



**UNIVERSIDAD DE TALCA  
FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERÍA CIVIL EN COMPUTACIÓN**

**Desarrollo de Back-End basado en una  
Arquitectura de Microservicios para la gestión y  
reventa de dominios .CL de la empresa Haulmer  
Inc.**

**HERNÁN ARTURO GÁLVEZ NECULMAN**

Profesor Guía: LUIS GREGORIO SILVESTRE QUIROGA

Memoria para optar al título de  
Ingeniero Civil en Computación

Curicó – Chile  
Enero, 2019

## CONSTANCIA

La Dirección del Sistema de Bibliotecas a través de su encargado Biblioteca Campus Curicó certifica que el autor del siguiente trabajo de titulación ha firmado su autorización para la reproducción en forma total o parcial e ilimitada del mismo.



Two circular stamps and signatures. The left stamp is from the 'DIRECCIÓN SISTEMA DE BIBLIOTECAS UNIVERSIDAD DE TALCA' with a signature over it. The right stamp is from the 'SISTEMA DE BIBLIOTECAS CAMPUS CURICO' with a signature over it.

Curicó, 2019

*Dedicado a Juana, Nivia y Humberto.*

## AGRADECIMIENTOS

**S**on muchas las personas que han contribuido al proceso y conclusión de esta memoria. En primer lugar quiero agradecer a mis padres, Manuel y Yovana, por ser los principales promotores de mis sueños, gracias a ellos por cada día confiar y creer en mí y en mis expectativas. También por enseñarme que cada obstáculo se puede superar con perseverancia y esfuerzo. A mis hermanos, Nacho y Matilde, por su apoyo y motivación para cada día continuar sin tirar la toalla.

No ha sido sencillo terminar este proceso, pero gracias a sus aportes, amor, bondad y apoyo, los momentos complicados se han notado menos. Les agradezco, y hago presente mi afecto hacia ustedes, mi hermosa familia Gálvez-Neculman.

También quiero agradecer a mi profesor guía, Luis Silvestre, por cada detalle y momento dedicado para aclarar cualquier tipo de duda que me surgiera. Por su enorme apoyo y motivación brindada durante este proceso, además por su constante preocupación para cumplir todas las metas propuestas asociadas al desarrollo de esta memoria.

El desarrollo de esta memoria no fue un camino fácil, pero durante todo este tiempo pude disfrutar cada momento, y no fue porque simplemente me dispuse a que así fuera, sino porque mis amigos siempre estuvieron ahí, apoyándome y diciéndome palabras de aliento cuando más lo necesité, en los momentos difíciles.

Agradezco a la empresa Haulmer Inc. por otorgarme la oportunidad de realizar este proceso bajo su infraestructura y trabajar conjuntamente con su equipo. Sin duda el ambiente laboral fue agradable y me sentí respaldado de principio a fin.

Finalmente quiero decir que este es un momento muy especial y que espero que perdure en el tiempo, no solo en las personas que agradecí, sino también a quienes invirtieron tiempo para echarle una mirada a mi proyecto de titulación.

## RESUMEN

Actualmente existe una gran variedad de proveedores de servicios de internet, entre ellos están los que se dedican a la venta de nombre de dominios. Sin embargo, los clientes buscan adquirir nombres de dominios con TLD .CL y no todos los proveedores lo ofrecen.

En esta memoria trata de solventar los requerimientos básicos que exige NIC Chile a la empresa Haulmer para que pueda revender dominios .CL utilizando estándares y protocolos de venta compatibles con los requerimientos exigidos. Además se implementó una plataforma que gestiona a los revendedores y clientes pertenecientes al negocio.

En general, la metodología utilizada permitió establecer los requisitos iniciales de la problemática, además de crear los diseños necesarios para la implementación de la solución. Para desarrollar la propuesta de solución se utilizaron herramientas que permitieron implementar los microservicios y la plataforma que unifica la comunicación entre ellos. Estas herramientas son Python, Laravel y MySQL Workbench.

La implementación de la solución se realizó según las fases de la metodología que sugirió la empresa para el desarrollo. En cada fase, se especificaron los requisitos, el diseño y construcción del software.

Como resultado, se obtuvo microservicios y una plataforma de venta que permite la gestión y reventa de nombre de dominios .CL. Las principales funciones son la compra de dominios, registro de usuarios y entrega de información acerca de un dominio registrado.

Para la validación de la plataforma se aplicaron encuestas a los usuarios de prueba sobre si cumple o no con las características de funcionalidad y utilidad.

Los resultados obtenidos en la validación de la plataforma son satisfactorios dado que los usuarios de prueba interactuaron correctamente, esto se evidencia en el análisis de los resultados. Finalmente para trabajos futuros se considera la opción de integrar medios de pagos reales, validación de caracteres especiales, visualización de información relevante y notificación del proceso de compra.

## TABLA DE CONTENIDOS

	página
<b>Dedicatoria</b>	<b>I</b>
<b>Agradecimientos</b>	<b>II</b>
<b>Tabla de Contenidos</b>	<b>III</b>
<b>Índice de Figuras</b>	<b>VI</b>
<b>Índice de Tablas</b>	<b>VIII</b>
<b>Resumen</b>	<b>IX</b>
<b>1. Introducción</b>	<b>10</b>
1.1. Motivación . . . . .	11
1.2. Contexto del Proyecto . . . . .	11
1.3. Descripción del Problema . . . . .	12
1.4. Propuesta de Solución . . . . .	13
1.5. Objetivos . . . . .	14
1.5.1. Objetivo General . . . . .	14
1.5.2. Objetivos Específicos . . . . .	14
1.6. Alcances . . . . .	15
1.7. Trabajos Relacionados . . . . .	15
<b>2. Marco Teórico</b>	<b>17</b>
2.1. Protocolo EPP . . . . .	17
2.2. Arquitectura de Microservicios . . . . .	19
2.3. Back-End . . . . .	24
2.4. Metodología de Desarrollo Híbrido . . . . .	26
2.5. Metodología de Evaluación basada en Experimentos . . . . .	28
2.6. Herramientas de Desarrollo . . . . .	30

<b>3. Desarrollo de la Propuesta de Solución</b>	<b>33</b>
3.1. Concepción del Proyecto . . . . .	33
3.2. Diseño Arquitectónico . . . . .	35
3.2.1. Arquitectura Física . . . . .	35
3.2.2. Arquitectura Lógica . . . . .	38
3.3. Implementación de la Metodología Híbrida . . . . .	40
3.3.1. Investigación . . . . .	40
3.3.2. Diseño Funcional . . . . .	40
3.3.3. Diseño . . . . .	50
3.3.4. Requisitos . . . . .	59
3.3.5. Construcción y Pruebas . . . . .	62
<b>4. Validación de la Solución</b>	<b>70</b>
4.1. Fases de Experimentación . . . . .	70
4.1.1. Definición . . . . .	70
4.1.2. Diseño . . . . .	72
4.1.2.1. Características de Evaluación . . . . .	72
4.1.2.2. Instrumentos de Medición . . . . .	72
4.1.2.3. Proceso de Experimentación . . . . .	73
4.1.3. Ejecución . . . . .	73
4.1.4. Experimentación . . . . .	74
4.1.5. Análisis . . . . .	77
4.1.5.1. Resultados Usuario Cliente . . . . .	78
4.1.5.2. Resultados Usuario Partner . . . . .	81
4.1.5.3. Resultados Evaluación General . . . . .	84
<b>5. Conclusiones y Trabajo Futuro</b>	<b>88</b>
5.1. Conclusiones . . . . .	88
5.2. Lecciones Aprendidas . . . . .	90
5.3. Trabajos a Futuro . . . . .	91

<b>Bibliografía</b>	<b>92</b>
<b>Anexos</b>	<b>94</b>
<b>Anexo A. Documentos de Apoyo</b>	<b>95</b>
A.1. Documento de Requisitos . . . . .	95
<b>Anexo B. Complementos de Evaluación</b>	<b>107</b>
B.1. Lista de Actividades de Usuarios . . . . .	107
B.2. Encuesta Realizada . . . . .	113

## ÍNDICE DE FIGURAS

	página
2.1. Intercambio de mensajes entre cliente-servidorEPP. . . . .	18
2.2. Imagen de referencia de una arquitectura de microservicios. . . . .	20
2.3. Capas de servicios en una arquitectura de software. . . . .	21
2.4. Representación gráfica de arquitectura de microservicios. . . . .	22
2.5. Back-end y Front-end en un sistema informático [10]. . . . .	25
2.6. Diagrama de las etapas de la Metodología Híbrida de Haulmer Inc. .	27
2.7. Diagrama de secuencia del proceso de compra de un nombre de do- minio en particular. . . . .	29
2.8. Ranking de popularidad de bases de datos [15]. . . . .	30
3.1. Diagrama de actividad del flujo de compra de un dominio. . . . .	34
3.2. Arquitectura Física del Sistema. . . . .	36
3.3. Arquitectura Lógica del Sistema. . . . .	38
3.4. Diagrama de actividad de registro de dominio. . . . .	41
3.5. Diagrama de actividad para renovar vigencia de un dominio. . . . .	42
3.6. Diagrama de actividad para borrar un dominio. . . . .	43
3.7. Diagrama de actividad para obtener información de un dominio. . . .	43
3.8. Diagrama de actividad para registrar un contacto. . . . .	44
3.9. Diagrama BPMN para acceder a la plataforma Partner. . . . .	45
3.10. Diagrama BPMN para listar clientes. . . . .	46
3.11. Diagrama BPMN para registro de usuarios. . . . .	47
3.12. Diagrama BPMN para comprar un nombre de dominio. . . . .	49
3.13. Diagrama entidad/relación conector NIC Chile. . . . .	51
3.14. Diagrama entidad/relación plataforma partner. . . . .	52
3.15. Diagrama de clases de la plataforma partner. . . . .	54
3.16. Mock-up crear cliente. . . . .	55
3.17. Mock-up editar cliente. . . . .	56
3.18. Mock-up editar cliente. . . . .	56
3.19. Mock-up eliminar cliente. . . . .	57
3.20. Mock-up listar cliente. . . . .	58
3.21. Mock-up listar las transacciones . . . . .	59

3.22. Interfaz para verificar disponibilidad dominio. . . . .	67
3.23. Interfaz cuando el dominio está disponible. . . . .	67
3.24. Interfaz cuando el dominio no está disponible. . . . .	67
3.25. Interfaz del formulario para comprar un dominio. . . . .	68
3.26. Interfaz del formulario para renovar un dominio. . . . .	68
3.27. Interfaz del formulario para crear nuevos contactos. . . . .	69
4.1. Usuarios de prueba. . . . .	77
4.2. Caracterización de los usuarios de prueba. . . . .	77
4.3. Resultado de evaluación de la característica funcionalidad. . . . .	78
4.4. Tabla de datos de la funcionalidad de la plataforma. . . . .	79
4.5. Resultado de evaluación de la característica utilidad. . . . .	80
4.6. Tabla de datos de la utilidad de la plataforma. . . . .	80
4.7. Resultado de evaluación de la característica funcionalidad. . . . .	81
4.8. Tabla de datos de la funcionalidad de la plataforma. . . . .	82
4.9. Resultado de evaluación de la característica utilidad. . . . .	83
4.10. Tabla de datos de la utilidad de la plataforma. . . . .	83
4.11. Resultados evaluación de fallos de los clientes en el sistema. . . . .	84
4.12. Resultados evaluación de fallos de los partners en la plataforma. . . . .	84
4.13. Resultados evaluación general del sistema. . . . .	87
4.14. Tabla de datos de la evaluación general del sistema. . . . .	87

## ÍNDICE DE TABLAS

	página
1.1. Tabla comparativa de proveedores de nombres de dominios. . . . .	15
1.2. Entidades que permiten reventa de servicios. . . . .	16
1.3. Proveedores de nombres de dominios extranjeros. . . . .	16
2.1. Ventajas y Desventajas de los microservicios [8]. . . . .	23
3.1. Requisitos de usuario para registrar un dominio. . . . .	60
3.2. Requisitos de usuario para verificar un dominio. . . . .	60
3.3. Requisitos de usuario para renovar un dominio. . . . .	61
3.4. Requisitos de usuario para registrar un dominio. . . . .	61
4.1. Actividades de la ejecución de la experimentación para el usuario cliente.	74
4.2. Actividades de la ejecución de la experimentación para el usuario partner. . . . .	75
4.3. Tabla resumen de la ejecución de la experimentación. . . . .	76

# 1. Introducción

---

Haulmer Inc. es una empresa ubicada en la ciudad de Curicó, que se dedica al desarrollo de productos y venta de servicios en Internet. Actualmente Haulmer Inc. ofrece servicio de venta de dominio a sus clientes, pero cuando un cliente en particular necesita adquirir un dominio .CL tiene que obtenerlo a través de un proveedor externo.

En la actualidad Haulmer Inc. quiere expandir el negocio ofreciendo a sus clientes la posibilidad de obtener un dominio con TLD(*Top-Level Domain*, en español: dominio de nivel superior) .CL y a su vez ampliar el mercado objetivo. Para ello se requiere implementar una conexión e integración con alguna entidad externa que le permita la gestión y reventa de este producto. Algunas entidades que ofrecen dominios .Cl son: Bluehosting<sup>1</sup>, NIC Chile<sup>2</sup>, Hosting<sup>3</sup>, Planeta Hosting<sup>4</sup>, DonWeb<sup>5</sup>, entre otros.

Sin embargo, implementar esta conexión con un proveedor externo de dominios para revender este producto no es algo trivial, puesto que la empresa Haulmer debe cumplir con ciertos estándares y protocolos de comunicación. Luego de finalizar el desarrollo e implementación del proyecto, se espera que la empresa permita gestionar y revender dominios .CL. Para ello se pretende implementar una plataforma que cumpla con los estándares y protocolos exigidos por el proveedor. Además, finalmente integrar sus plataformas y flujos de venta.

---

<sup>1</sup><https://www.bluehosting.cl/>

<sup>2</sup><https://www.nic.cl/>

<sup>3</sup><https://www.hosting.cl/>

<sup>4</sup><https://www.planetahosting.cl/>

<sup>5</sup><https://donweb.com/es-cl/>

## 1.1. Motivación

La empresa quiere ampliar su mercado objetivo ofreciendo a sus clientes la posibilidad de adquirir nuevos servicios, como por ejemplo, la compra de dominios con TLD .CL. Por otra parte, también busca otorgar nuevas experiencias para los clientes que quieran formar parte de la empresa Haulmer Inc. y actuar como revendedor de sus productos. Facilitando todas las herramientas necesarias para que puedan operar en igualdad de condiciones bajo sus estándares y protocolos de venta.

Cabe mencionar que para la empresa Haulmer es importante vender dominios con TLD .CL ya que amplía la posibilidad a sus clientes de obtener nombres de dominio con una variada cantidad de TLD disponibles en el mercado. A su vez se posiciona para competir con otras empresas de similares características y así poder captar más clientes incorporando este nuevo servicio.

La expectativa que tiene a futuro la empresa Haulmer luego de finalizar el proyecto es posicionarse a nivel nacional como una de las mejores empresas revendedoras de nombres de dominio. Para ello se pretende utilizar una estrategia que permita a sus clientes ser parte del negocio.

## 1.2. Contexto del Proyecto

NIC Chile situado como único proveedor que actualmente ofrece el servicio de reventa de dominios .CL, exige a sus registradores de nombres de dominio que cumplan con ciertos requisitos para funcionar como revendedor. Es por esto que, la propuesta presentada está sujeta a los términos y condiciones que exige este proveedor externo para gestionar y revender dominios punto CL.

Para la reventa de dominios, la empresa requiere implementar una conexión e integración con NIC Chile, la cual le otorgaría un certificado que lo acredite como registrador autorizado. Sin embargo, para realizar dicha implementación se requiere desarrollar microservicios que permitan la gestión y reventa de dominios, los cuales podría variar entre 3 a 5 microservicios, a quienes conjuntamente llamaremos *ecosistema de microservicios* [1]. Cada microservicio puede contar con servicios más pequeños. Estos interactúan entre sí para un correcto funcionamiento del sistema, llamados *Nano-Servicios*.

Por otra parte, la empresa Haulmer otorgará a sus clientes la opción de ser *Reseller*<sup>6</sup>, quienes podrán revender este producto como estimen conveniente. Sin embargo, es importante considerar la gestión de los *Reseller* y sus clientes en el sistema, ya que ambos actores cumplen un rol fundamental para el negocio, debido a que las peticiones de registro de un dominio de los clientes finales pasaran a través de los estos usuarios. Entonces podemos decir que los *Resellers* serán un canal de comunicación para masificar la venta de dominios .CL que quiere ofrecer la empresa Haulmer.

Cabe destacar que la propuesta sólo contempla el desarrollo back-end del ecosistema de microservicios que realiza la gestión de dominios, *Reseller*<sup>6</sup> y clientes finales, permitiendo realizar las siguientes acciones:

- Gestión de Reseller y sus clientes.
- Gestión y Reventa de dominios.
- Interacción con sistemas externos de la empresa.

### 1.3. Descripción del Problema

La problemática pasa principalmente por la manera en que la empresa realiza el proceso de reventa dominios debido a la falta de requerimientos básicos que exige un proveedor externo para gestionar los dominios punto CL.

Por otra parte, la empresa debe cumplir con estándares y protocolos de reventa de dominios que actualmente le exigen sus proveedores de servicio, lo que conlleva también a respetar reglamentos para la gestión de *Resellers*, quienes podrán revender nombres de dominios como estimen conveniente.

Finalmente cabe mencionar la carencia de la infraestructura tecnológica que posee la empresa para gestionar dominios punto CL. Actualmente sus clientes no pueden obtener este servicio, reduciendo el mercado objetivo y las posibilidades de crecer. Sin embargo, es posible mencionar que la empresa actualmente cuenta con gestión de *Resellers*, pero este servicio está externalizado, por lo tanto la empresa desea una gestión propia de resellers para la toma de decisiones estrategicas.

---

<sup>6</sup>**Revendedor:** adj. Que revende.

## 1.4. Propuesta de Solución

Para procurar resolver el problema y así mejorar la experiencia de los clientes de Haulmer Inc, se considera desarrollar un ecosistema de microservicios para tener un mayor control en las peticiones que realizan. Además, como el sistema está basado en una arquitectura de microservicios permite una escalabilidad [2] más eficiente, en caso que la empresa decida expandir su mercado objetivo agregando nuevos TLD's. Cada microservicio se encarga de implementar una funcionalidad completa del negocio, esto quiere decir que se contempla un microservicio para:

- **Gestión de Usuarios:** Este microservicio permite gestionar los *Resellers* y clientes registrados en el sistema, considerando también la manipulación de la base de datos que involucra la información referente al proceso de compra de un nombre de dominio.
- **Gestor de Peticiones:** Este microservicio permite la gestión de las peticiones de los clientes. Estas peticiones pasan a través de la plataforma de venta de los reseller.
- **Conector Nic Chile:** Microservicio que se conecta con el proveedor de nombres de dominios .CL, que en este caso es NIC Chile. También está encargado de realizar las diferentes peticiones al servidor bajo el protocolo EPP que involucren operaciones sobre nombre de dominio.
- **Servicio “Whois”:** Es un pequeño servicio que toda empresa que vende nombres de dominios debe tener y que permite verificar información acerca de un dominio en particular.

