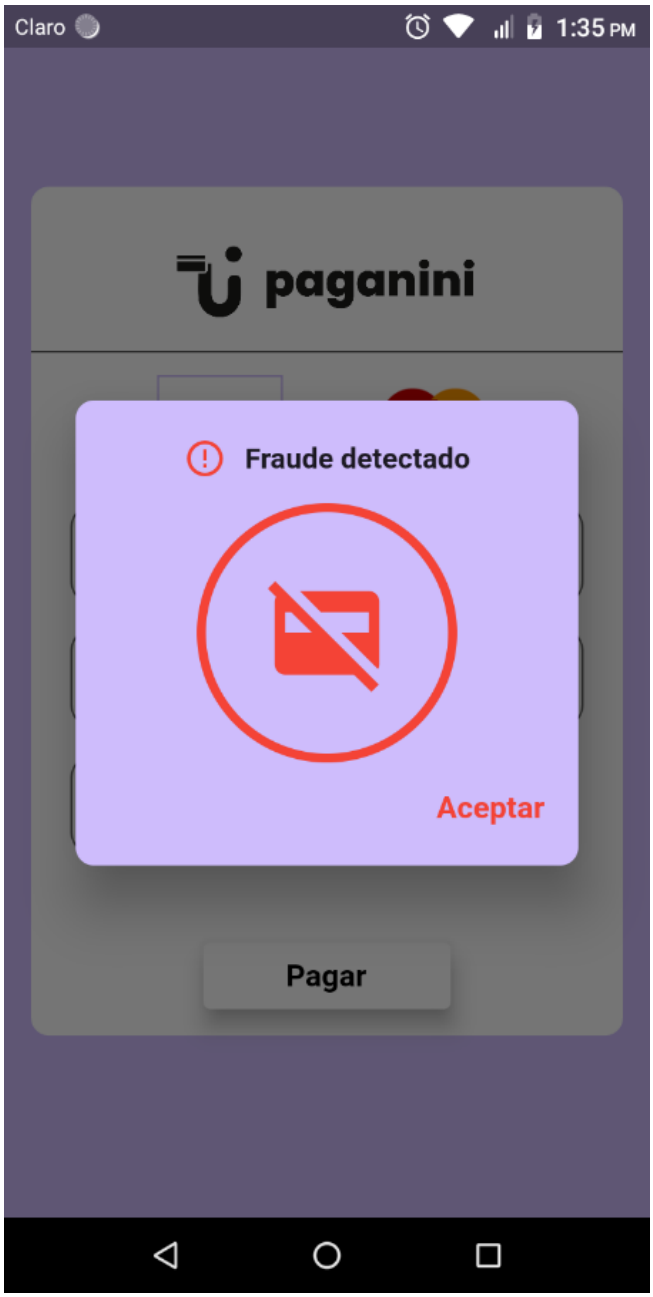


Figura 3.17 Mensaje tarjeta fraudulenta. [autoría propia]

Las pruebas realizadas muestran el proceso de validación funcionando correctamente utilizando los datos de prueba provistos por *Braintree*, gracias a esto se puede asegurar que los datos que el cliente siempre serán validados, evitando los posibles errores en el proceso del pago.

3.4.4 Verificación

Para realizar la prueba se usa como punto de inicio la pantalla mostrada en Figura 3.14. Luego de que se mostró esta pantalla, se recibió un código de verificación por SMS, dicho código se muestra en la Figura 3.18.

Figura 3.18 Código enviado al cliente. [autoría propia]

Cuando se ingresó el código de verificación incorrecto, como resultado se obtuvo el mensaje que se muestra en la Figura 3.19. Este mismo mensaje se obtuvo cuando el código de verificación correcto es enviado tarde.

Figura 3.19 Mensaje de Código invalido. [autoría propia]

Cuando se ingresó el código de verificación correcto, y dentro del tiempo aceptado, se obtuvo como resultado el mensaje mostrado en la Figura 3.20. Culminando el proceso de pago completo.

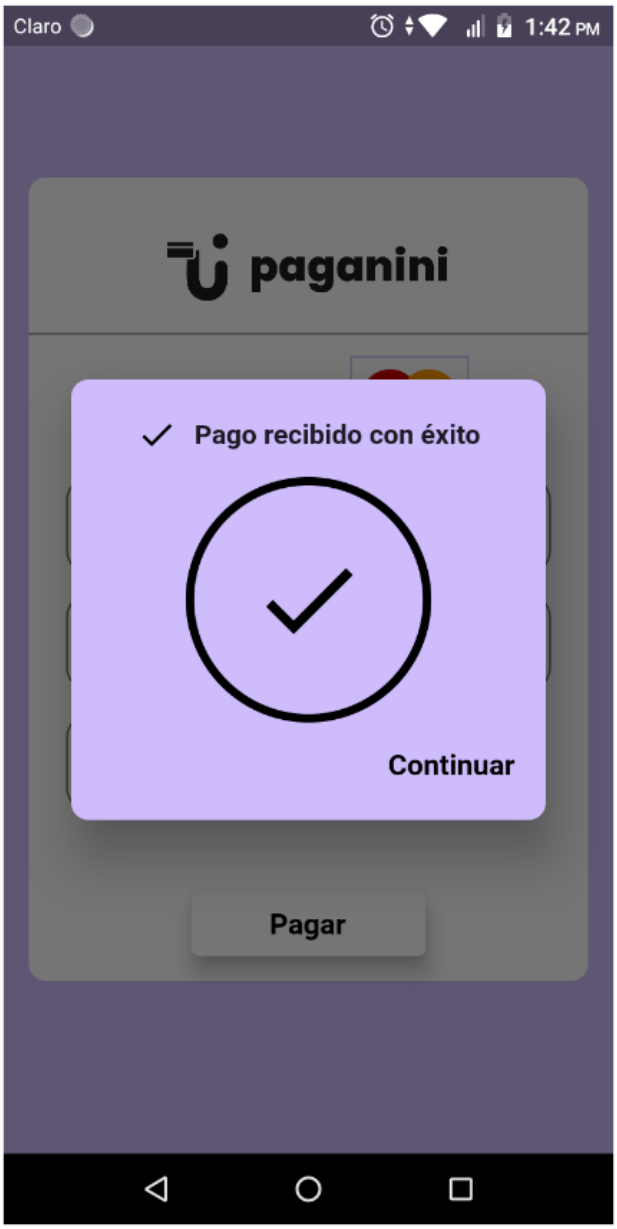


Figura 3.20 Transacción exitosa. [autoría propia]

3.4.5 Transacción

Para comprobar los resultados de esta prueba, se presentarán las transacciones creadas y actualizadas en las diferentes etapas del proceso de pago con su correspondiente código de estado. Estos códigos corresponden a los errores mencionados en las pruebas anteriores. Debido a que el resultado a presentar

consiste únicamente en el código con el cual se almacena una transacción, se utilizará como referencia la Figura 3.21.

Figura 3.21 Objeto de transacción. [autoría propia]

Transacción en estado de espera

Luego de completar el primer proceso de pago, que corresponde a la validación, la transacción que se creó en el sistema mostró el código “0090”, lo cual indica que la transacción se encuentra en un estado de espera a ser completada.

Transacción con errores

Durante la ejecución de las pruebas se pudo comprobar que las transacciones se actualizaron con los códigos correspondiente a los errores presentados anteriormente. Los resultados son los siguientes:

- Error de validación de tarjeta: La transacción se actualizó con el código “0020”.
- Error de código CVV: La transacción se actualizó con el código “0024”.
- Error de fraude detectado: La transacción se actualizó con el código “0021”.
- Error de código de verificación: La transacción se actualizó con el código “0050”.

Transacción aprobada

Por último, cuando el proceso de pago se completó sin errores, se actualizó la transacción con el código “0000” el cual representa un estado de aprobación de la transacción.

3.5 Análisis de costos

Para el desarrollo de este proyecto se han requerido varios servicios, los cuales se detallan en la Tabla 3.1. Los precios detallados corresponden a valores obtenidos en los sitios oficiales de los servicios mencionados al momento del desarrollo del proyecto.

La columna precio mínimo hace mención ya sea a un plan básico o mediante suscripción gratis, esto permite acceder a credenciales de prueba y servicios limitados; mientras que la columna precio máximo cubre todos los beneficios tanto en *hardware* como en *software*.

Tabla 3.1 Costos de desarrollo. [autoría propia]

Servicio	Tipo	Precio mínimo mensual	Precio máximo mensual
Digital Ocean Droplet	Servidor en la nube.	$6.00	$84.00
MongoDB	Servicio de base de datos.	$0.00	$57.00
Dominio web	Dirección web	$4.99 (primer año)	$19.99
Braintree	Servicio de pasarela de pago.	$0.00	$13.41

Twilio	Servicio de envío de SMS	$0.0075 (por envió de mensaje)	$0.1253
Mano de obra	Servicio por desarrollo de *software*	$600.00	$1200.00
Total		**$611.00**	**$1374.53**

Para el desarrollo de este proyecto, se utilizaron los precios mínimos ofrecidos por estos servicios.

Como costo final del desarrollo, se tiene un precio de $11. Debido a que el prototipo fue desarrollado como proyecto de materia integradora, lo que implica que no se tomaron en cuenta los costos por el mano de obra. Sin embargo, al momento de poner el proyecto en producción, se deberá tomar en cuenta la columna de los precios máximos de la Tabla 3.1, con un costo adicional por mano de obra en caso de requerirse equipo de desarrollo con experiencia.

3.6 Cierre de proyecto

Para culminar el proyecto, se realizó una reunión con el cliente en la cual se presentó y se ejecutó el plan de pruebas. Para comprobar la satisfacción del cliente con el proyecto, se elaboró un acta de aceptación la cual se presenta en el APÉNDICE F, la cual el cliente firmó. Los entregables de este proyecto fueron:

- Manuales de uso e instalación
- Diagramas del proyecto
- Código fuente del sistema

CAPÍTULO 4

4. CONCLUSIONES Y RECOMENDACIONES

En este capítulo se detallan las conclusiones que busca responder los objetivos que se describieron en el CAPÍTULO 1. Además, de dar ciertas recomendaciones para futuros trabajos y mejoras en el servicio de pago.

4.1 Conclusiones

- Se obtuvo un prototipo funcional que cumplió con las expectativas y requerimientos solicitados por el cliente. Además, se logró diseñar una interfaz gráfica intuitiva y sencilla de utilizar, cumpliendo con el principio KISS para disminuir la complejidad y simplicidad del flujo de iteraciones del proceso de pago, ofreciendo métodos de ayuda para el usuario tales como: validación de la información ingresada en los campos del formulario y la presentación de mensajes de error descriptivos.

- Con la implementación de mecanismos de seguridad en el sistema tales como: el cifrado de datos sensibles del usuario, el uso del protocolo HTTPS para el envío de la información de forma segura por medio de internet y la verificación del usuario antes de realizar una transacción, se logró una comunicación y protección segura de información, para mantener la integridad y cifrado de los datos al ser transmitida. Además, se logró generar una gran confiabilidad hacia el cliente y el usuario gracias a la seguridad ofrecida en el sistema ya que, durante las pruebas, la información del usuario no se vio comprometida.

- Se logró registrar la información requerida de la transacción en la base de datos del sistema, lo cual le facilita al cliente realizar seguimiento de las transacciones realizadas en el aplicativo e identificar los diferentes estados de la transacción en los procesos de pago, además de identificar al usuario del aplicativo quien realiza la transacción y los servicios que este adquirió.

- El uso de los servicios de *Braintree* facilitó procesos importantes dentro de una pasarela de pago, tal como lo es la validación de la información de la tarjeta del usuario. Gracias a este servicio, se logró identificar los errores más comunes que pueden presentar las tarjetas de los usuarios, por ejemplo: número inválido, código CVV inválido o incluso la detección de fraude. Adicionalmente, sirve como almacén temporal de métodos de pago, por lo cual la información sensible del usuario solo se utiliza una vez dentro del proceso de pago, ofreciendo mayor seguridad al momento de completar la transacción.