Es importante mencionar que a lo largo del desarrollo del proyecto se pueden crear microservicios adicionales que aporten valor al sistema de acuerdo a las necesidades de la empresa para cumplir con su objetivo principal.

Finalmente, es necesario mencionar que los microservicios que conjuntamente trabajan para realizar la interacción con el servidor de NIC Chile se desarrollan en lenguaje Python. Adicionalmente, la plataforma que gestiona usuarios (resellers y clientes) y dominios se implementa en lenguaje PHP, el cual contempla una capa de presentación para que los usuarios puedan tener una visualización más intuitiva del proceso de compra y venta de un nombre de dominio.

## 1.5. Objetivos

En esta sección se detallan los objetivos específicos y el objetivo general que tiene el desarrollo de este proyecto.

### 1.5.1. Objetivo General

- Desarrollar un ecosistema basado en microservicios que permita la integración de sistemas externos que posibilite la administración y reventa de nombres de dominio .CL, así como también la gestión de Reseller y sus clientes.

### 1.5.2. Objetivos Específicos

- Especificar los requerimientos que satisfacen la reventa y administración de dominios .CL.
- Investigar acerca de la implementación de una Arquitectura de Microservicios y el protocolo EPP.
- Generar un diseño de la capa lógica para la gestión y reventa de dominios .CL basado en una arquitectura de microservicios.
- Implementar la interacción para la reventa de dominios .CL a través de un proveedor externo.
- Desarrollar back-end del microservicio para la gestión y reventa de dominios .CL.
- Desarrollar back-end del microservicio para la gestión de Reseller y sus clientes finales.
- Desarrollar capa de presentación que nos permita visualizar el flujo que tiene la administración y reventa de un dominio .CL.
- Aplicar evaluación del ecosistema de microservicios en conjunto con el cliente para comprobar una completa integración tanto con la plataforma de venta de los Reseller como con sistemas externos de la empresa.

## 1.6. Alcances

- En este proyecto se espera implementar un ecosistema de microservicios, por lo tanto se propone desarrollar la capa lógica del sistema que permita la interacción con un proveedor externo de venta de dominios punto CL.
- En este proyecto se desarrolla una interfaz de usuario, la cual permite la gestión de usuarios registrados (resellers y clientes) y además la gestión de dominios.
- Este proyecto se limita a utilizar el protocolo de comunicación EPP para la interacción con el proveedor de dominios NIC Chile. Este proveedor exige a sus clientes utilizar dicho protocolo de comunicación.

## 1.7. Trabajos Relacionados

En la actualidad, existen una variedad de empresas que ofrecen el servicio de venta de nombres de dominios (.com, .net, .info, etc.) a sus clientes en Chile. Algunos de estos proveedores de dominios se listan a continuación:

Proveedores de nombres de dominio		
Nombre	Ventajas	Desventajas
- Bluehosting - SitioHost - Rackeo - Hosty - Livehost	Estas plataformas cuenta con la venta de servicios de internet y tiene una interfaz de usuario simple de usar. Estas entidades ofrecen entre sus servicios la venta de nombres de dominios, como el .com, .pe, .name, .us, .biz, entre otros.	No poseen venta de dominios .CL.

TABLA 1.1: Tabla comparativa de proveedores de nombres de dominios.

Por otra parte, existen entidades que entregan a sus clientes la posibilidad de ser revendedores de sus servicios, entre los cuales encontramos:

<b>Proveedores de reventa de servicios</b>		
<b>Nombre</b>	<b>Ventajas</b>	<b>Desventajas</b>
- DonDominio - Resellerclub	Estas entidades entre sus servicios ofrecen la venta nombres de dominios, hosting y también otorgan la capacidad de revender sus servicios.	No poseen venta de dominios .CL.

TABLA 1.2: Entidades que permiten reventa de servicios.

Finalmente, a nivel internacional podemos encontrar empresas como United Domains<sup>7</sup>, Marcaria<sup>8</sup> y GrupoHost<sup>9</sup> quienes proporcionan servicios de internet, específicamente dominios .CL:

<b>Proveedores de nombres de dominio extranjeros</b>		
<b>Nombre</b>	<b>Ventajas</b>	<b>Desventajas</b>
- United Domains - Marcaria - GrupoHost	Estas plataformas cuenta con la venta de nombres de dominios .CL, hosting, entre otros servicios.	Los precios son demasiados costosos, los cuales van entre los \$13.000 a \$105.000 pesos chilenos. Además, hay que pagar con tarjetas de créditos que cuenten con dólares o euros.

TABLA 1.3: Proveedores de nombres de dominios extranjeros.

<sup>7</sup>**Página web:** <https://www.united-domains.de/domain-registrieren/>.

<sup>8</sup>**Página web:** <https://www.marcaria.com/ws/es/dominios/chile-registrar-dominio-cl>.

<sup>9</sup>**Página web:** <https://www.grupohost.cl/dominios/cl>.

## 2. Marco Teórico

---

### 2.1. Protocolo EPP

Garvin Brown [3] señala que EPP es un protocolo basado en XML de estados utilizado por la industria de Internet particularmente por los *Registrar*<sup>1</sup> y *Registry*<sup>2</sup> en la gestión de nombres de dominio (registrar, renovar, modificar, eliminar, transferir) y otros elementos en un entorno de Sistema de Registro Compartido.

De acuerdo con Scott Hollenbeck[4], el protocolo EPP nos otorga cuatro elementos básicos de servicio, que se señalan a continuación:

- Descubrimiento de servicios.
- Comandos.
- Respuestas.
- Marco de trabajo que soporta la gestión y relación de solicitudes y respuestas de objetos.

Todos los comandos EPP son atómicos, es decir, no hay éxito ni fallo parcial, sólo existe una ejecución completa o incompleta del comando. Están diseñados para que puedan volverse idempotentes, esto significa que ejecutar un comando más de una vez tiene el mismo efecto neto en el estado del sistema que ejecutar el comando con éxito una única vez.

---

<sup>1</sup>**Registrador:** Quien registra dominios y provee servicios a sus clientes.

<sup>2</sup>Proveedor de dominios, por ejemplo, Nic Chile.

Por otra parte, el protocolo EPP se puede estratificar sobre múltiples protocolos de transporte. Está protegido mediante protocolos de seguridad inferior, lo cual permite la identificación, autenticación y participación en una serie de intercambios de comandos y respuestas iniciadas por el cliente. Es importante mencionar que el servidor EPP debe responder a la comunicación iniciada por el cliente, ya sea a través de una solicitud de conexión por intermedio de una capa inferior o un servicio EPP. El servidor deber ser capaz de entregar una respuesta rápida a cada comando recibido, siendo capaz de coordinar la respuestas y describir los resultados del procesamiento del comando [4]. A continuación en la Figura 2.1, se presenta un diagrama de procesos que ilustra el intercambio de mensajes entre el cliente y el servidor EPP:

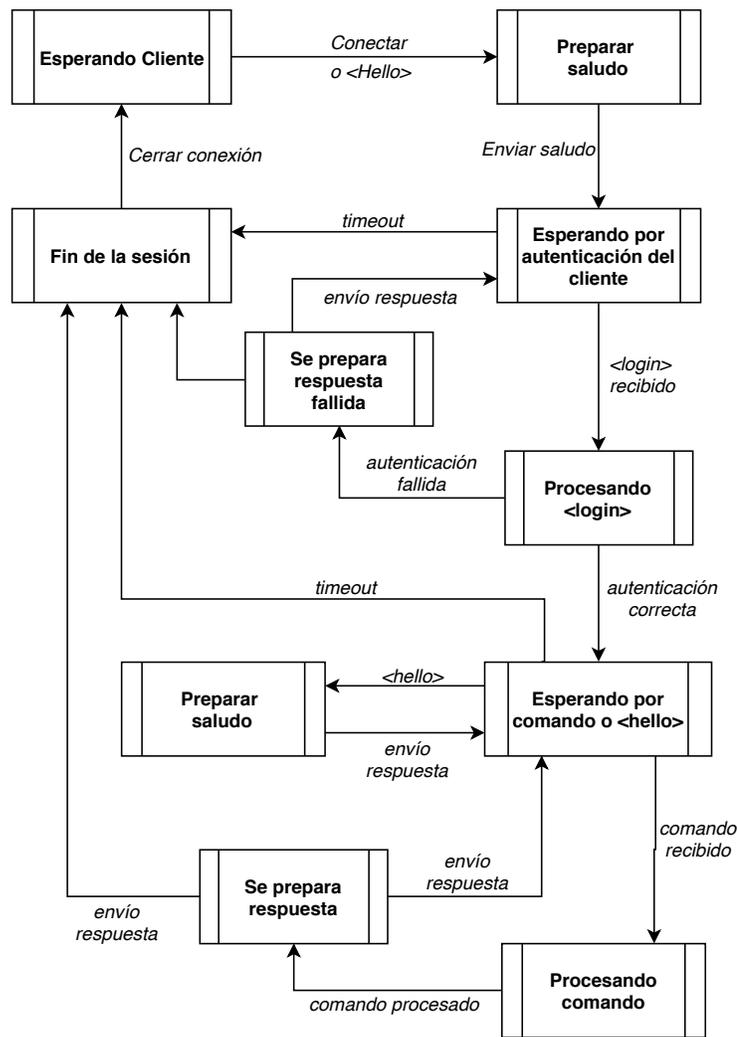


FIGURA 2.1: Intercambio de mensajes entre cliente-servidor EPP.

## 2.2. Arquitectura de Microservicios

Para lograr un mayor entendimiento del concepto de Arquitectura de Microservicios, derivaremos el término en dos definiciones más amplias, como lo son una Arquitectura de Software y un Servicio.

En primer lugar una *arquitectura de software* es la estructura o estructuras del sistema que comprenden los elementos del software, los elementos visibles externamente, las propiedades de esos elementos y las relaciones que existe entre ellos [5]. Si se selecciona la definición oficial de arquitectura de software según la IEEE Std 1471-2000, se señala lo siguiente: “*La Arquitectura del Software es la organización fundamental de un sistema formada por sus componentes, las relaciones entre ellos y el contexto en el que se implantarán, y los principios que orientan su diseño y evolución*” [6].

Por lo general. no es común inventar una nueva arquitectura para desarrollar un software, sino más bien se adopta una arquitectura existente de acuerdo a los requisitos del mismo. Dentro de las arquitecturas más comunes encontramos:

- Descomposición Modular.
- Cliente-Servidor.
- Modelo Vista Controlador.
- Orientada a Servicios.
- Arquitectura de Microservicios.

Para efecto del desarrollo de este proyecto se utiliza una arquitectura de microservicios ya que la empresa Haulmer desarrolla sus software basados en esta arquitectura. A continuación en la Figura 2.2 se ilustra un diseño de una arquitectura de microservicios:

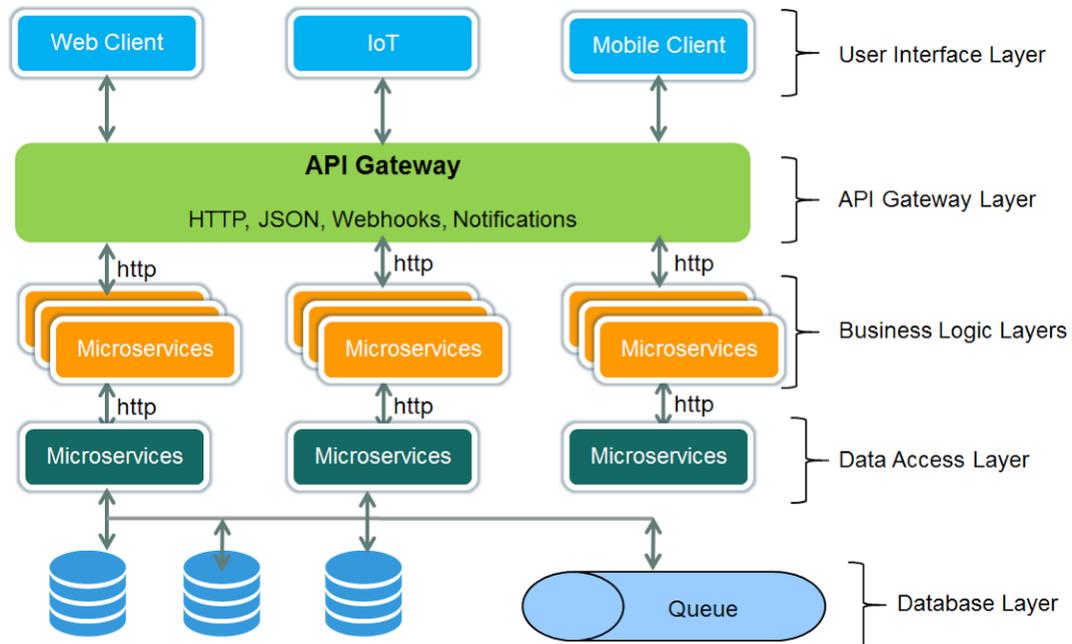


FIGURA 2.2: Imagen de referencia de una arquitectura de microservicios.

Por otra parte, cuando hablamos del termino *servicios* en el área de informática aludimos a una **representación de las operaciones, acciones o actividades que no pertenecen conceptualmente a ningún objeto de dominio concreto** [7]. Los servicios no tienen ni estado propio ni un significado más allá que la acción que los definen, además deben cumplir 3 características principales:

- La operación que lo define está relacionada con un concepto de dominio, pero no es natural modelarlo como una entidad o un value object [7].
- Su interfaz se especifica usando otros elementos del modelo de dominio [7].
- La operación no tiene estado, por lo que un cliente en particular podría usar cualquier instancia del servicio sin tener en cuenta las operaciones que se han realizado con anterioridad en esa instancia [7].

Es posible dividir los servicios en tres tipos según la relación que existe con el dominio.

**Servicios de Dominio:** Son responsables del comportamiento más específico del dominio, es decir, realizan acciones que no dependen de la aplicación concreta que estemos desarrollando [7].

**Servicios de Aplicación:** Son responsables del flujo principal de la aplicación, es decir, son los casos de uso de nuestra aplicación. Su función es coordinar entidades, value objects, servicios de dominio y servicios de infraestructura para llevar a cabo una acción [7].

**Servicios de Infraestructura:** Son los comportamientos que no pertenece realmente al dominio de la aplicación pero que debemos ser capaces de realizar como parte de este [7].

En la Figura 2.3 se ilustran las capas de servicios presentes en una arquitectura de software:

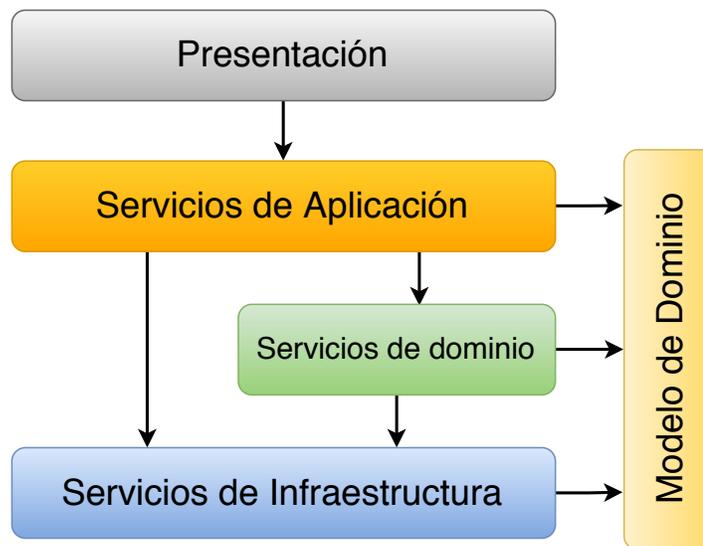


FIGURA 2.3: Capas de servicios en una arquitectura de software.

Una vez definido los términos de *arquitectura de software* y *servicios* podemos explicar lo que es una *Arquitectura de Microservicios*. De acuerdo con Martin Fowler y James Lewis [8], expertos en el tema, nos señalan en su artículo “*Microservices*” que este estilo de arquitectura es un enfoque para desarrollar una única aplicación como un conjunto de pequeños servicios, cada uno de los cuales se ejecuta en su propio proceso y se comunica con mecanismos ligeros, a menudo una API de recursos HTTP.

Existe un mínimo de gestión centralizada de estos servicios, que pueden estar escritos en diferentes lenguajes de programación y utilizar diferentes tecnologías de almacenamiento de datos. En la Figura 2.4, se ilustra el diseño de una arquitectura de microservicios y además cómo los servicios que están programados en lenguajes diferentes se comunican con bases de datos distintas:

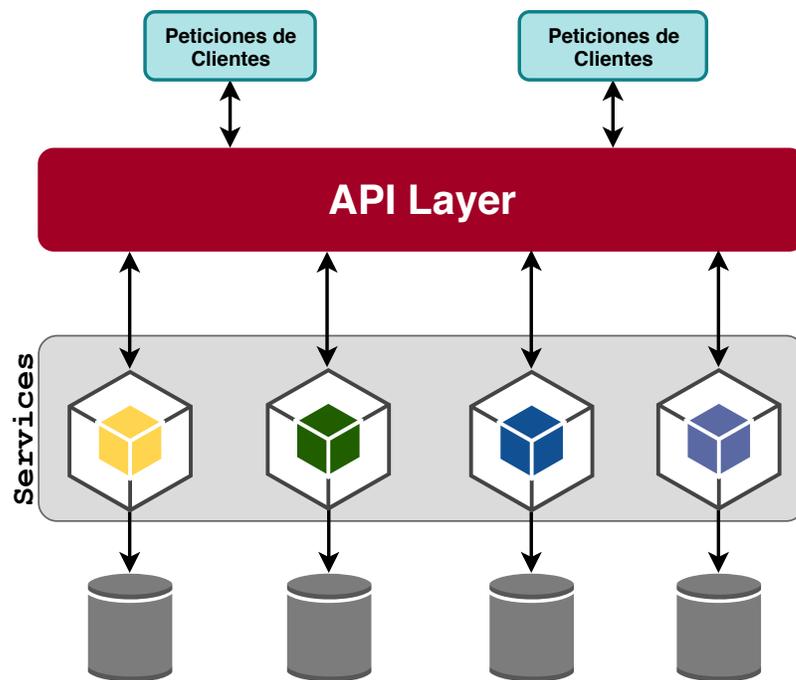


FIGURA 2.4: Representación gráfica de arquitectura de microservicios.

Sam Newman [8] hace alusión a las ventajas y desventajas que conlleva implementar una arquitectura de microservicios para el desarrollo de un software, las cuales se describen en la Tabla 2.1:

Ventajas	Desventajas
Cada servicio es relativamente pequeño y esto ayuda al entendimiento por parte del desarrollador.	Los desarrolladores deben hacer frente a la complejidad adicional de trabajar con un sistema distribuido.
Cada servicio puede ser desplegado independientemente de otros.	Las herramientas de desarrollo/IDEs están orientadas a construir aplicaciones monolíticas.
Mejora el aislamiento de fallas.	El testing o las pruebas son más difíciles.
Fácil de escalar ya que cada equipo puede desarrollar, desplegar y escalar su servicio independientemente de los otros equipos.	Los desarrolladores deben implementar mecanismos de comunicación entre procesos.
Cada servicio puede ser desarrollado y desplegado independientemente.	La implementación de transacciones distribuidas que atraviesan múltiples servicios es compleja.
Elimina el compromiso de atarse con una tecnología en particular.	Complejidad operativa de despliegue y gestión cuando existen muchos tipos de servicios diferentes.

TABLA 2.1: Ventajas y Desventajas de los microservicios [8].

### 2.3. Back-End

Si adoptamos el término de back-end como nos explica la compañía de educación *Codecademy* en un artículo [9], nos describe el concepto mencionado como toda la tecnología necesaria para procesar la solicitud entrante sobre un sistema informático, para así poder generar y enviar la respuesta al cliente. Esto típicamente incluye tres partes principales:

- **El servidor:** Es un computador que recibe las peticiones entrantes. En el mercado actualmente hay máquinas fabricadas y optimizadas para esta función. Sin embargo, cualquier computador que tenga conexión a una red puede ejercer el rol de servidor.
- **La aplicación:** Es la rutina que se ejecuta en el servidor que contiene la lógica de cómo responder las peticiones basadas en HTTP de acuerdo a un enrutamiento. Por otra parte, existen los *middleware* que es un código que se ejecuta entre el servidor que recibe una solicitud y el que envía una respuesta. Los *middleware* pueden modificar el objeto de solicitud, consultar la base de datos o procesar la solicitud entrante. Eventualmente, se llama a una función de *middleware* que termina el ciclo de solicitud-respuesta enviando una respuesta HTTP al cliente.
- **La base de datos:** Las bases de datos se utilizan para organizar y mantener los datos de forma persistente en el sistema. Sin embargo, cuando se almacenan los datos en una base de datos se reduce la carga de la memoria principal de la CPU del servidor, adicionalmente permite la recuperación de datos frente a un eventual fallo del servidor (bloqueo o pérdida de energía).

Muchas peticiones enviadas al servidor pueden requerir una consulta a la base de datos. Un cliente puede solicitar información que está almacenada en la base de datos, o un cliente puede enviar datos con su solicitud para ser agregados a la base de datos.

Es importante mencionar que los tres componentes anteriormente descritos cumplen un rol fundamental para consolidar un backend y que conjuntamente trabajan para un completo funcionamiento y despliegue.

En la Figura 2.5 se hace referencia acerca del rol que cumple el back-end en un sistema informático:

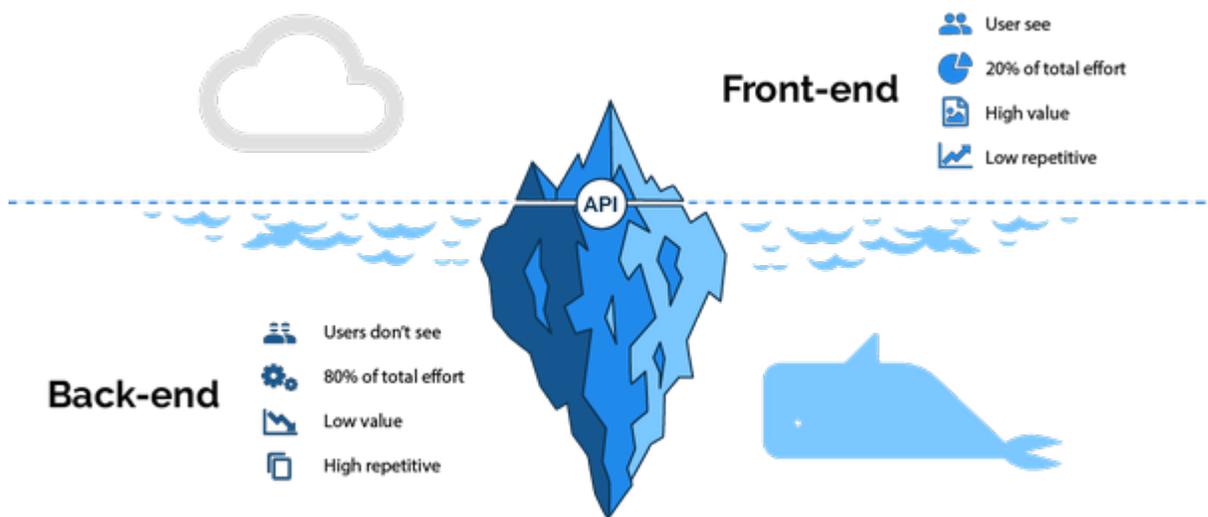


FIGURA 2.5: Back-end y Front-end en un sistema informático [10].

Finalmente es necesario considerar una definición más acotada que realizó la revista PCMag, donde se señala que el término back-end a menudo se refiere al “*sistema de gestión de base de datos (DBMS), que es el almacén de los datos que residen en un servidor. También puede referirse al software en un servidor Web o servidor de aplicaciones que realiza el procesamiento iniciado (Parte lógica) por una persona que utiliza un navegador Web (lado cliente)*” [11].

## 2.4. Metodología de Desarrollo Híbrido

Para el desarrollo de este proyecto se utiliza una metodología de desarrollo híbrida. Esta metodología consiste en emular las fases genéricas de una metodología de desarrollo tradicional [12], las cuales son *análisis de requisitos, diseño, construcción, pruebas, despliegue/mantenimiento*. En la metodología híbrida la documentación es más simple y menos formal, por lo que las fases de *Análisis de Requisitos* y *Diseño*, al final entregan una documentación más acotada y precisa en su contenido. Estas dos fases se fusionan para dar lugar a cuatro etapas más precisas en cuanto a desarrollo. A continuación se explica cada una de las etapas:

- **Investigación:** Se realiza toda la investigación y el estudio previo antes de comenzar a diseñar. Se revisan otras soluciones del mercado, los problemas que puedan presentarse, etc.
- **Diseño Funcional:** En esta etapa se define la base de todo lo que se va a desarrollar. Se identifican los involucrados en los procesos de la aplicación, se definen los procesos que llevarán a cabo y se establecen todas las restricciones y flujos que se requieran. Este es el fuerte del diseño, todo debe estar claro y nada puede quedar al azar, siempre pensando todo desde el punto de vista funcional. No involucrarse demasiado en los procesos informáticos. Estás serán las instrucciones/restricciones para darle vida a una interfaz.
- **Diseño:** En esta etapa, con todo el funcionamiento claro y con ideas de cómo debe presentarse cada vista de la interfaz, se debe comenzar a diseñar la UI con el diseñador.
- **Requisitos:** Al finalizar esta etapa se espera un documento entregable para el área de desarrollo, un compendio de todo lo anterior, que incluye referencias al prototipo terminado y sus vistas. Se debe detallar todo lo que no se pueda desprender del mismo prototipo o de los diagramas de procesos. Es un documento al detalle.

Por otra parte, si detallamos cómo se comportan las fases de *Construcción*, *Validación y Pruebas*, *Despliegue/Mantenimiento* podemos decir que se aplican técnicas que comúnmente se utilizan en metodologías ágiles, específicamente en *SCRUM*[12], como lo son; *product backlog*, *sprints*, *entre otros*. Sin embargo, podemos mencionar que los requisitos documentados previamente, en esta fase usualmente se traducen a *historias de usuario*, tal como se realizan en una metodología ágil.

En la Figura 2.6 podemos visualizar las fases de una metodología híbrida a lo largo de todo el proceso de desarrollo de software:

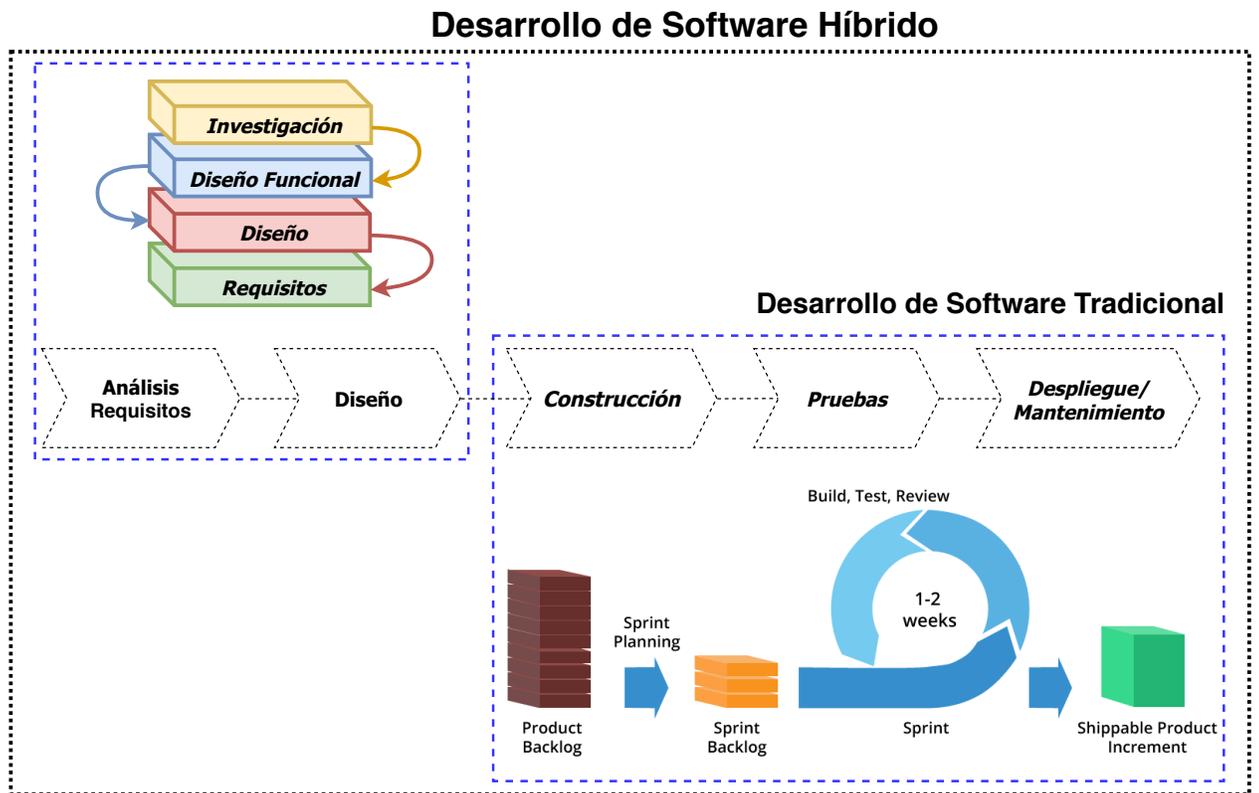


FIGURA 2.6: Diagrama de las etapas de la Metodología Híbrida de Haulmer Inc.

Finalmente cabe mencionar que se utilizó esta metodología principalmente porque la empresa Haulmer Inc. usa una metodología híbrida para el desarrollo de sus proyectos de software por lo que es necesario adaptarse a la manera de trabajo de la empresa, ya que finalmente se está desarrollando un producto para ellos.

## 2.5. Metodología de Evaluación basada en Experimentos

Para el desarrollo del proyecto usaremos una metodología de evaluación basada en experimentos controlados. Esta metodología de evaluación proporciona una manera sistemática, disciplinada, cuantificable y controlada de evaluar actividades desarrolladas por humanos [13]. La experimentación pretende emparejar con hechos las suposiciones y especulaciones que surgen durante la construcción y mantenimiento del software [14].

Según el artículo “Experimentación en Ingeniería de Software”[14], los experimentos son apropiados para investigar diferentes aspectos, incluidos:

- Confirmar teorías (probar teorías existentes).
- Confirmar sabiduría convencional (creencias de la gente).
- Explorar relaciones (probar que existe cierta relación).
- Validar métricas.

También nos señala que existe un proceso experimental, el cual se describe de la siguiente manera:

**Definición:** Responde a las preguntas ¿Qué se estudia?, ¿Cuál es la intención?, ¿Cuál es el efecto estudiado?, ¿Dónde se lleva a cabo el estudio?.

**Planificación:** De la definición se realiza una selección del contexto, de ser necesario se formula una hipótesis, posteriormente se seleccionan las variables y sujetos. Luego se realiza un diseño del experimento y su respectiva instrumentación. Finalmente se evalúa su validez.

**Operación:** En esta etapa se prepara, ejecuta y validan los datos del diseño del experimento creado en la fase previa. Finalmente se obtiene los datos experimentales.

**Análisis e Interpretación:** De los datos experimentales se realizan estadísticos descriptivos, reducción del conjunto de datos y un contraste de la hipótesis para dar lugar a las conclusiones

**Presentación y Difusión:** De las conclusiones se realiza una recopilación de la información, generalmente a través de un informe escrito.

Para evaluar el sistema usaremos un segmento de usuarios conocidos. Para este caso está formado por uno o varios usuarios que la empresa conoce e identifica debido a que son usuarios que la empresa maneja. Estos usuarios navegan a través de la plataforma de Reseller, realizando distintas pruebas de usabilidad y verificando las funcionalidades más relevantes del sistema.

Se analiza el flujo que realiza un cliente cuando quiera comprar un dominio, permitiendo evaluar la operación y consistencia de cada microservicio que interactúa con el cliente para efectuar la acción de compra de dominio. En la Figura 2.7, se muestra el flujo que debería seguir el sistema cuando un usuario compró un dominio:

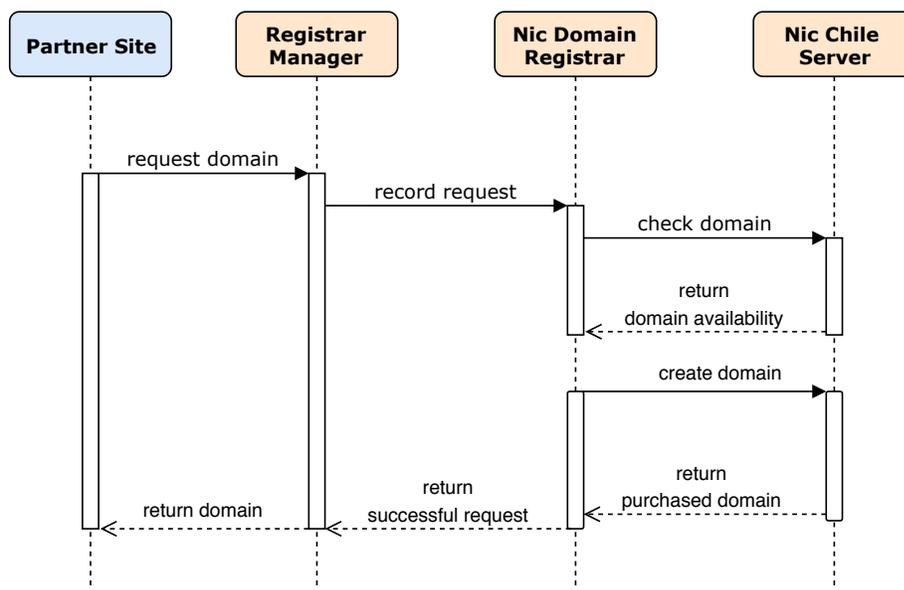


FIGURA 2.7: Diagrama de secuencia del proceso de compra de un nombre de dominio en particular.

## 2.6. Herramientas de Desarrollo

Para la implementación de la base de datos se utiliza MySQL como sistema de gestión de base de datos del software. Sin embargo, para obtener un manejo más intuitivo y práctico, se utiliza la herramienta MySQL Workbench en su versión 6.3 de 64 bits.

**MySQL:** El sistema de gestión de base de datos relacionales está desarrollado bajo Licencia Pública General (GPL<sup>3</sup>) o licencia comercial distribuida por Oracle Corporation. Actualmente es considerada como la base de datos de código abierto más popular del mundo, posicionándose en segundo lugar según el ranking que realiza *db-engines* (ver Figura 2.8).

Rank			DBMS	Database Model	Score		
Oct 2018	Sep 2018	Oct 2017			Oct 2018	Sep 2018	Oct 2017
1.	1.	1.	Oracle +	Relational DBMS	1319.27	+10.15	-29.54
2.	2.	2.	MySQL +	Relational DBMS	1178.12	-2.36	-120.71
3.	3.	3.	Microsoft SQL Server +	Relational DBMS	1058.33	+7.05	-151.99
4.	4.	4.	PostgreSQL +	Relational DBMS	419.39	+12.97	+46.12
5.	5.	5.	MongoDB +	Document store	363.19	+4.39	+33.79
6.	6.	6.	DB2 +	Relational DBMS	179.69	-1.38	-14.90
7.	↑8.	↑9.	Redis +	Key-value store	145.29	+4.35	+23.24
8.	↓7.	↑10.	Elasticsearch +	Search engine	142.33	-0.28	+22.09
9.	9.	↓7.	Microsoft Access	Relational DBMS	136.80	+3.41	+7.35
10.	10.	↓8.	Cassandra +	Wide column store	123.39	+3.83	-1.40

FIGURA 2.8: Ranking de popularidad de bases de datos [15].

MySQL se ha convertido en la principal opción de base de datos para aplicaciones basadas en la web debido a su facilidad de uso, confiabilidad y rendimiento comprobado [16].

**MySQL Workbench:** Herramienta visual unificada para arquitectos de bases de datos, desarrolladores y DBAs. MySQL Workbench proporciona modelado de datos, desarrollo SQL y herramientas integrales de administración para la configuración del servidor, administración de usuarios y respaldo, entre otras funcionalidades.

<sup>3</sup>GPL: General Public License.

Como herramienta para la implementación del sistema de Partner (Reseller) y la conexión con el servidor de Nic Chile, se utiliza el lenguaje de programación Python en su versión 3.6.4 para el conector con Nic Chile. Para el caso de la plataforma de Partner se utiliza el lenguaje PHP en su versión 7.1.7, el cual se utiliza en conjunto a Laravel en su versión 5.5.2.

**Python:** De acuerdo a una extracción de información del libro “*Python language reference manual*” [17] se pueden sintetizar algunas características relevantes de este lenguaje, como por ejemplo que Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible. Es un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional.

Es un lenguaje interpretado, usa tipado dinámico y es multiplataforma. Posee una licencia de código abierto, denominada Python Software Foundation License, que es compatible con la Licencia pública general de GNU a partir de la versión 2.1.1.

**PHP:** Según una extracción del libro “*PHP Manual: Volume 1*”[18], este lenguaje de programación es de código que es ejecutado en el servidor, el cual genera HTML enviándolo al cliente. Este lenguaje es utilizado para el desarrollo de aplicaciones web y tiene la particularidad de incorporarse directamente en un documento HTML en lugar de llamar a un archivo externo para procesar los datos.

Fue creado originalmente por Rasmus Lerdorf en el año 1995. Actualmente el lenguaje sigue siendo desarrollado con nuevas funciones por el grupo PHP. Entre las principales características encontramos:

- El código fuente es invisible para el navegador y sobre todo para el cliente, ya que este lenguaje se ejecuta en el servidor.
- Contiene una amplia documentación en su sitio web.
- Tiene capacidad de conexión con la mayoría de los motores de base de datos.
- Permite realizar programación orientada a objetos.

**Laravel:** Con la ayuda del libro *Laravel 5 essentials*[19] algunos extractos de este y así dar a conocer algunas características principales de Laravel. En primera instancia cabe mencionar que es uno de los frameworks de código abierto más fáciles de asimilar para PHP. Fue creado en 2011 y tiene una gran influencia de frameworks como Ruby on Rails, Sinatra y ASP .NET MVC.