- Al implementar una metodología ágil basada en *scrum*, facilito el desarrollo del proyecto al promover la interacción constante con el cliente, permitiendo la detención de requerimientos y errores durante el desarrollo para cumplir con los objetivos específicos del sistema de pagos.

4.2 Recomendaciones

- Se recomienda que, para mejorar las necesidades del cliente en la reducción de costos, se busque una alternativa como usar las puertas de enlace *Visa* o *MasterCard*, debido a los bajos costos por transacciones que generan estas puertas de enlace. Del mismo modo, se sugiere contactarse directamente con estas banderas de tarjetas para obtener las credenciales y códigos de prueba para una mejor adaptación al proyecto.

- Debido a que *Visa* y *MasterCard* tienen una comunicación directa con los bancos emisores de tarjetas de crédito o débito, se recomienda utilizar los servicios de verificación de usuario que ofrecen estas redes de tarjetas. De este modo, se ofrece mayor seguridad y confiabilidad al momento de completar una transacción ya que se está realizando la verificación de identidad del dueño de la tarjeta utilizada para dicha transacción.

- Se recomienda realizar las evaluaciones requeridas por el PCI DSS relacionadas a la infraestructura a utilizar para la comercialización y alojamiento del sistema, para que el sistema pueda ser puesto en producción.

- Debido al rápido crecimiento de los comercios en línea, se recomienda integrar una mayor variedad de métodos de pago, los cuales podrían ser: mayor soporte a emisores de tarjetas de crédito o débito, transferencias bancarias, monedas virtuales o carteras electrónicas. De esta manera se contará con un sistema más robusto y flexible a las posibilidades del usuario que utilice el sistema.

- Se recomienda realizar un análisis de riesgos cuando se depende de servicios externos en la implementación de un proyecto, de esta forma se tiene en cuenta la mayoría de los posibles inconvenientes que puedan ocurrir al momento del desarrollo, permitiendo crear un plan de acción para cada riesgo analizado. De esta forma se logrará evitar un posible escenario en el cual el proyecto no pueda ser completado, completado parcialmente o que presente retrasos extensos a mitad del desarrollo.

BIBLIOGRAFÍA

[1] R. Y. Sumba Bustamante, S. M. Almendariz Gonzalez, C. L. Baque Chacay y V. G. Aliatis Bravo, «Emprendimientos en tiempo de covid-19: De lo tradicional al comercio electrónico,» *FIPCAEC,* vol. 5, nº 4, pp. 137-164, 2020.

[2] Asociación de Bancos del Ecuador, «ASOBANCA,» 19 8 2021. [En línea]. Available: https://asobanca.org.ec/wp-content/uploads/2021/08/Boletín-Económico-agosto-2021.pdf. [Último acceso: 5 6 2022].

[3] VTEX, «Tutoriales y soluciones,» [En línea]. Available: https://help.vtex.com/es/tutorial/diferenca-entre-adquirentes-subadquirentes-e-gateways-no-brasil?locale=pt. [Último acceso: 9 6 2022].

[4] Braintree, «Braintree a PayPal service,» [En línea]. Available: https://www.braintreepayments.com/faq. [Último acceso: 17 08 2022].

[5] MasterCard, «MasterCard,» [En línea]. Available: https://developer.mastercard.com/product/payment-gateway-services-mpgs/. [Último acceso: 17 08 2022].

[6] PayPal, «About,» [En línea]. Available: https://about.pypl.com/who-we-are/default.aspx. [Último acceso: 17 08 2022].

[7] Visa, «Visa Developer Center,» [En línea]. Available: https://developer.visa.com/developer_program. [Último acceso: 17 08 22].

[8] PAYCOMET, «Pasarela de pagos online,» [En línea]. Available: https://www.paycomet.com/news/glosario/que-es-pasarela-de-pagos-online/. [Último acceso: 9 6 2022].

[9] R. Avani, «Security of online transactions,» *Asian Journal of Research in Business Economics and Management,* vol. 7, nº 8, p. 230, 2017.

[10] Pasarela de pago, «La seguridad en una pasarela de pago,» [En línea]. Available: https://pasarelapago.com/la-seguridad-en-una-pasarela-de-pago/. [Último acceso: 9 6 2022].

[11] Visa Developer Center, «Visa Token Service,» 2015-1022. [En línea]. Available: https://developer.visa.com/capabilities/vts. [Último acceso: 9 6 2022].

[12] ComputerWeekly.es, «¿Qué es Tokenización? - Definición en WhatIs.com,» 12 8 2021. [En línea]. Available: https://www.computerweekly.com/es/definicion/Tokenizacion#:~:text=La%20tokenizaci%C3%B3n%20es%20el%20proceso,datos%20sin%20comprometer%20su%20seguridad.. [Último acceso: 16 6 2022].

[13] OSI - Oficina de Seguridad del Internauta, «Métodos de pago y su seguridad,» [En línea]. Available: https://www.osi.es/es/actualidad/blog/2022/04/20/metodos-de-pago-y-su-seguridad. [Último acceso: 16 6 2022].

[14] Banco Santander, «Diferencia entre tarjeta de crédito y débito,» [En línea]. Available: https://www.bancosantander.es/faqs/particulares/tarjetas/diferencias-tarjeta-credito-debito#:~:text=En%20una%20tarjeta%20de%20d%C3%A9bito,cobro%20hasta%20el%20mes%20siguiente.. [Último acceso: 16 6 2022].