El objetivo de Laravel es ser un framework que permita el uso de una sintaxis refinada y expresiva para crear código de forma sencilla, evitando el código espagueti y permitiendo multitud de funcionalidades. Aprovecha todo lo bueno de otros frameworks y utiliza las características de las últimas versiones de PHP.

La mayor parte de su estructura está formada por dependencias, especialmente de Symfony, lo que implica que el desarrollo de Laravel dependa también del desarrollo de sus dependencias.

Para desarrollar la plataforma de Partner y poder mostrar los resultados de las peticiones, es necesario implementar una arquitectura MVC con la ayuda del framework Laravel.

**Modelo:** El ORM (*Object-Relational mapping*) Eloquent de Laravel implementa *ActiveRecord* para trabajar con base de datos. Cada tabla de base de datos tiene un Modelo correspondiente que se utiliza para interactuar con esa tabla. Los modelos permiten consultar datos en sus tablas, así como insertar nuevos registros.

**Vista:** Blade es un motor de creación de plantillas simple de usar, pero potente. A diferencia de otros populares motores de plantillas de PHP, Blade permite usar código PHP en sus vistas, esto hace que el código sea más limpio. De hecho, todas las vistas de Blade se compilan en código PHP y se almacenan en la memoria caché hasta que se modifiquen. Los archivos de Blade View se almacenan en el directorio propio para las vistas.

**Controlador:** Los controladores son un mecanismo que nos permite agrupar la lógica de peticiones HTTP relacionadas y de esta forma organizar mejor nuestro código. Laravel maneja los controladores en un directorio.

## 3. Desarrollo de la Propuesta de Solución

---

En esta sección se describe específicamente cómo se desarrolla la propuesta de solución, el diseño arquitectónico del sistema y cómo se aplica la metodología de desarrollo híbrido durante el proceso. También se mencionan las tecnologías descritas en la Sección 2.6 para el desarrollo de la propuesta.

### 3.1. Concepción del Proyecto

Durante este proceso, que corresponde a la *Fase de Investigación* de la metodología de desarrollo híbrida, se realiza un estudio previo antes de comenzar a diseñar. Se revisaron otras soluciones presentes en el mercado y también los problemas que puedan presentarse durante el desarrollo. Luego, para entender de mejor manera la información, se realizaron varias entrevistas con los involucrados (stakeholders), con esto se pudo resolver el flujo que realiza un cliente cuando compra un dominio.

En la Figura 3.1, se ilustra el proceso que realiza un cliente para solicitar un dominio en la plataforma *Partner Site*:

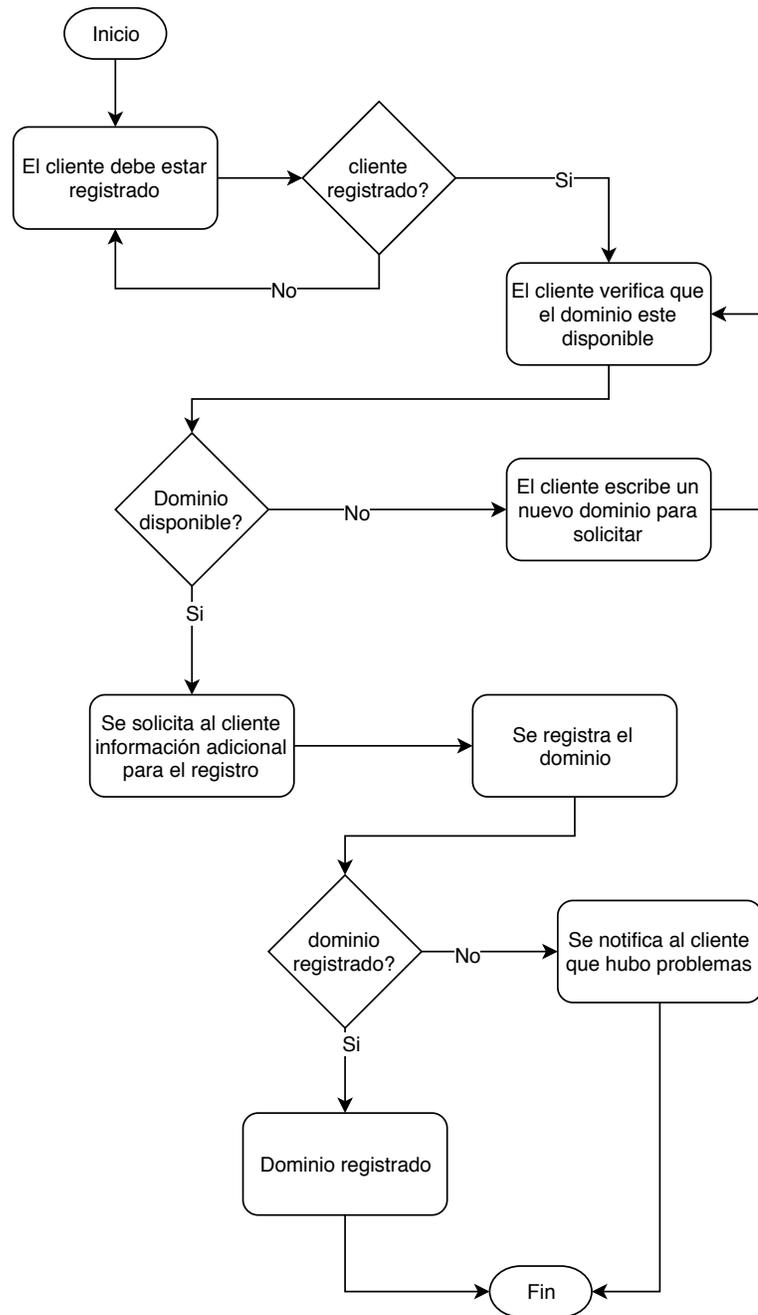


FIGURA 3.1: Diagrama de actividad del flujo de compra de un dominio.

Por otra parte, el desarrollo del proyecto se separa en dos hitos, los cuales se detallan a continuación:

**Definición Hito Uno:** En esta etapa se desarrolla el microservicio que se conecta con el servidor de NIC Chile, el cual interactúa directamente cuando se trata de resolver peticiones que involucren venta de dominios con TLD .CL, ya que las interacciones se realizan bajo el protocolo EPP.

**Definición Hito Dos:** En esta etapa se implementa la capa lógica para la gestión de Reseller y Clientes que interactúan en el sistema, gestionando los dominios solicitados. Adicionalmente se agrega una capa de presentación con la finalidad de mostrar una visualización más cómoda para que el cliente pueda realizar sus peticiones y por otra parte, poder administrar la información de los usuarios que interactúan en el sistema. Por último cabe mencionar que a la plataforma de gestión de Resellers y clientes se le llamará “*Partner Site*”.

## 3.2. Diseño Arquitectónico

En esta sección se puntualiza el diseño arquitectónico que solventa los requisitos del sistema, tanto las condiciones y políticas de la empresa Haulmer Inc. como también las de NIC Chile para una correcta integración entre ambas entidades.

### 3.2.1. Arquitectura Física

La arquitectura física que se diseña particularmente para este sistema contempla como parte de esta, un extracto de la arquitectura física que la empresa tiene definida para incluir y manejar nuevos microservicios a su arquitectura general de servicios. En la Figura 3.2, se ilustra un diagrama que representa dicha arquitectura:

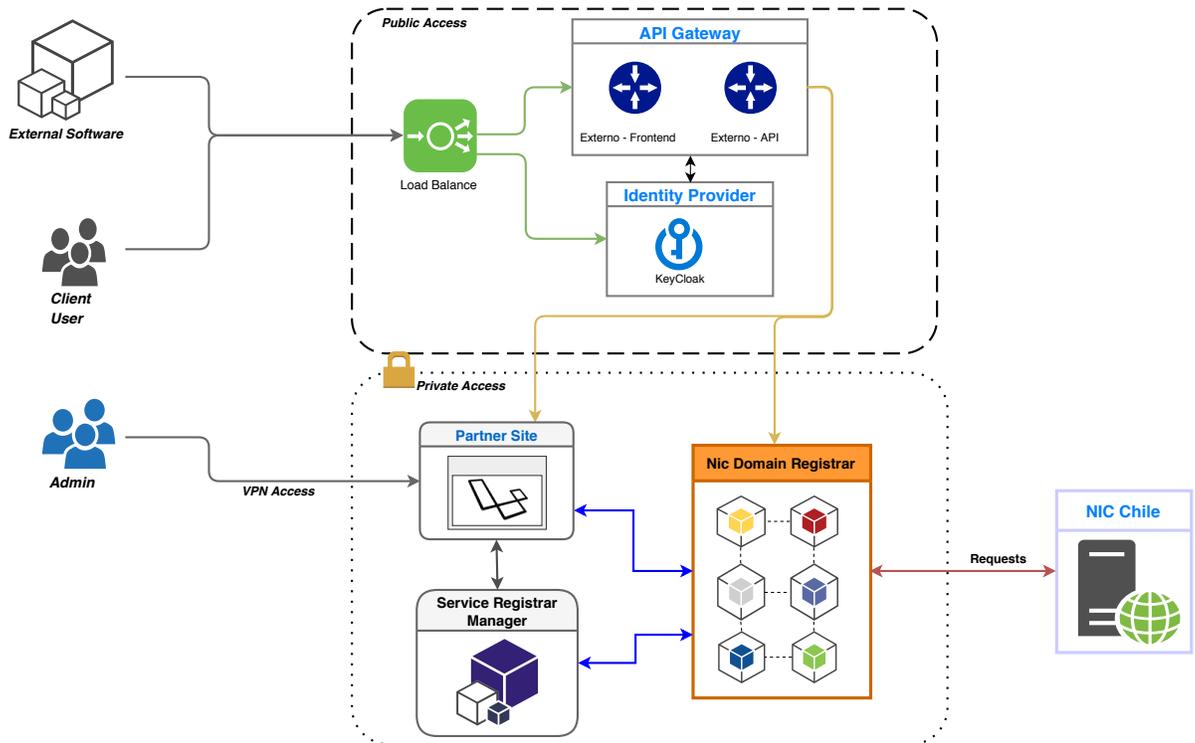


FIGURA 3.2: Arquitectura Física del Sistema.

Según el diagrama anterior, es importante mencionar que la parte superior corresponde a la arquitectura que Haulmer utiliza para todos sus microservicios que se integran y comunican con sus otros servicios. Podemos mencionar que tienen un acceso público restringido mediante tokens personalizados por usuario. Particularmente estos accesos son controlados por una API Gateway y el software Keycloak, quienes conjuntamente validan la identificación y, según los permisos que tenga el usuario, lo direcciona a los microservicios correspondientes.

La parte inferior del diagrama corresponde a la arquitectura física del sistema de reseller y también la conexión con el servidor de NIC Chile. Cabe mencionar que esta parte de la arquitectura es de acceso privado y que es regulada por los sistemas descritos en el párrafo anterior. Sin embargo, tal como se muestra en el diagrama, los usuarios administradores de la empresa pueden tener un acceso directo a estos servicios mediante una VPN con sus respectivas credenciales.

La arquitectura inferior del diagrama está compuesta por tres microservicios principales, los cuales se definen de la siguiente manera:

**Partner Site:** Este microservicio es el encargado de la gestión de Resellers y sus productos. Para este caso, el único producto disponible consiste en los nombres de dominio. Adicionalmente, mantiene registro de los clientes que adquieren este producto, así como también las transacciones que ocurren cuando un cliente compra un producto a un reseller en particular. En resumen, es la plataforma de venta que tienen los reseller para ofrecer nombres de dominios a sus clientes, y que proporciona las funcionalidades necesarias para realizar y validar el proceso de compra.

**Service Registrar Manager:** Este microservicio tiene como objetivo capturar las peticiones de los clientes referentes a la compra de un dominio, específicamente las peticiones de tipo *Post*, como por ejemplo *registrar un dominio*, *registrar un contacto para un dominio*, *renovar el periodo de vencimiento de un dominio* y *eliminar un dominio*. Este microservicio funciona como orquestador entre los microservicios de *Partner Site* y *Service Registrar Manager*.

**Nic Domain Registrar:** Este microservicio se encarga de realizar la conexión con el servidor de NIC Chile, ejecutando los diferentes comandos bajo el protocolo EPP, para llevar a cabo las peticiones que los clientes demandan mediante el microservicio antes señalado. Sin embargo, también mantiene una comunicación con el microservicio *Partner Site* para ejecutar las peticiones que requieren de una respuesta inmediata, como por ejemplo las peticiones de verificación de la disponibilidad de un dominio o el servicio “*WhoIs*” presente en todos los proveedores de nombre de dominios para obtener información relevante de un dominio en particular.

### 3.2.2. Arquitectura Lógica

La Arquitectura Lógica que se diseña para el sistema contempla una capa de presentación y los diferentes microservicios que involucran las funcionalidades para la gestión de Reseller, clientes y dominios. Además, hay que considerar el proceso de venta de dominios y la interacción entre los actores antes mencionados (Resellers y clientes) para efectuar esta acción. Es por esto, que en la Figura 3.3 se ilustra un diseño de la arquitectura lógica que atiende las necesidades del sistema:

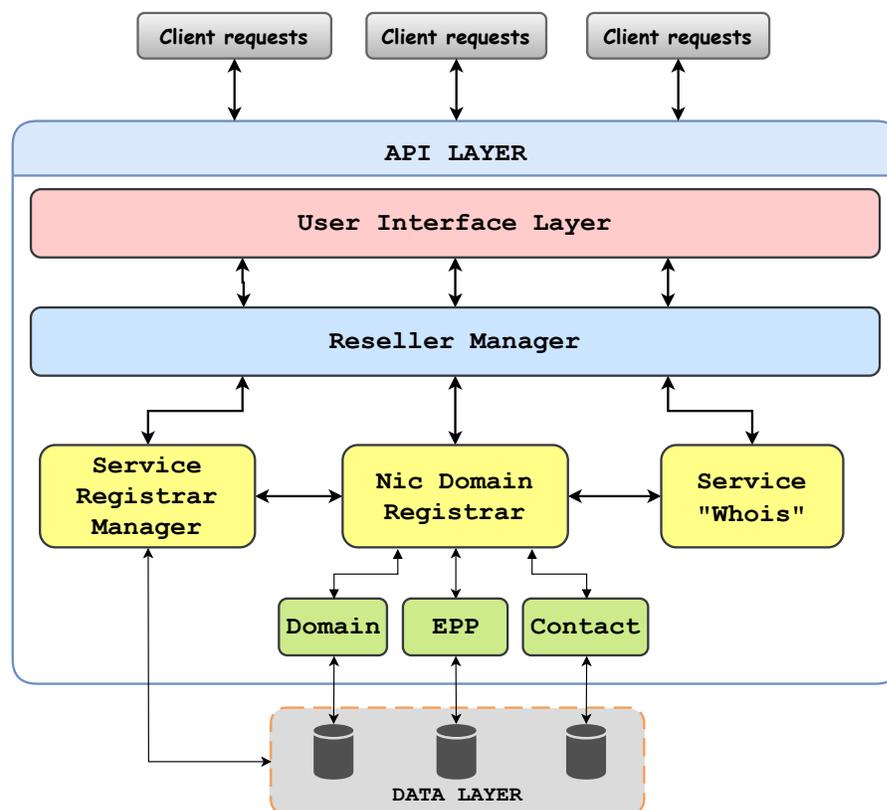


FIGURA 3.3: Arquitectura Lógica del Sistema.

**Client Requests:** Representan las peticiones que hacen los clientes que tienen relación sobre acciones de venta y gestión de dominios.

**User Interface Layer:** Simboliza la capa de presentación de la plataforma de Reseller, la cual otorga a los clientes una visualización más intuitiva para la gestión y compra de dominios.

**Reseller Manager:** Este microservicio es la capa lógica de la plataforma de reseller, la cual se encarga de la gestión y venta de los dominios para que los clientes puedan tener un servicio eficiente. Sin embargo, también se encarga de la gestión de usuarios, ya sea reseller o clientes que se registran en el sistema.

**Service Registrar Manager:** Este microservicio cumple la funcionalidad de realizar las peticiones de los clientes, específicamente las que tienen que ver con el registro y gestión de los dominios. También cumple la función de orquestador entre el microservicio de *Reseller Manager* y *Nic Domain Registrar*.

**NIC Domain Registrar:** Este microservicio cumple la función de conectar con el proveedor de dominios NIC Chile. Particularmente se encarga de realizar peticiones a sus servidores mediante el protocolo EPP para realizar el registro y gestión de dominios solicitados por los clientes. Adicionalmente, se detalla los nano-servicios principales, los cuales se explican a continuación:

- **Domain:** Este nano-servicio se encarga de gestionar los comandos bajo el protocolo EPP para las peticiones que se refieren a las acciones de dominios, ya sea registrar, renovar, verificar disponibilidad y eliminar.
- **EPP:** Este nano-servicio se encarga de realizar la comunicación con el servidor de NIC Chile, específicamente la autenticación con el servidor.
- **Contact:** Este nano-servicio gestiona todas las peticiones que se refieren a las acciones de contactos, ya sea crear o editar un contacto.

**Service “Whois”:** Este microservicio tiene la funcionalidad de dado un nombre de dominio, entregar toda la información respecto a este. Este servicio está implementado en todos los proveedores de registro de dominios.

**Data Layer:** Corresponde a la capa de datos del sistema, donde se aloja la base de datos que mantiene los registros de todas las entidades que involucran el negocio del sistema.

### 3.3. Implementación de la Metodología Híbrida

En esta sección se describe la implementación de la metodología híbrida, detallando cada etapa del proceso de desarrollo del software que se ilustra en la Figura 2.6 de la Sección 2.4. Para el desarrollo del proyecto, como se explica en la Sección 3.1, se divide en dos hitos. Por cada fase de la metodología híbrida se detalla que se hizo en cada hito respectivamente.

#### 3.3.1. Investigación

**Hito Uno:** En la fase de investigación durante el desarrollo del hito uno, se realizan indagaciones sobre el protocolo EPP y cómo utilizarlo para tener comunicación con el servidor de NIC Chile. Para realizar este proceso, se utiliza como referencia la documentación que entrega NIC Chile a sus socios, con la finalidad que entiendan cómo ellos trabajan el protocolo EPP.

**Hito Dos:** Por otra parte, en el hito dos durante la fase de investigación, se realiza un estudio de alguna solución que ofrece actualmente el mercado.

En esta oportunidad, en conjunto con la empresa Haulmer, se toma la decisión de homologar algunos procesos de la página web *Reseller Club*. Cabe destacar que sólo se toma como referencia los procesos genéricos que involucran el negocio.

#### 3.3.2. Diseño Funcional

**Hito Uno:** En esta fase durante el desarrollo de este hito, se realizan diagramas de procesos de los llamados de API's más importantes del microservicio que se encarga de la comunicación con el servidor de NIC Chile. Entre las llamadas más importantes podemos mencionar:

- Registrar un dominio.
- Renovar el periodo de expiración de un dominio.
- Borrar un dominio.
- Obtener información de un dominio.
- Registrar un contacto.