APÉNDICES

APÉNDICE A

TABLA DE CÓDIGOS

Tabla A.1 Códigos de éxito o error del sistema. [autoría propia]

Código	Razón	Descripción
0000	*APPROVED*	Pago recibido con éxito
0010	*AUTHORIZATION_FAILED*	Error en la autenticación por falta de cabecera de autorización o credenciales inválidas
0011	*AUTHORIZATION_FAILED*	Token inválido
0020	*DECLINED*	Tarjeta inválida
0021	*DECLINED*	Fraude detectado
0022	*DECLINED*	Error en el procesador de pagos
0023	*DECLINED*	Fondos insuficientes
0024	*DECLINED*	Código CVV incorrecto
0025	*DECLINED*	Error en transacción
0026	*DECLINED*	Error en la verificación de la tarjeta
0060	*INVALID_CLIENT*	No se encuentran las credenciales del cliente
0061	*INVALID_CLIENT*	Fallo en la autenticación del cliente
0070	*INVALID_GRANT_TYPE*	Tipo de acceso no permitido
0080	*INVALID_DATA*	Datos incorrectos
0090	*ON_HOLD*	Transacción en espera para liquidación
0050	*VERIFICATION_CODE_ERROR*	Código de verificación incorrecto
0040	*TRANSACTION_ERROR*	Transacción cancelada

Los códigos detallados en la Tabla A.1 son los presentados en cada respuesta de transacción y conexión entre el aplicativo y el sistema.

TABLAS DE COSTOS

Tabla A.2 Costos por transacción visa. [autoría propia]

Visa		
% por transacción	**Tarifa fija**	**Descripción**
1,25%	$0,10	Precio base de la transacción

0,65%	$0,15	Precio por transacción en línea
1,90%	**$0,25**	**Total**

Tabla A.3 Costos por transacción *MasterCard*. [autoría propia]

MasterCard		
% por transacción	**Tarifa fija**	**Descripción**
1,25%	$0,10	Precio base de la transacción
0,19%	$0,63	Precio por transacción en línea
1,44%	**$0,73**	**Total**

Tabla A.4 Costos por transacción PayPal. [autoría propia]

PayPal		
% por transacción	**Tarifa fija**	**Descripción**
2,59%	$0,49	Precio base de la transacción
1,50%	$0,00	Por ventas internacionales
0,00%	$1,95	Vinculación y verificación de tarjeta
4,09%	**$2,44**	**Total**

Tabla A.5 Costos por transacción *Braintree*. [autoría propia]

Braintree		
% por transacción	**Tarifa fija**	**Descripción**
2,59%	$0,49	Precio base de la transacción
1,00%	$0,00	Precio adicional por transacciones fuera de USA.
3,59%	**$0,49**	**Total**

TABLA DE SERVICIOS JAMASANA

Tabla A.6 Precio por servicio. [autoría propia]

Servicio	**Valor**
1 comidas	$6,00
3 comidas	$18,00
*1 persona mes	$360,00

*1 persona año	$4.320,00

TABLA DE SUBSCRIPTORES JAMASANA

Tabla A.7 Cantidad de subscriptores esperados. [autoría propia]

	No. clientes	Suscripciones
Mensual	50	50
Anual	600	600

Tabla A.8 Total de transacción esperada por usuario. [autoría propia]

	Valor por suscripción
Mensual	$360,00
Anual	$4.320,00

APÉNDICE B

PROTOTIPO PAGANINI

Para generar un pago, el usuario deberá insertar los datos en los campos requeridos, proporcionando el nombre del titular de la tarjeta y los detalles de la tarjeta con la que desea pagar, luego dar clic en el botón pagar. Esta pantalla también proveerá la opción para que el usuario selecciones la red de tarjeta con la que desea pagar como se muestra en la Figura B.1.

Figura B.1 Pantalla de pago modo oscuro. [autoría propia]

El sistema contara con mensajes de ayuda como se muestra en la Figura B.2, para guiar al usuario e indicar donde se encuentra el código CVV solicitado por el formulario.

Figura B.2 Guía código CVV. [autoría propia]

Cuando el usuario ingrese un dato o información incorrecta a la solicitada, el sistema generar una advertencia que indique que la tarjeta ha sido rechazada, tal como se muestra en la Figura B.3.

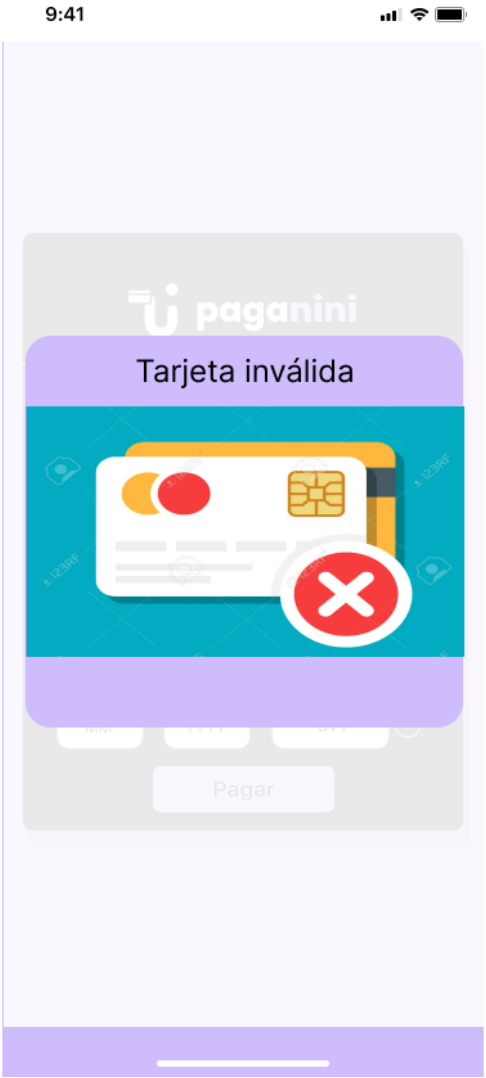


Figura B.3 Pantalla de advertencia, tarjeta rechazada. [autoría propia]

El sistema procederá a verificar los datos de la tarjeta y detectará fraude, fecha expirada, código inválido, si se detecta alguno de estos problemas, el sistema responderá con una pantalla de tarjeta invalida, como se muestra en la Figura B.4.