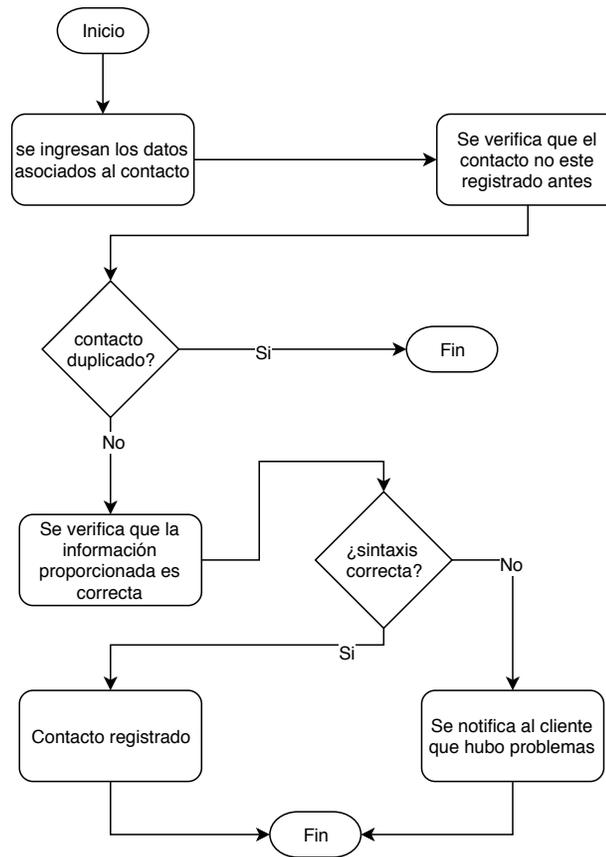


FIGURA 3.4: Diagrama de actividad de registro de dominio.

En la Figura 3.4 se describe el proceso de registro de un dominio en particular, el cual debe completar los siguientes pasos para llevarse a cabo:

1. Ingresar datos asociado al nombre de dominio: años de vigencia, contactos administrativo, técnico y de facturación.
2. Verificar la disponibilidad del dominio.
3. Si el dominio está disponible, se procede a verificar la información asociada y la sintaxis del comando EPP.
4. Si no existen problemas con lo anterior, se procede a registrar el dominio, en caso contrario se informa que la operación no se pudo concretar de manera correcta.

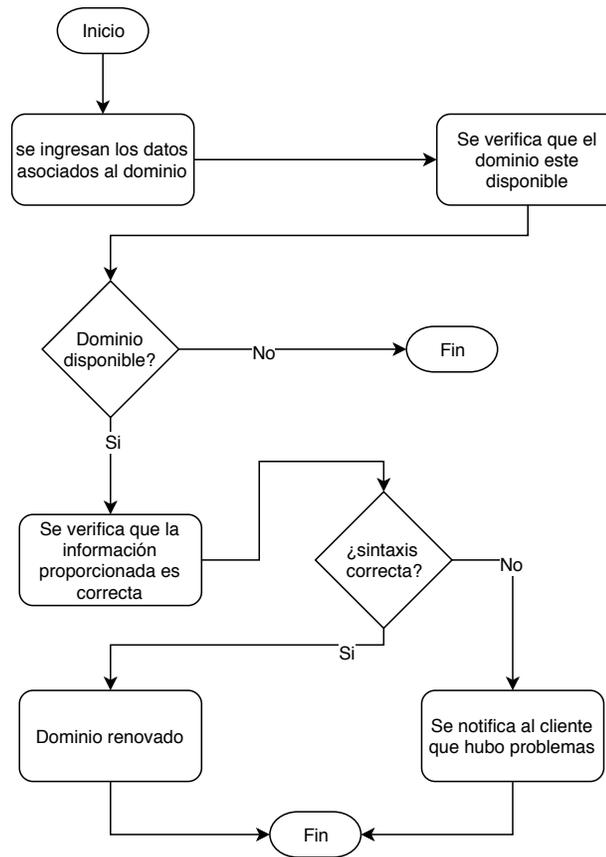


FIGURA 3.5: Diagrama de actividad para renovar vigencia de un dominio.

En la Figura 3.5 se describe el proceso de renovación del periodo que tiene de vigencia un dominio. Es importante seguir los siguientes pasos para completar el proceso:

1. Ingresar los datos asociados. Es importante ingresar el nombre de dominio que se quiere renovar así como también su fecha de expiración y el periodo que se quiere renovar.
2. Se verifica que los datos antes mencionados estén correctos.
3. Si todo esta bien, se procede a renovar la vigencia del dominio por el periodo indicado.

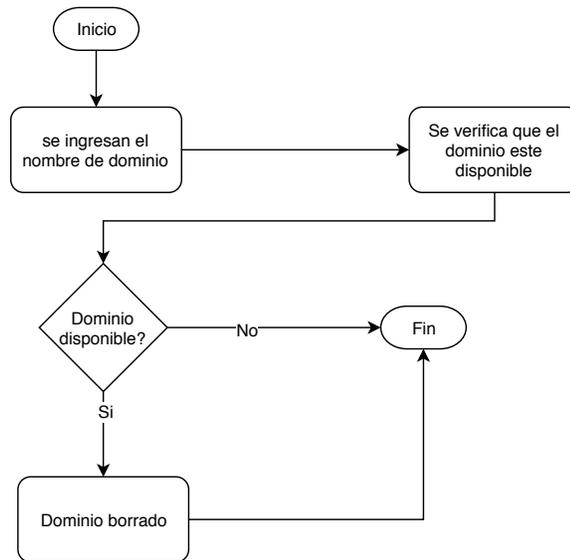


FIGURA 3.6: Diagrama de actividad para borrar un dominio.

En la Figura 3.6 se detalla el proceso de borrar un dominio en particular, cuyo proceso consta de los siguientes pasos:

1. Verifica que el dominio se encuentre registrado.
2. Si se encuentra registrado, el dominio se procede a eliminar.

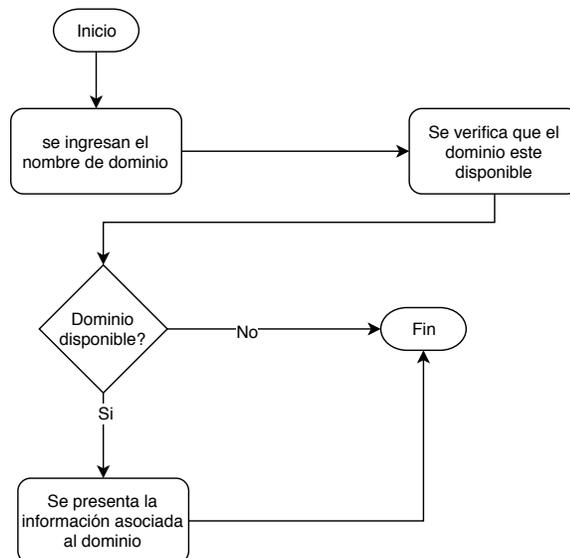


FIGURA 3.7: Diagrama de actividad para obtener información de un dominio.

En la Figura 3.7 se ilustra el proceso de obtención de información de un dominio en particular. Para llevar a cabo este proceso se realizan los siguientes pasos:

1. Se ingresa el nombre de dominio que se quiere consultar su información.
2. Se procede a verificar si el dominio se encuentra registrado.
3. Si el paso anterior es válido, se entrega la información sobre el dominio consultado.

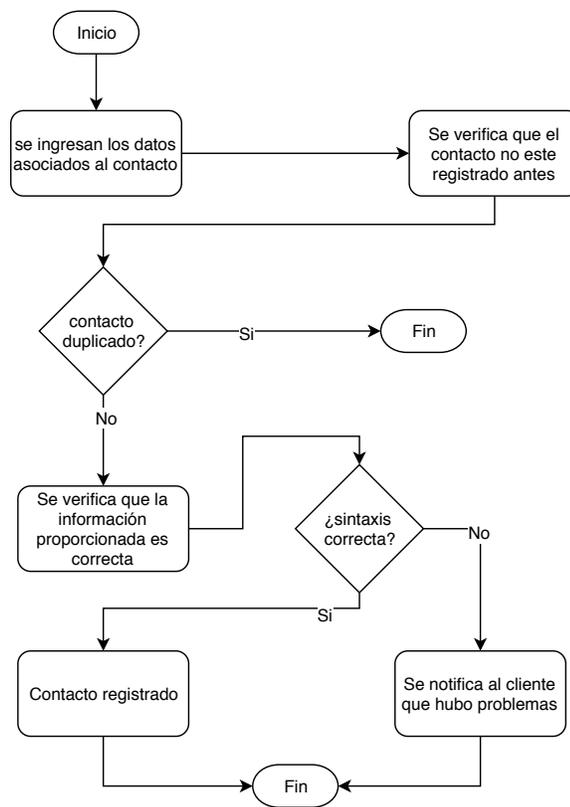


FIGURA 3.8: Diagrama de actividad para registrar un contacto.

En la Figura 3.8 se señala el proceso de registro de un contacto para asociarlo a un dominio. Para dicho proceso se realizan los siguientes pasos:

1. Se ingresa la información de un contacto, como por ejemplo el nombre, email, dirección, número de teléfono, etc.
2. Luego se verifica que el contacto no esté registrado previamente.

3. Posteriormente se verifica que la sintaxis esté correcta.
4. Finalmente, si todo lo anterior está correcto se procede a registrar el contacto.

**Hito Dos:** Por otra parte, en el hito dos durante el desarrollo de la fase de diseño funcional, se consideran algunos flujos relevantes que se realizan en el sistema en el cual se diseñó diagramas BPMN (**Business Process Model and Notation**, en español, Modelo y Notación de Procesos de Negocio). Entre los procesos más relevantes encontramos los siguientes:

- Acceso por parte de los usuarios a la plataforma de partner mediante credenciales.
- Registro de usuarios.
- Listar clientes propios de un partner.
- Proceso de compra de un nombre de dominio.

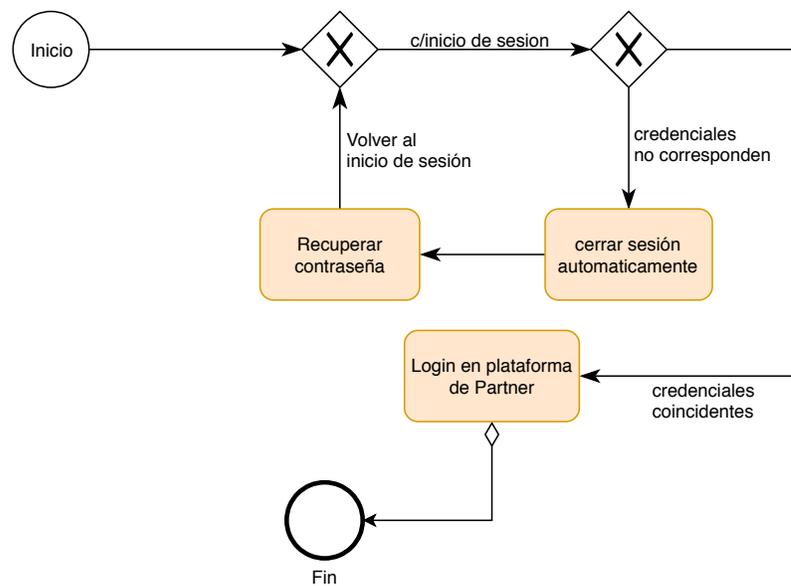


FIGURA 3.9: Diagrama BPMN para acceder a la plataforma Partner.

En la Figura 3.9 se muestra un diagrama BPMN del proceso de acceso a la plataforma de partner por parte de los usuarios, ya sea Reseller o Clientes, respectivamente. Para acceder a la plataforma se debe seguir el siguiente protocolo:

1. Se accede a la plataforma con las respectivas credenciales pertenecientes a un usuario en particular.
2. Si las credenciales no corresponden, no se permite acceder a la plataforma, proporcionando un mensaje en la interfaz para que el usuario tenga conocimiento de lo ocurrido.
3. En el caso particular, que el cliente o reseller haya olvidado su contraseña de acceso, podrá recuperarla mediante el correo electrónico.
4. Si las credenciales coinciden con las proporcionadas al usuario para acceder a la plataforma, dependiendo del tipo de usuario, podrá acceder a la plataforma de venta o a la plataforma administrativa del sistema.

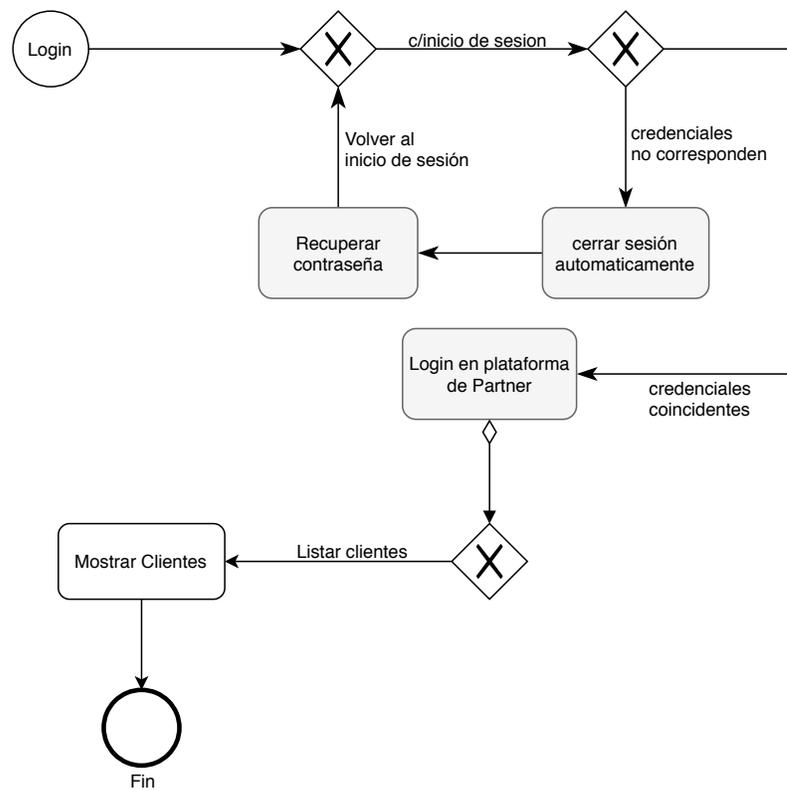


FIGURA 3.10: Diagrama BPMN para listar clientes.

En la Figura 3.10 se ilustra el proceso para mostrar los clientes pertenecientes a un partner o reseller en particular, el cual sigue el siguiente protocolo:

1. El partner debe acceder al sistema con su respectivas credenciales.
2. Luego deben ingresar a la interfaz de clientes y finalmente seleccionar en el menú de listar clientes.

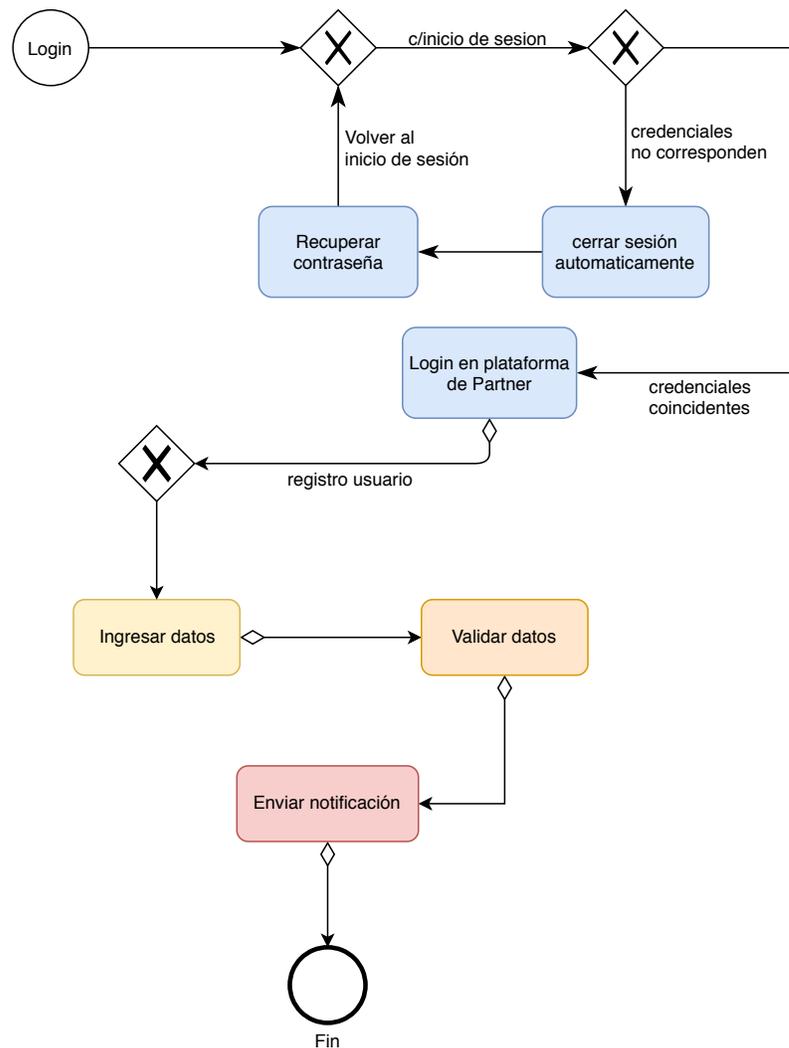


FIGURA 3.11: Diagrama BPMN para registro de usuarios.

En la Figura 3.11 se ilustra el proceso de registro de usuarios al sistema. Los usuarios registrados pueden ser de dos tipos; Reseller ó Clientes. Este proceso sigue el siguiente protocolo:

1. El usuario padre (quién registra) debe realizar el proceso de acceso a la plataforma con sus respectivas credenciales.
2. Posteriormente debe entrar en la interfaz de registro de usuario, según los permisos otorgados ya que sólo el administrador puede registrar Partners y el usuario Partner puede registrar clientes.
3. Luego se ingresan los datos solicitados para el registro del nuevo usuario.
4. Se validan los datos solicitados.
5. Si los datos están correctamente validados, se procede a enviar una notificación al nuevo usuario registrado.

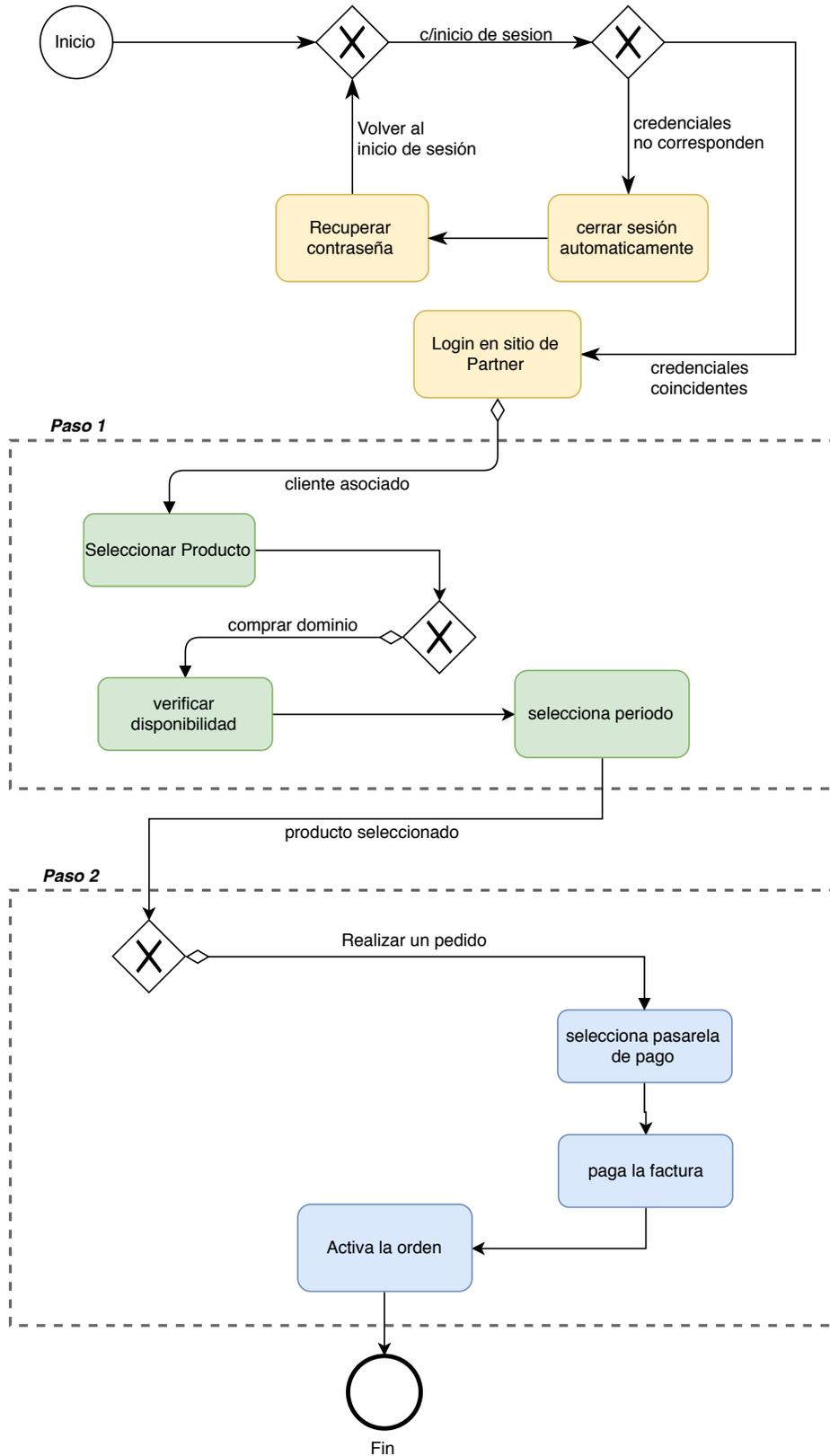


FIGURA 3.12: Diagrama BPMN para comprar un nombre de dominio.

En la Figura 3.12 se muestra el proceso de compra de un dominio, dicho proceso lo realiza un cliente registrado en el sistema. Para que este cliente pueda realizar la compra, es necesario seguir el siguiente protocolo:

1. El cliente debe acceder a la plataforma de Partner con sus respectivas credenciales.
2. El cliente debe ingresar el nombre de dominio y verificar su disponibilidad.
3. Si existe disponibilidad sobre el dominio solicitado, se procede a completar los datos solicitados para el registro de un dominio.
4. Finalmente se procede a pagar el producto y se envía una notificación al cliente sobre el éxito de la compra.

### 3.3.3. Diseño

**Hito Uno:** Durante el desarrollo de este hito en la fase de Diseño se realiza, en primera instancia, un diagrama de la arquitectura física y luego otro correspondiente a la arquitectura lógica. Estos diagramas se presentan en la Sección 3.2, en la cual se presentan los diagramas y la explicación respectivamente.

Posteriormente se diseña el diagrama de Entidad/Relación de la base de datos que se integra al sistema. En este proceso se realizan dos diseños, para lograr un mejor entendimiento, de los cuales un modelo pertenece a las entidades que participan en la conexión con el servidor de NIC Chile y el otro modelo pertenece a las entidades que participan en la plataforma de reseller.

En el desarrollo de este hito sólo explicaremos el modelo entidad/relación de la base de datos que involucra las entidades del conector con NIC Chile.

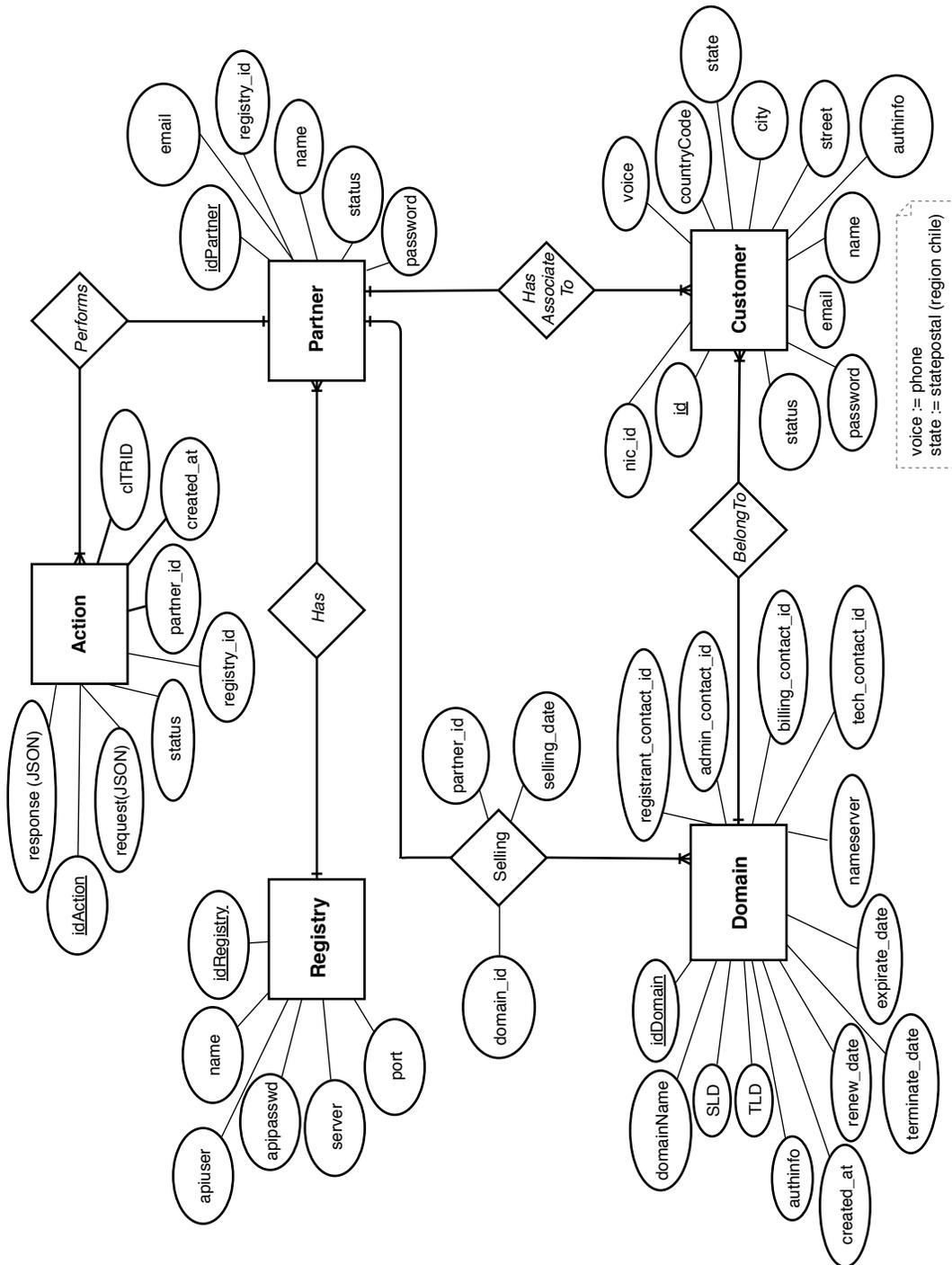


FIGURA 3.13: Diagrama entidad/relación conector NIC Chile.

En la Figura 3.13, se ilustra el diagrama Entidad/Relación que almacena los datos pertenecientes a los procesos de registro de nombres de dominio.

**Hito Dos:** Para el desarrollo de este hito durante la fase de diseño se trabaja en un bosquejo de un diagrama de entidad/relación perteneciente a la plataforma de partner y que, conjuntamente al diagrama de la Figura 3.13, trabajan para almacenar los datos pertenecientes a los procesos del negocio del sistema.

Por otra parte, también se realiza un diagrama de clases del sistema de partner, con la finalidad de explicitar los atributos y operaciones de las clases más relevantes.

Finalmente se bosquejan Mock-up de las principales interfaces para que los usuarios puedan interactuar en el sistema de partner.

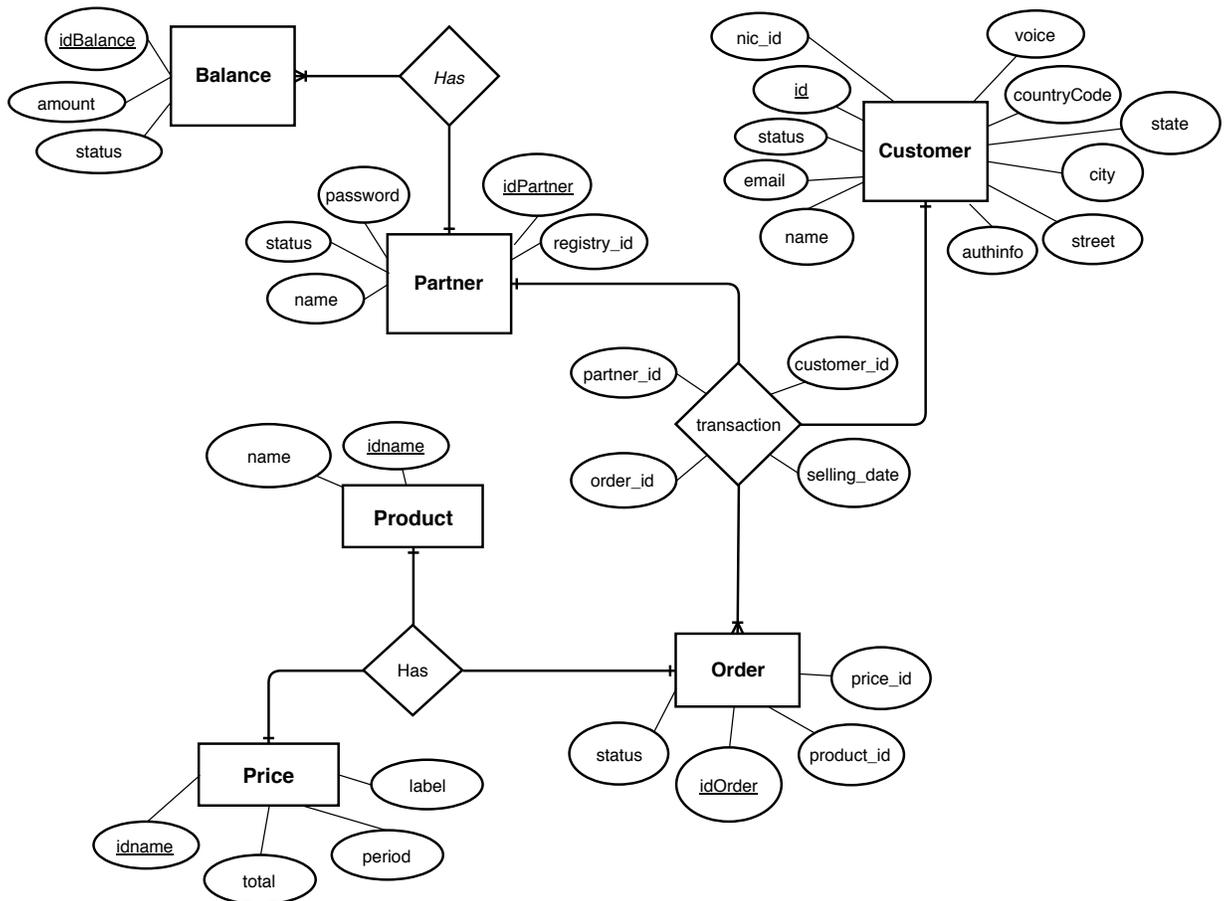


FIGURA 3.14: Diagrama entidad/relación plataforma partner.

En la Figura 3.14 se muestra las principales entidades y sus relaciones que pertenecen a la plataforma de partner. Cabe mencionar que se repiten las entidades de “*Partner*” y “*Customer*” debido a que son dos entidades principales y lo que se quiere lograr es una mejor explicación de las relaciones que tienen con las entidades que involucran el conector con NIC Chile como también la plataforma de reseller. En otras palabras, se quiere enseñar las relaciones que involucran el proceso de venta y facturación de un nombre de dominio.

Por otra parte, en la Figura 3.15 se ilustra un diagrama de clases que nos permite identificar los atributos y operaciones de las principales clases del sistema de reseller. Las clases principales que encontramos son:

- **Partner:** son todos los atributos y operaciones que tiene esta entidad.
- **Customer:** son los atributos y operaciones que un cliente tiene en el sistema.
- **Balance:** Tiene un vínculo directo con la entidad Partner, la cual pertenece al balance financiero que tiene en el sistema.
- **Money:** Esta directamente asociado a la moneda de uso por parte del usuario partner.
- **Transaction:** Son todas las transacciones de las operaciones que involucran acciones de venta de nombre de dominio que ocurren en el sistema y que pertenecen a los clientes o partners.
- **Order:** Tiene directa relación con el proceso de compra de un dominio por parte del cliente.
- **Product:** En este caso hay un único producto, nombre de dominio. Este producto es ofrecido por los partners para que adquisición por parte de los clientes.
- **Price:** Pertenece al precio de venta que tiene cada acción correspondiente a un nombre de dominio, ya sea registro o renovación de este.

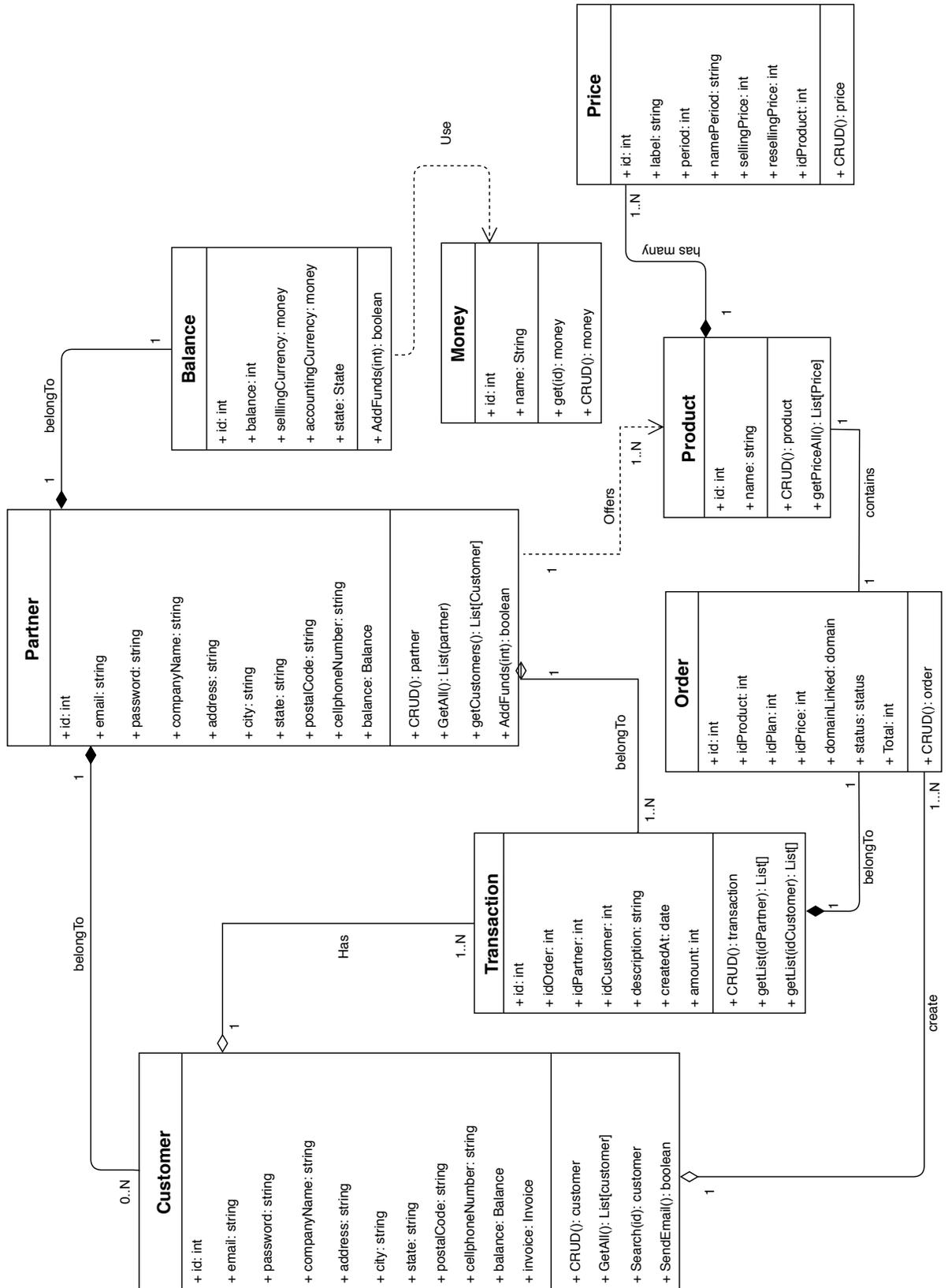


FIGURA 3.15: Diagrama de clases de la plataforma partner.

Finalmente se diseñaron mock-ups de la plataforma *Partner* con las interfaces más relevantes del sistema, principalmente operaciones administrativas del sistema, como por ejemplo:

- Registrar usuarios (Partner o Clientes).
- Editar un usuario en particular (Partner o Clientes).
- Eliminar un usuario en particular (Partner o Clientes).
- Listar usuarios (Partner o Clientes).
- Listar las transacciones para un usuario en particular.

Sin embargo, sólo se ilustra las interfaces del usuario partner.