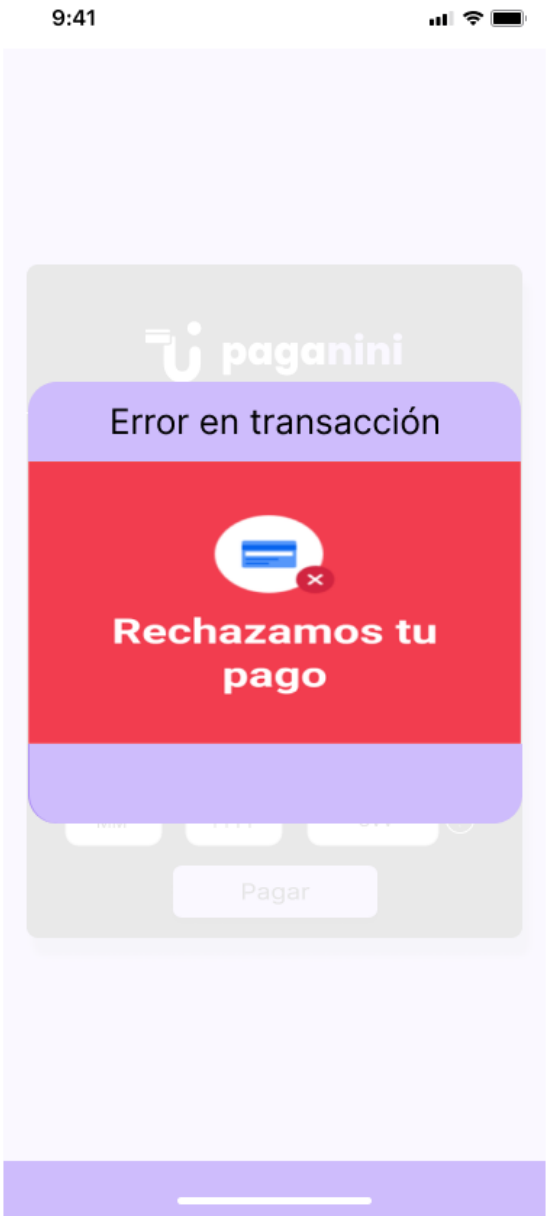


Figura B.4 Pantalla de transacción cancelada. [autoría propia]

Si los datos de la tarjeta son correctos y no detecta ni una novedad durante la verificación de la tarjeta, el sistema retornara una pantalla de confirmación de pago, como se muestra en la Figura B.5.

Figura B.5 Pantalla de orden confirmada. [autoría propia]

APÉNDICE C

CARTA DE ACEPTACIÓN DEL PROTOTIPO DE BAJO NIVEL

Guayaquil, 15 de junio del 2022

Estimado Ph.D Boris Vintimilla,

Como cliente del proyecto "Desarrollo de un prototipo de pasarela de pago para la aceptación de tarjeta de crédito y débito", el cual está siendo desarrollado por los estudiantes: Ronny Hugo Segura Merchán y Miguel Enrique Rivadeneira Segovia dentro de la Materia Integradora por usted dictada, certifico que he revisado y aprobado el prototipo de dicho proyecto.

Ing. Leonardo Castro

Cliente del Proyecto

Figura C.1 Carta de aceptación del prototipo. [autoría propia]

Características del prototipo

El formulario tiene los siguientes campos:

- Número de tarjeta
- Nombre del titular
- Correo electrónico
- Fecha de expiración: MM/YY
- Código CVV
- Botón para pagar

Figura C.2 Características del prototipo. [autoría propia]

Prototipo – Modo claro

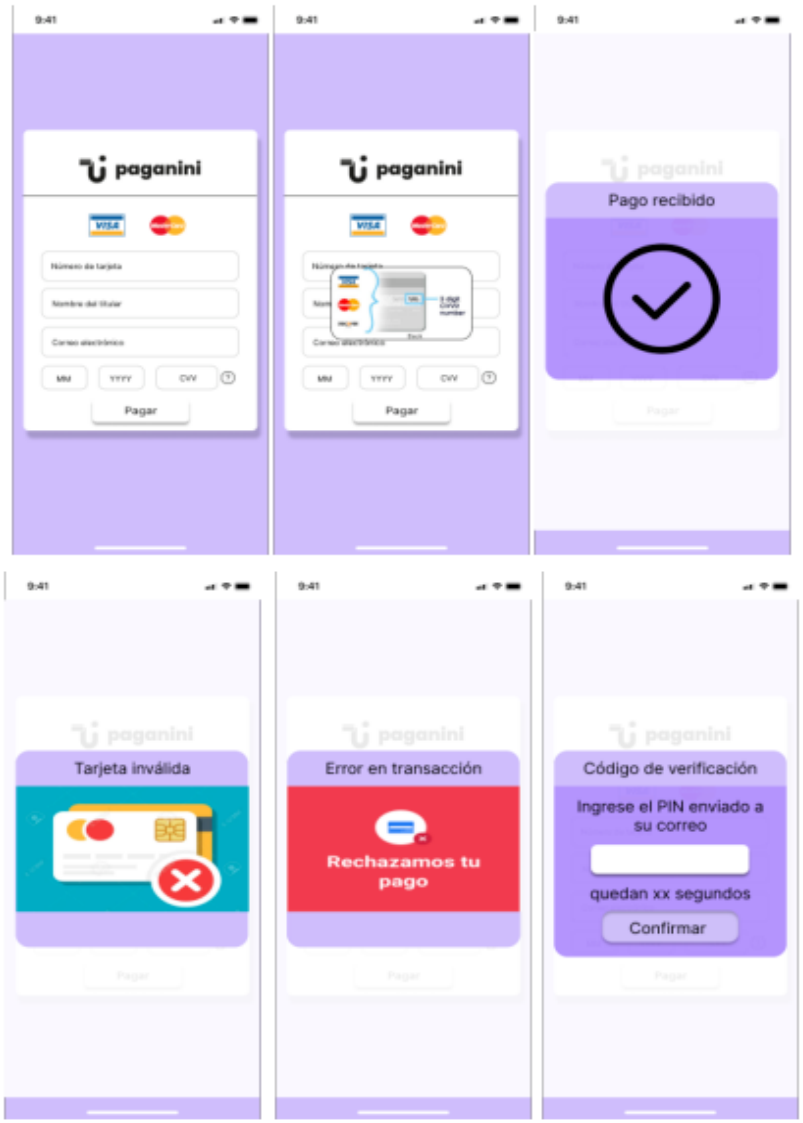


Figura C.3 Pantallas modo claro del prototipo. [autoría propia]