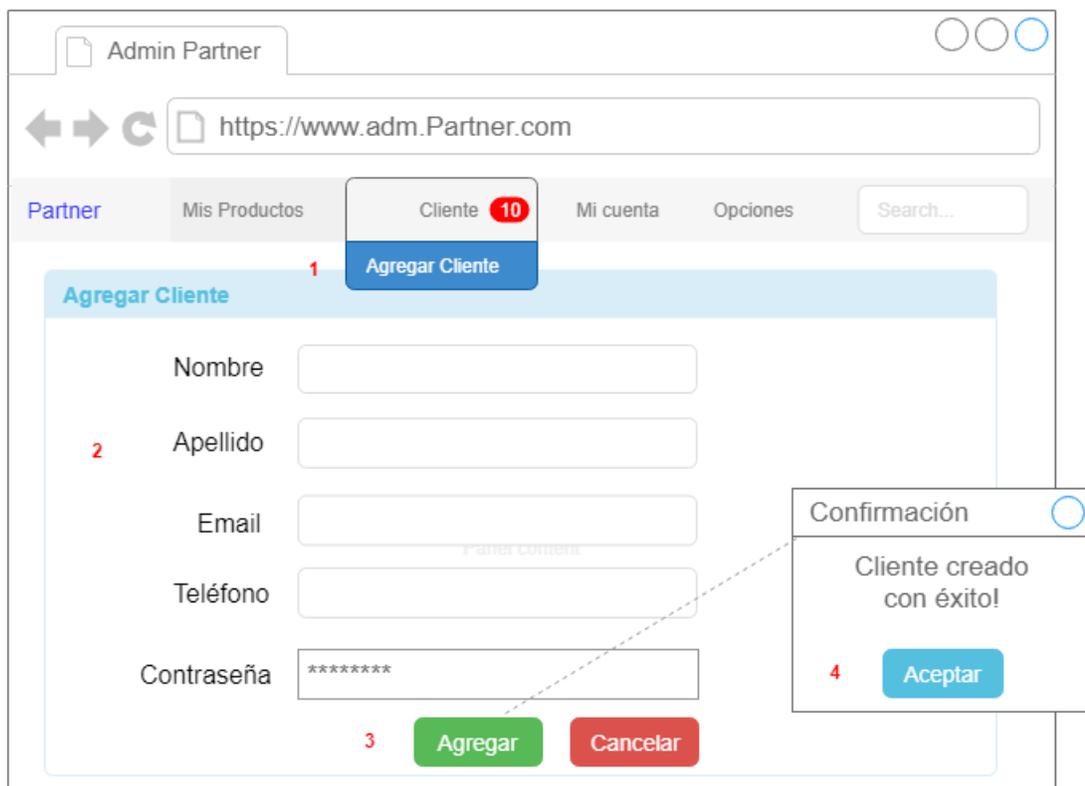


FIGURA 3.16: Mock-up crear cliente.

En la Figura 3.16 se ilustra una interfaz para que los usuarios partners puedan registrar nuevos clientes al sistema.

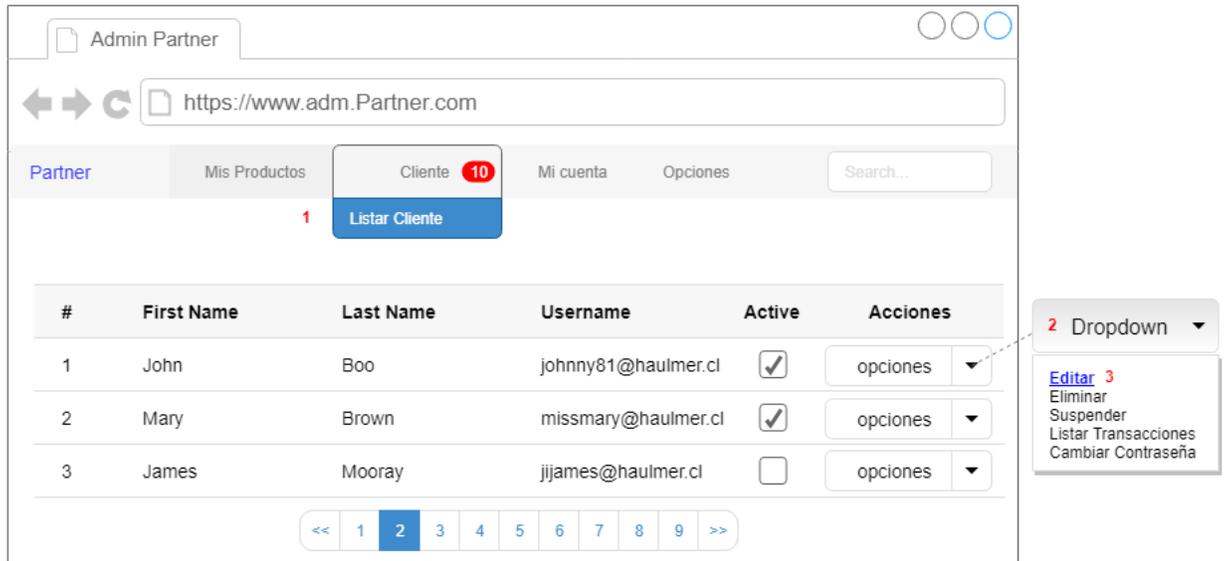


FIGURA 3.17: Mock-up editar cliente.

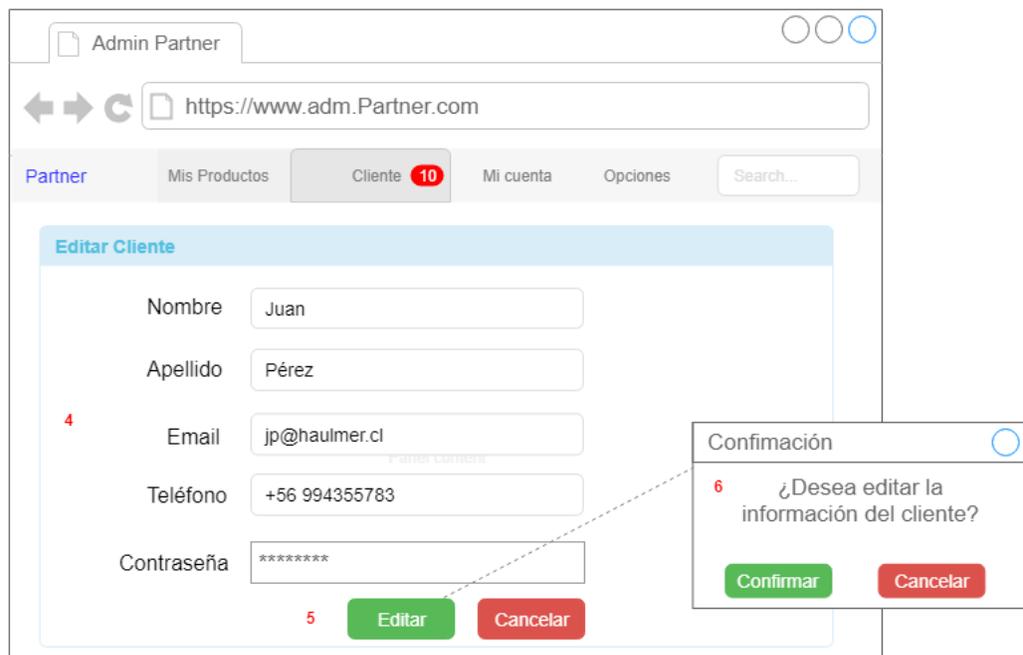


FIGURA 3.18: Mock-up editar cliente.

Las Figuras 3.17 y 3.18 representan el flujo que sigue la interfaz de usuario de un partner para editar los datos de un cliente en particular.

Para editar un usuario en particular, es necesario seguir los siguientes pasos:

1. Listar Cliente.
2. Buscar al cliente e ir al botón de acciones.
3. Seleccionar en acciones la opción “editar”.
4. Editar información que necesitemos.
5. Presionar el botón editar o cancelar.
6. Si la opción anterior fue editar, entonces se procede a confirmar la acción.

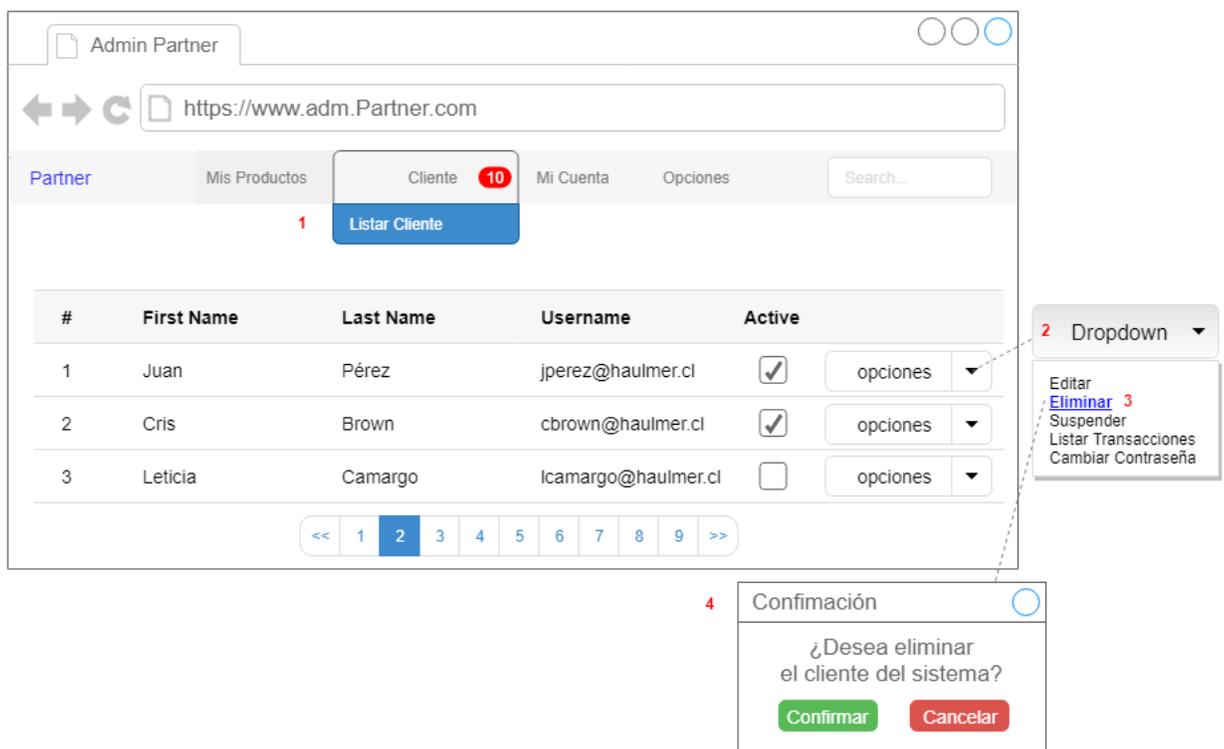


FIGURA 3.19: Mock-up eliminar cliente.

En la Figura 3.19 se ilustra la interfaz de usuario de un partner para eliminar un cliente en particular, siguiendo los siguientes pasos:

1. Listar Cliente.
2. Buscar al cliente e ir al botón de acciones.
3. Seleccionar en acciones la opción “eliminar”.
4. Se procede a confirmar o cancelar la operación.

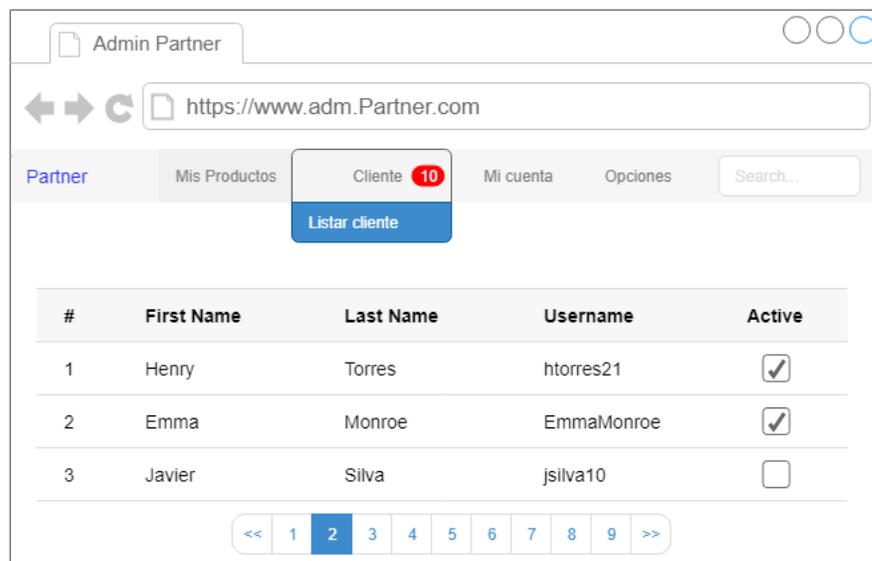


FIGURA 3.20: Mock-up listar cliente.

En la Figura 3.20 se muestra la interfaz del usuario partner para realizar la acción que involucra listar sus clientes registrados en el sistema, los cuales se muestran en una tabla en conjunto con los atributos más representativos de cada usuario.

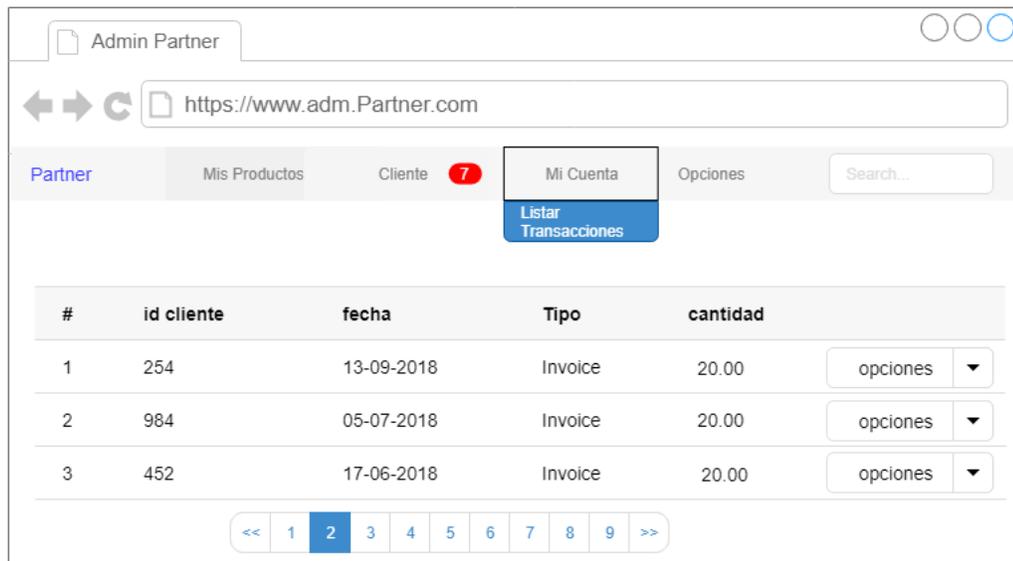


FIGURA 3.21: Mock-up listar las transacciones

En la Figura 3.21 representa la interfaz del usuario partner para listar las transacciones ocurridas en el sistema con su cuenta registrada.

### 3.3.4. Requisitos

En esta etapa, con la intención de ejemplificar, sólo se muestran los requisitos más relevantes del sistema que aportan valor y nos permiten comprender las reglas del negocio. Las funcionalidades más importantes del sistema son:

- Verificar la disponibilidad de un dominio.
- Registrar un nombre de dominio.
- Renovar el periodo de adquisición de un nombre de dominio.
- Registrar un contacto en el sistema.

<b>RU-001 - Registrar Dominio</b>
<b>Descripción:</b> El sistema debe permitir a los usuarios de tipo “cliente” poder registrar un dominio en particular.
<b>Fuente:</b> Cliente
<b>Prioridad:</b> Alta
<b>Estabilidad:</b> Intransable
<b>Fecha Actualización:</b> 06-11-2018
<b>Tipo:</b> Funcional
<b>Tipo Usuario Asociado:</b> Usuario cliente

TABLA 3.1: Requisitos de usuario para registrar un dominio.

<b>RU-002 - Verificar Dominio</b>
<b>Descripción:</b> El sistema debe permitir a los usuarios de tipo “cliente” poder verificar la disponibilidad de un dominio en particular.
<b>Fuente:</b> Cliente
<b>Prioridad:</b> Alta
<b>Estabilidad:</b> Intransable
<b>Fecha Actualización:</b> 06-11-2018
<b>Tipo:</b> Funcional
<b>Tipo Usuario Asociado:</b> Usuario cliente

TABLA 3.2: Requisitos de usuario para verificar un dominio.

<b>RU-003 - Renovar Dominio</b>
<b>Descripción:</b> El sistema debe permitir a los usuarios de tipo “cliente” poder renovar el periodo de caducidad que tiene un dominio en particular.
<b>Fuente:</b> Cliente
<b>Prioridad:</b> Alta
<b>Estabilidad:</b> Intransable
<b>Fecha Actualización:</b> 06-11-2018
<b>Tipo:</b> Funcional
<b>Tipo Usuario Asociado:</b> Usuario cliente

TABLA 3.3: Requisitos de usuario para renovar un dominio.

<b>RU-004 - Registrar Contacto</b>
<b>Descripción:</b> El sistema deber permitir registrar nuevos contactos para que se puedan usar cuando se quiere registrar un dominio.
<b>Fuente:</b> Cliente
<b>Prioridad:</b> Alta
<b>Estabilidad:</b> Intransable
<b>Fecha Actualización:</b> 06-10-2018
<b>Tipo:</b> Funcional
<b>Tipo Usuario Asociado:</b> Usuario Partner

TABLA 3.4: Requisitos de usuario para registrar un dominio.

### 3.3.5. Construcción y Pruebas

**Hito Uno:** Durante la elaboración de este hito en la fase de desarrollo del software, según los requisitos mencionados en la fase anterior, se implementa a nivel de capa lógica las diferentes API's que permiten realizar las acciones que involucran los requisitos más relevantes del sistema.

Los requisitos más relevantes son los siguientes:

1. Verificación de disponibilidad de un dominio.
2. Registro de un dominio.
3. Renovación de periodo de caducidad de un dominio.
4. Registro de un contacto.

**Requisito 1:** Para implementar el requisito propuesto en la Tabla 3.2, que corresponde a la funcionalidad de verificar la disponibilidad de un dominio, se desarrolla la siguiente API:

- `http://localhost:5000/api/v1.0/registrar/checkDomain/<domainName>`

Esta API tiene como parámetro de entrada `<domainName>` que corresponde al nombre de dominio consultado. La API responde a las peticiones de tipo `GET`, y como respuesta por parte del servidor de NIC retorna lo siguiente:

```
1 {
2   "Available": true,
3   "Code": 1000,
4   "Command": "Check Domain",
5   "Msg": "Command completed successfully",
6   "Response": "Domain name available"
7 }
```

La respuesta entregada por parte del servidor de NIC Chile la procesa el micro-servicio *Nic Domain Registrar* y la encapsula de la forma antes presentada. Esta respuesta tiene varios campos que podemos describir de la siguiente manera:

- **Available:** Puede ser *True* o *False* según corresponda la disponibilidad del dominio.
- **Code:** Si el código es “1000” significa que la petición se realizó con éxito.
- **Command:** Corresponde al comando EPP ejecutado para dicha petición.
- **Msg:** Corresponde al mensaje entregado por parte del servidor.
- **Response:** Para este caso puntual corresponde a la respuesta de la petición, que en este caso nos señala que el dominio está disponible para ser comprado.