Prototipo – Modo oscuro

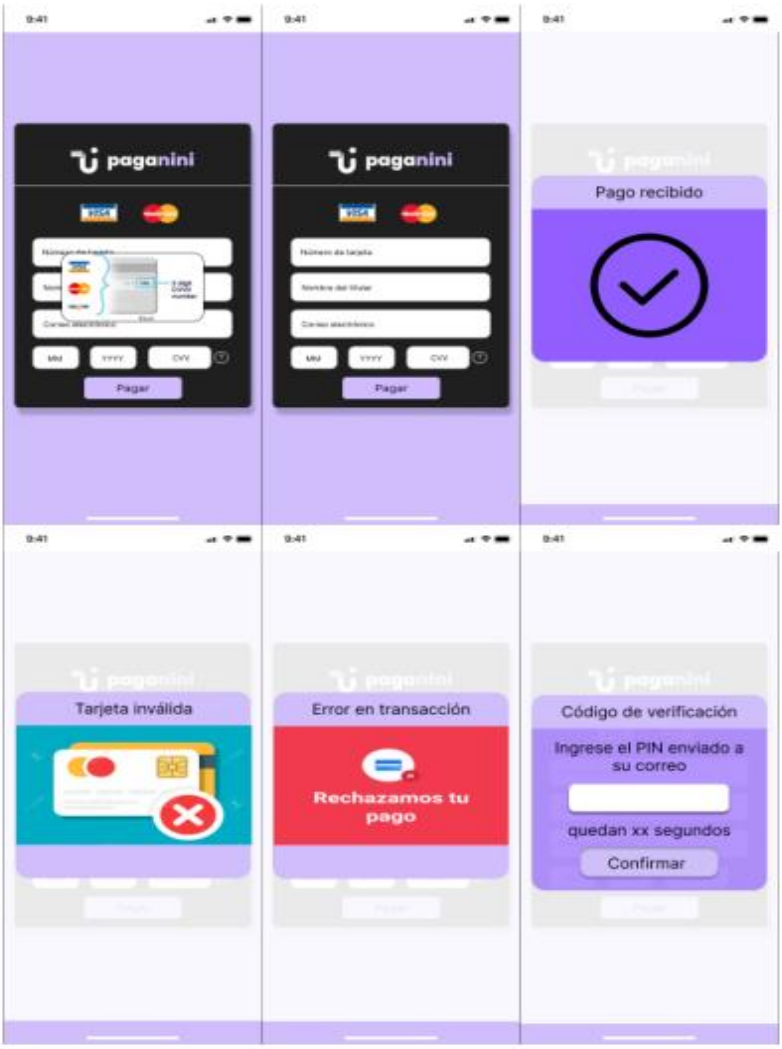


Figura C.4 Pantallas modo oscuro del prototipo. [autoría propia]

APÉNDICE D

MANUAL DE IMPLEMENTACIÓN LOCAL

El sistema PAGANINI se encuentra desarrollado en una estructura cliente - servidor que permite ser ejecutado en un mismo o diferente sistema. En esta guía se detallará la información pertinente para ejecutar el sistema PAGANINI en un entorno local.

INSTALACIÓN Y EJECUCIÓN DEL BACKEND

1. Tener instalado *Node.js*[3], para ello debe acceder a la página oficial como se muestra en la Figura D.1 y descargar e instalar el paquete adecuado al sistema operativo que tiene instalado.

Figura D.1 Página oficial para obtener Node.js. [autoría propia]

[3] https://nodejs.org/es/

2. Una vez instalado *node.js*, mediante consola de su preferencia asegúrese de tener instalado *express.js*[4]. En este caso se hará uso del *PowerShell* de *Windows* tal como se muestra en la Figura D.1

Figura D.1 Terminal de *Windows*. [autoría propia]

3. Si en el caso que no tenga instalado *express.js*, mediante consola use el comando que se muestra en la Figura D.2.

Figura D.2 Instalación de express.js. [autoría propia]

4. Una vez que tenga instalada la estructura web, proceda a clonar el repositorio paganiniBackend tal y como se muestra en la Figura D.3, use el siguiente comando:

git clone https://github.com/rohusemer/paganiniBackend.git

Figura D.3 Comando para clonar el repositorio *backend*. [autoría propia]

5. Diríjase a la carpeta donde clonó el repositorio y acceda al directorio paganiniBackend tal como se muestra en la Figura D.4 con el siguiente comando:

cd paganiniBackend

[4] http://expressjs.com/es/starter/installing.html

Figura D.4 Acceda al directorio del *Backend*. [autoría propia]

6. Proceda a instalar todas las dependencias que necesita el proyecto, tal como se muestra en la Figura D.5, use el siguiente comando:

   npm install o npm i

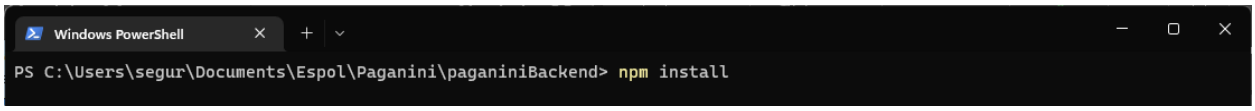


Figura D.5 Instalación de las dependencias del proyecto. [autoría propia]

7. Finalmente ejecute el servidor, como se muestra en la Figura D.6, use el comando:

   npm run dev

Figura D.6 Ejecución del servidor en entorno local. [autoría propia]

Asegúrese de tener la conexión con los servicios de *Braintree* y la base de datos de *MongoDB*. Ejecute el editor de su preferencia en este caso se usará *Visual Studio Code*, abra la carpeta del repositorio clonado, tal como se muestra en la Figura D.7.

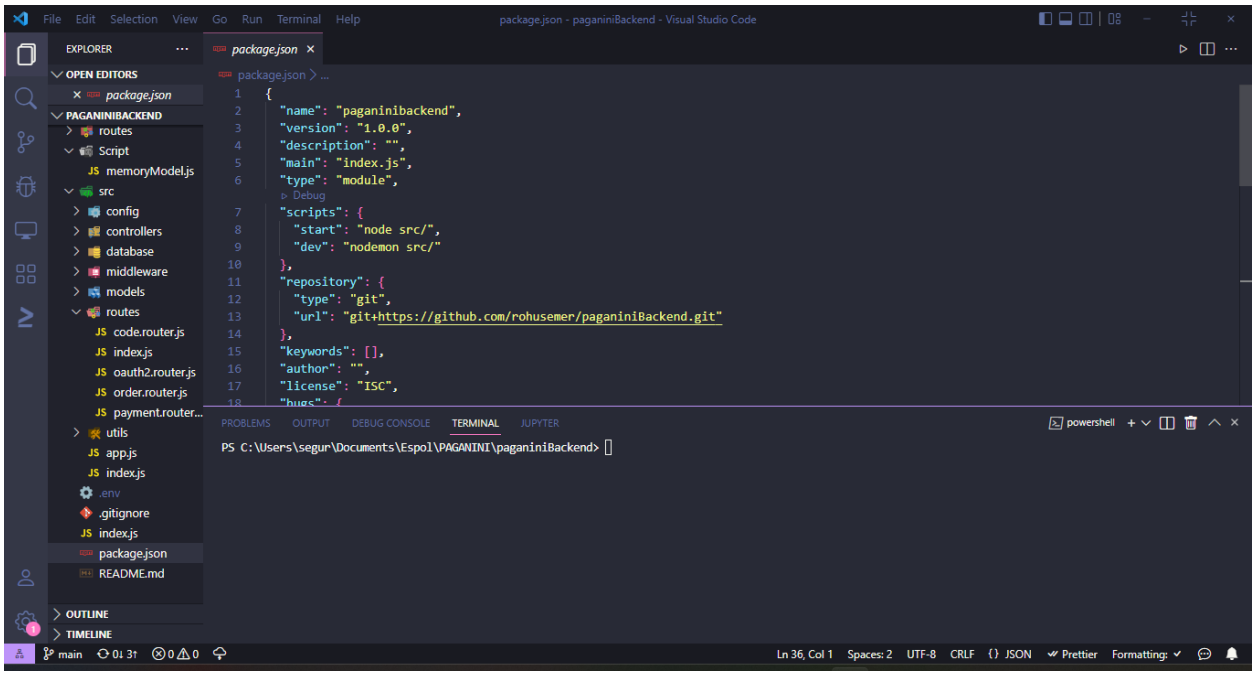


Figura D.7 PAGANINI en Visual *Code*. [autoría propia]

Cree un archivo *.env* y proceda escribir las siguientes credenciales de conexión, tal como se muestra en la Figura D.8.

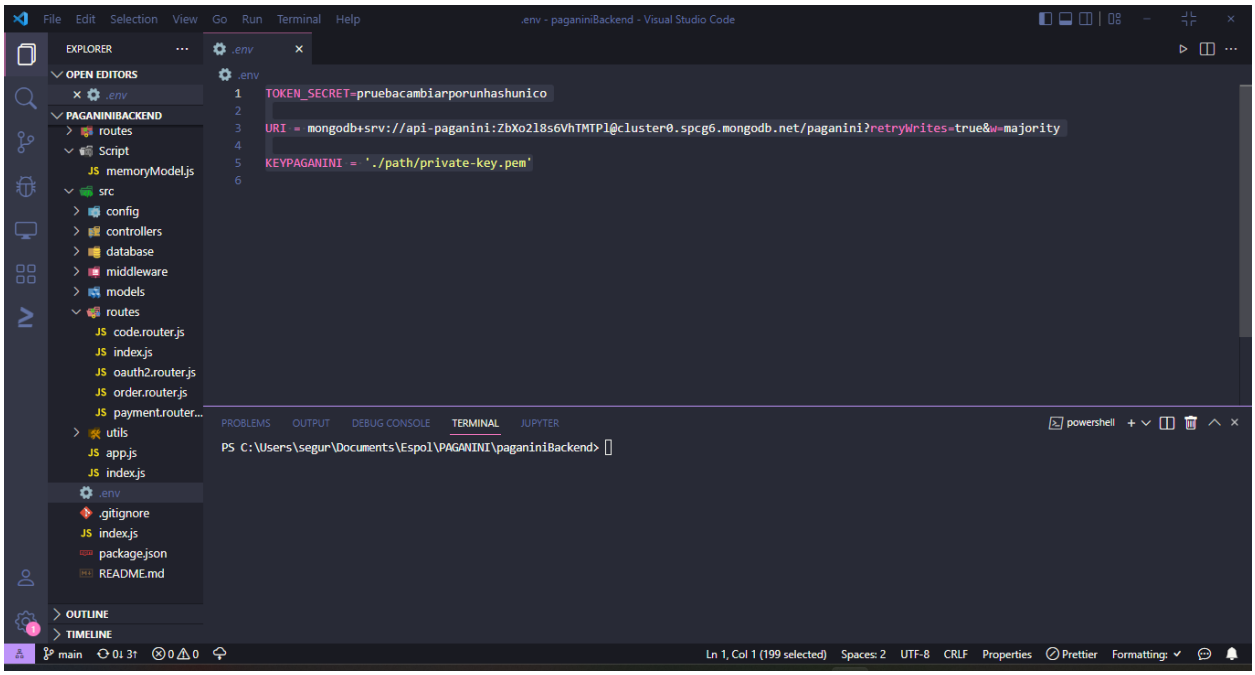


Figura D.8 credenciales de conexión. [autoría propia]

APÉNDICE E

PLAN DE PRUEBAS

En este documento se presentarán las pruebas de integración necesarias para corroborar el funcionamiento del prototipo desarrollado. Dichas pruebas corresponden a los procesos principales del sistema y a la seguridad de la información.

PRUEBAS

A continuación, en la Tabla E.1, se muestran las pruebas de integración principales del sistema. Los códigos mencionados en las pruebas se encuentran en la sección de Códigos de estado.

Tabla E.1 Plan de pruebas. [autoría propia]

Categoría	Nombre	Descripción	Resultado esperado	Correcto	Incorrecto
Acceso al Sistema	Autenticación y autorización	Se debe autenticar al usuario y entregarle la información de acceso como respuesta.	Un objeto con información del token de acceso, tiempo de validez del token y el tipo de token. Se muestra el formulario en el *frontend*.	x	
	Cliente no registrado	Se envían credenciales del cliente que no existen en la base de datos del sistema.	Un objeto indicando un error de autenticación del cliente. Se muestra la pantalla de cliente no autorizado en el *frontend*.	x	
Validación	Tarjeta válida	Se envían los datos de prueba correctos de la tarjeta del cliente.	Se inicia el proceso de verificación	x	
	Número de tarjeta inválido	Se envía el número de tarjeta erróneo desde el formulario de pago.	Se muestra el mensaje en el *frontend* indicando el error de tarjeta inválida.	x	
	Código CVV inválido	Se envía el código CVV inválido desde el formulario de pago.	Se muestra el mensaje en el *frontend* indicando el error de código CVV inválido.	x	

	Detección de fraude	Se envían los datos de la tarjeta del cliente erróneos desde el formulario de pago.	Se muestra el mensaje en el *frontend* indicando el error de detección de fraude.	x	
Verificación	Envío de mensaje de confirmación	Se debe enviar un mensaje de texto al cliente luego de comprobar que la información de la tarjeta ingresada sea correcta.	Se almacena el código de verificación en la base de datos de Redis en el *backend* y se envía un objeto indicando el ID de la transacción y el ID del pago. Se muestra la pantalla de ingreso de código de verificación en el *frontend*.	x	
	Código de verificación incorrecto	Se envía un código de verificación incorrecto desde el *frontend*.	Se actualiza la transacción a un estado de cancelación por rechazo del código de verificación. Se muestra el mensaje en el *frontend* indicando el error.	x	
Transacción	Creación de transacción	Se crea una transacción en la base de datos del sistema la cual será actualizada en función de las respuestas de los procesos involucrados.	Un registro de transacción en la base de datos.	x	
	Estado de transacción en espera	Cuando se complete el primer proceso de pago, la transacción debe tener el código de estado *0090*.	Transacción actualizada en la base de datos	x	
	Estado de transacción aprobado	Cuando se complete el proceso de pago completo sin errores, la transacción se actualizará con el código de estado *0000*.	Transacción actualizada en la base de datos.	x	

	Estado de transacción con errores.	Cuando ocurra un error en el proceso de pago, la transacción se actualizará con el código correspondiente al error descrito.	Transacción actualizada en la base de datos.	x	

DATOS DE PRUEBA

Tabla E.2 Números de tarjeta. [autoría propia]

Número	Tipo de tarjeta	Error
4009348888881881	*VISA*	Válida
4005519200000004	*VISA*	Válida
5555555555554444	*MASTERCARD*	Válida
2223000048400011	*MASTERCARD*	Válida
4000111111111115	*VISA*	Inválida
5105105105105100	*MASTERCARD*	Inválida
4000111111111511	*VISA*	Fraude detectado

Tabla E 3 Códigos CVV

Código	Error
200	Inválido
201	Inválido

Tabla E.4 Códigos de estado. [autoría propia]

Código	Razón	Descripción
0000	*APPROVED*	Pago recibido con éxito
0010	*AUTHORIZATION_FAILED*	Error en la autenticación por falta de cabecera de autorización o credenciales inválidas
0011	*AUTHORIZATION_FAILED*	Token inválido
0020	*DECLINED*	Tarjeta inválida
0021	*DECLINED*	Fraude detectado
0022	*DECLINED*	Error en el procesador de pagos
0023	*DECLINED*	Fondos insuficientes
0024	*DECLINED*	Código CVV incorrecto

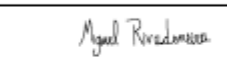
0025	*DECLINED*	Error en transacción
0026	*DECLINED*	Error en la verificación de la tarjeta
0060	*INVALID_CLIENT*	No se encuentran las credenciales del cliente
0061	*INVALID_CLIENT*	Fallo en la autenticación del cliente
0070	*INVALID_GRANT_TYPE*	Tipo de acceso no permitido
0080	*INVALID_DATA*	Datos incorrectos
0090	*ON_HOLD*	Transacción en espera para liquidación
0050	*VERIFICATION_CODE_ERROR*	Código de verificación incorrecto
0040	*TRANSACTION_ERROR*	Transacción cancelada

APÉNDICE F

ACTA DE ACEPTACIÓN

Acta de Reunión

1	DATOS GENERALES		
Tema:	Revisión del prototipo final		
Fecha:	27/08/2022	**Horario:**	9:00-10:00

2.	ASISTENTES	
Cargo	**Nombre**	**Firma**
Estudiante	Miguel Rivadeneira	
Estudiante	Ronny Segura	
Cliente	Leonardo Castro	

3.	TEMAS TRATADOS / ACUERDOS
1.	Aceptación de pruebas de integración del prototipo con aplicativo Jama Sana.
2.	Aceptación del prototipo final-
3.	Aceptación de entregables del proyecto.

Figura F.1 Acta de aceptación firmada por el cliente. [autoría propia]