**Requisito 2:** Para implementar el requisito propuesto en la Tabla 3.1, que corresponde a la funcionalidad de registro de un dominio, se desarrolla la siguiente API:

- `http://localhost:5000/api/v1.0/registrar/createDomain/`

Esta API tiene como parámetro de entrada un json con la siguiente estructura:

```
1 {
2   "domainName": "utalca.cl",
3   "period": "2",
4   "ns": "ns.icc.cl",
5   "registrant": "id1",
6   "admin": "id1",
7   "billing": "id3",
8   "tech": "id2",
9   "authinfo": "Nic-Haulmer-1234567"
10 }
```

La estructura de este json corresponden a los campos obligatorios que se requieren para realizar la petición de registro de un dominio al servidor de NIC Chile. Esta API responde a las peticiones de tipo *POST*, y como respuesta por parte del servidor de NIC retorna lo siguiente:

```
1 {
2   "Command": "Create Domain",
3   "Info Server": {
4     "Code": 1000,
5     "Command": "Create Domain",
6     "Msg": "Command completed successfully",
7     "Response Server": {
8       "createDate": "2018-11-03T16:15:06.196Z",
9       "exDate": "2020-11-03T16:15:06.196Z",
10      "name": "utalca-icc.cl",
11      "roid": "5951-NIC"
12    }
13  },
14  "Response": "Success"
15 }
```

En este caso en particular el microservicio nos entrega una respuesta diferente para esta petición, la cual se divide de la siguiente manera:

- **Command:** Corresponde al comando EPP ejecutado en el servidor de NIC Chile.
- **Info Server:** Corresponde a la respuesta devuelta por parte del servidor de NIC Chile, la cual nos proporciona la siguiente información: el nombre de dominio, el periodo de expiración, la fecha de registro, etc.
- **Response:** Corresponde al estado de la petición, en este caso es *"Success"* ya que la petición fue ejecutada satisfactoriamente.

**Requisito 3:** Para implementar el requisito propuesto en la Tabla 3.3, que corresponde a la funcionalidad de renovar el periodo de caducidad de un dominio, se desarrolla la siguiente API:

- `http://localhost:5000/api/v1.0/registrar/renewDomain/`

Esta API tiene como parámetro de entrada un json con la siguiente estructura:

```
1 {
2     "domainName": "utalca-icc.cl",
3     "period": "9",
4     "exDate": "2020-11-03"
5 }
```

Esta entrada cuenta con tres campos obligatorios, los cuales se describen a continuación:

- **domainName:** Corresponde al nombre de dominio que le queremos renovar su periodo de caducidad.
- **period:** Corresponde al periodo en años que queremos renovar el dominio.
- **exDate:** Corresponde a la fecha de caducidad actual que tiene el dominio.

La API responde a las peticiones de tipo *POST*, y como respuesta por parte del servidor de NIC retorna lo siguiente:

```
1 {
2     "Code": 1000,
3     "Expiration Date": "2022-11-03",
4     "Msg": "Command completed successfully",
5     "Response": "Renew Domain Success by 2 years"
6 }
```

**Requisito 4:** Para implementar el requisito propuesto en la Tabla 3.4, que corresponde a la funcionalidad de registrar un contacto en el sistema, se desarrolla la siguiente API:

- `http://localhost:5000/api/v1.0/registrar/createContact/`

Esta API tiene como parámetro de entrada un json con la siguiente estructura:

```
1 {
2     "id": "id5",
3     "name": "Luis",
4     "org": "Haulmer",
5     "street": "Chacabuco",
6     "city": "curico",
7     "sp": "VII",
8     "cc": "cl",
9     "voice": "+56.985589273",
10    "email": "luis@haulmer.com",
11    "authinfo": "Haulmer123456"
12 }
```

La estructura de este json corresponde a los campos obligatorios que se requieren para realizar la petición de registro de un contacto al servidor de NIC Chile. Esta API responde a las peticiones de tipo *POST*, y como respuesta por parte del servidor de NIC retorna lo siguiente:

```
1 {
2     "Code": 1000,
3     "Command": "Create Contact",
4     "Msg": "Command completed successfully",
5     "ResData": {
6         "CreateAt": "2018-11-03T13:49:34.268-03:00",
7         "id": "id5"
8     }
9 }
```

**Hito Dos:** Para el desarrollo de este hito en la fase de desarrollo del software se ilustran capturas de pantallas de la plataforma de reseller que cumplen con las funcionalidades descritas en los requisitos más relevantes.

**Requisito 1:** Para lograr este requisito se diseña una interfaz que permite al cliente verificar la disponibilidad de un nombre de dominio en dos pasos:



Comprar Dominio

Verificación      Completar Datos      Confirmar Compra

Verificar Disponibilidad      ① Ingrese el nombre de dominio      .cl      Verificar      ②

FIGURA 3.22: Interfaz para verificar disponibilidad dominio.

1. El cliente ingresa el nombre de dominio que quiere verificar.
2. El cliente hace click en el botón “Verificar” para realizar la acción.

Para está acción tenemos dos flujos alternativos. En la Figura 3.23 se muestra el flujo cuando el dominio esta disponible y en la Figura 3.24 cuando no está.



Comprar Dominio

Verificación      Completar Datos      Confirmar Compra

Verificar Disponibilidad      macro-facultad      .cl      Verificar

El nombre de dominio macro-facultad.cl está disponible

FIGURA 3.23: Interfaz cuando el dominio está disponible.



Comprar Dominio

Verificación      Completar Datos      Confirmar Compra

Verificar Disponibilidad      cidemich      .cl      Verificar

El nombre de dominio cidemich.cl no está disponible

FIGURA 3.24: Interfaz cuando el dominio no está disponible.

**Requisito 2:** Para lograr este requisito se desarrolla una interfaz que permite a los clientes realizar el proceso de compra de un nombre de dominio en particular. Para realizar esta acción es necesario completar los siguientes pasos:

1. Verificar disponibilidad del dominio (ver requisito anterior).
2. Ingresar información complementaria al nombre de dominio (ver Figura 3.25).
3. Confirmar la compra para generar la orden.

FIGURA 3.25: Interfaz del formulario para comprar un dominio.

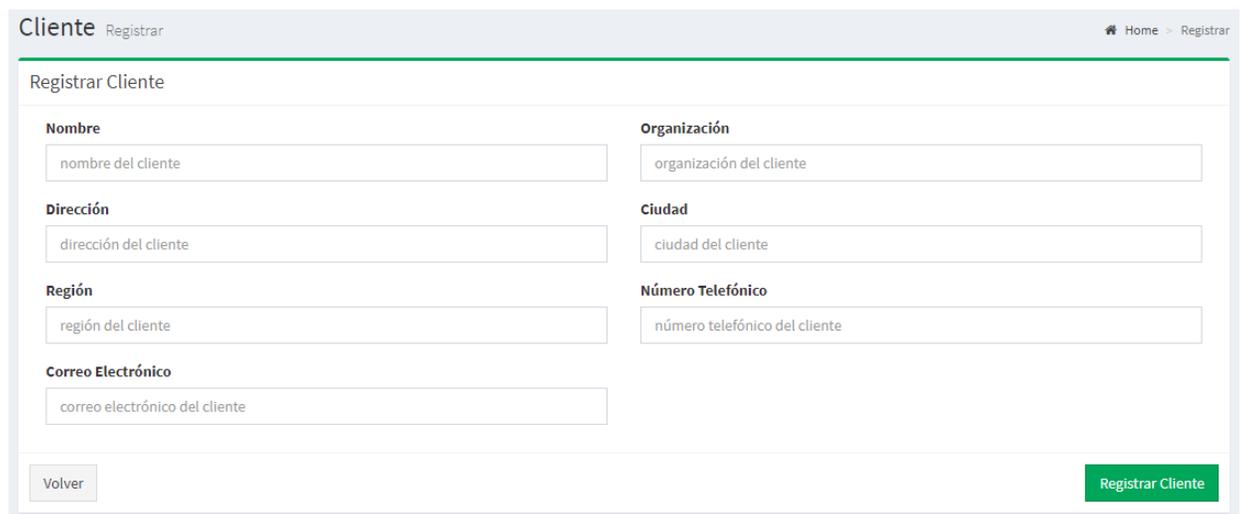
**Requisito 3:** Para lograr este requisito se implementa una interfaz (ver Figura 3.26) que permite al cliente renovar el periodo de caducidad que tiene un nombre de dominio comprado previamente.

Nombre de dominio	Registrado por	Fecha Creación	Fecha Expiración	Acciones
Bell.cl	cid-44	2018-11-13 23:55:42	2024-11-14 02:55:40	Renovar
Bic.cl	cid-44	2018-11-13 23:43:36	2020-11-14 02:43:35	Renovar

FIGURA 3.26: Interfaz del formulario para renovar un dominio.

**Requisito 4:** Para lograr este requisito se implementa una interfaz en la plataforma de reseller que permite crear nuevos contactos y registrarlos directamente en el servidor de NIC Chile para posteriormente ser utilizados en el proceso de registro de un nombre de dominio.

Para realizar esta acción es necesario completar la información solicitada en el formulario de registro de un nuevo cliente ó también se puede realizar en el proceso de registro de un dominio creando un nuevo contacto y añadirlo como parte del dominio como contacto administrativo, técnico o facturación. La Figura 3.27 se ilustra la interfaz del formulario para crear nuevos clientes (que a su vez son contactos para efectos de NIC Chile).



The image shows a web interface for registering a client. The page title is 'Cliente Registrar' with a breadcrumb 'Home > Registrar'. The main heading is 'Registrar Cliente'. The form contains several input fields:

- Nombre:** Input field with placeholder 'nombre del cliente'.
- Organización:** Input field with placeholder 'organización del cliente'.
- Dirección:** Input field with placeholder 'dirección del cliente'.
- Ciudad:** Input field with placeholder 'ciudad del cliente'.
- Región:** Input field with placeholder 'región del cliente'.
- Número Telefónico:** Input field with placeholder 'número telefónico del cliente'.
- Correo Electrónico:** Input field with placeholder 'correo electrónico del cliente'.

At the bottom left is a 'Volver' button, and at the bottom right is a green 'Registrar Cliente' button.

FIGURA 3.27: Interfaz del formulario para crear nuevos contactos.

## 4. Validación de la Solución

---

En este capítulo se muestra como se realiza la evaluación de la propuesta de solución implementada, la cual fue descrita en el Capítulo 3. Se explica cómo se aplica la metodología de evaluación seleccionada junto con los procesos que esta involucra. Los resultados de la experimentación son representados mediante gráficos y tablas, los cuales generaran las evidencias para saber cómo funcionaría la solución en el mundo real.

### 4.1. Fases de Experimentación

En esta sección se especifican los procesos de la fase de experimentación para la evaluación del sistema.

#### 4.1.1. Definición

En esta fase se definen los objetivos principales que se consideran en la evaluación. Estos objetivos tienen una directa relación con el flujo que realizan los usuarios finales para la compra de un dominio.

##### **Objetivo Experimental 1:** Gestión de usuarios

El objetivo evalúa la funcionalidad relacionada con la creación de usuarios, específicamente los partners y clientes del sistema. Este objetivo tiene la finalidad de evaluar el correcto funcionamiento de la solución junto con las características que definen la calidad de esta función. La funcionalidad es probada mediante el registro de un partner por parte de un usuario administrador y también el registro de un cliente por parte de un usuario partner.

**Objetivo Experimental 2:** Compra de un dominio

El objetivo evalúa la funcionalidad de compra de un dominio en la propuesta de solución desarrollada. Este tiene el propósito de validar el correcto funcionamiento del flujo que realiza un cliente cuando compra un nombre de dominio. Además, verifica que la solución realice la comprobación de la disponibilidad de un nombre de dominio previamente. Esta prueba es realizada por parte de un usuario cliente. También incluye un servicio “Whois” que permite a los usuarios consultar acerca de la información de un dominio en particular registrado en el sistema.

**Objetivo Experimental 3:** Gestión de facturación

El objetivo evalúa el proceso de cargar fondos a la cuenta del partner y de facturación, después que un cliente solicita la compra de un nombre de dominio en particular. Este proceso tiene dos sub-procesos:

1. **Generar Orden:** Cuando un cliente realiza la solicitud de compra de un dominio, se debe generar una orden de compra que se registra en el sistema para posteriormente ser ejecutada.
2. **Generar Transacción:** Luego que se realiza el registro del nombre de dominio, procede a generar una transacción en el sistema para que el partner pueda tener un registro de las ventas que ha tenido, además de los cargos de fondos que se realizan en su cuenta.

El objetivo 3 se intenta que los cumplan tanto los usuarios partners como clientes.

### 4.1.2. Diseño

En el proceso de diseño se seleccionan características que se evalúan en la experimentación del sistema. Además, se describen los instrumentos de evaluación que se utilizan para dicho proceso. Por último, se mencionan las actividades que permiten realizar la experimentación.

#### 4.1.2.1. Características de Evaluación

A continuación se describe las características que se evaluaron durante la experimentación:

- **Funcionalidad:** Responde a la pregunta ¿Las funciones y propiedades satisfacen las necesidades explícitas e implícitas? El conjunto de atributos permite calificar si la solución maneja de forma adecuada las funcionalidades que satisfagan las necesidades para lo cual fueron creadas. En este caso, el atributo que se evalúa es:
  - **Exactitud:** Este atributo evalúa si el software entrega resultados acorde a las necesidades que este cubre.
- **Utilidad:** Permite conocer qué tan convenientes son las funcionalidades del sistema. Esta característica se evalúa según la percepción de los usuarios para aplicar la gestión y venta de nombre de dominios.

#### 4.1.2.2. Instrumentos de Medición

Los instrumentos de medición que se utilizan para cumplir los objetivos experimentales descritos anteriormente son los siguientes:

- **Encuestas:** son un conjunto de sentencias que están orientadas para evaluar una determinada característica de experimentación. En estas se crean sentencias referentes a la evaluación de sistema específicamente. Cada sentencia se evalúa según la escala de Likert [20].

#### 4.1.2.3. Proceso de Experimentación

Para esta etapa se define una serie de actividades que permiten realizar la experimentación de manera secuencial. Las actividades definidas son las siguientes:

1. Presentación de la plataforma que unifica los microservicios.
2. Capacitación de funcionalidades que puede realizar un usuario en la plataforma.
3. Definición de las actividades experimentales para el usuario de prueba.
4. Aplicación de las actividades a través de la plataforma.
5. Aplicación de la evaluación por parte de los usuarios de prueba utilizando instrumentos de evaluación.

Con estas actividades se pretende establecer una serie de pasos para aplicar durante la experimentación. También se organizan los procedimientos que se efectúan en las reuniones con los sujetos de prueba.

#### 4.1.3. Ejecución

En la etapa de experimentación se realizaron diversas actividades que permitieron utilizar los instrumentos de evaluación definidos previamente en la sección 4.1.2.2. Estas actividades están relacionadas con la gestión de usuarios y nombres de dominio.

Por otra parte, la selección de los usuarios se realiza según los dos tipos de perfiles de usuario existentes en la plataforma de partners. Para realizar la experimentación se seleccionaron 10 usuarios de tipo cliente y 2 usuarios de tipo partner. Además, durante la ejecución de la experimentación se da a conocer la presentación de la plataforma de partners, una breve capacitación a los usuarios, las actividades a realizar y finalmente la evaluación del plataforma.

**Presentación:** Consiste en mostrar la finalidad y funcionamiento general de la plataforma de partners, para integrar a los usuarios al contexto de la solución.

**Capacitación:** Consiste en demostrar a los usuarios las funcionalidades que pueden realizar en la plataforma mediante un ejemplo guiado.

**Actividades:** Son un conjunto de acciones que debe realizar el usuario para posteriormente evaluar la plataforma.

**Evaluación:** Consiste en la aplicar una encuesta a los usuarios para obtener información acerca de la experiencia resultante luego de finalizar todas las actividades definidas por tipo de usuario.

#### 4.1.4. Experimentación

En este experimento se evalúa la gestión de usuario (tanto partners como clientes) y también la gestión de nombres de dominio. La experimentación se realiza en una computadora con conexión a internet. Luego se procede a realizar las actividades definidas para la experimentación por cada tipo de usuario. Finalmente se evalúan las características de la plataforma con los instrumentos de medición.

En las Tablas 4.1 y 4.2 se presentan las actividades realizadas para cada tipo de usuario que interactúa en el sistema.

<b>Experimentación para el usuario cliente</b>	
<b>Sujeto de prueba:</b>	Cliente 1, Cliente 2, Cliente 3, Cliente 4, Cliente 5, Cliente 6, Cliente 7, Cliente 8, Cliente 9, Cliente 10.
<b>Objetivos:</b>	Objetivo Experimental 2: Compra de un dominio. Objetivo Experimental 3: Gestión de facturación.
<b>Descripción:</b>	En la experimentación se da a conocer la plataforma y se capacita a los usuarios clientes de prueba para la compra de dominio y gestión de facturación. Luego se efectúa la instrucción de las actividades a realizar en la experimentación. Dentro de las actividades, como primera acción se realiza el registro de los sujetos de prueba en la plataforma, también la verificación de la disponibilidad de dominios, la compra de dominios y la creación de ordenes de compra. Finalmente se aplica la evaluación de la plataforma por parte de los sujetos de prueba mediante los instrumentos de medición definidos.

TABLA 4.1: Actividades de la ejecución de la experimentación para el usuario cliente.

<b>Experimentación para el usuario partner</b>	
<b>Sujeto de prueba:</b>	Partner 1, Partner 2.
<b>Objetivos:</b>	Objetivo Experimental 1: Gestión de usuarios. Objetivo Experimental 3: Gestión de facturación.
<b>Descripción:</b>	<p>En la experimentación se da a conocer la plataforma y se capacita a los partners de prueba para la gestión de usuarios y gestión de facturación. Luego se efectúa la instrucción de las actividades a realizar en la experimentación.</p> <p>Como primera acción a realizar dentro de las actividades se encuentra el registro de los sujetos de prueba en la plataforma y el registro de clientes, también el ingreso de fondos, verificar las transacciones efectuadas en la plataforma, visualización de dominios registrados.</p> <p>Finalmente se aplica la evaluación de la plataforma por parte de los sujetos de prueba mediante los instrumentos de medición definidos.</p>

TABLA 4.2: Actividades de la ejecución de la experimentación para el usuario partner.

Ejecución de la Experimentación			
Nro	Usuarios	Tiempo	Proceso
Bloque Uno	Cliente 1	6 sesiones de 40 min. aprox.	<ul style="list-style-type: none"> <li>- Presentación</li> <li>- Capacitación</li> <li>- Actividades</li> <li>- Evaluación</li> </ul>
	Cliente 2		
	Cliente 3		
	Cliente 4		
	Cliente 5		
	Partner 1		
Bloque Dos	Cliente 6	6 sesiones de 40 min. aprox.	<ul style="list-style-type: none"> <li>- Presentación</li> <li>- Capacitación</li> <li>- Actividades</li> <li>- Evaluación</li> </ul>
	Cliente 7		
	Cliente 8		
	Cliente 9		
	Cliente 10		
	Partner 2		

TABLA 4.3: Tabla resumen de la ejecución de la experimentación.

En la Tabla 4.3 se muestra en detalle la ejecución de la experimentación, en la cual se realizaron 12 sesiones, en las que contamos con 10 usuarios de tipo cliente y 2 usuarios de tipo partner.

Cada sesión tuvo una duración de 40 minutos aproximadamente, considerando la ejecución de la lista de actividades y posteriormente la realización de la encuesta. La lista de actividades por tipo de usuario está disponible en el **Anexo B.1**.

Por otra parte, es necesario mencionar que existió dos tipos de perfiles de usuarios de prueba, entre los cuales destacan ingenieros civiles en computación o similares y estudiantes. En cuanto a los ingenieros podemos decir que pertenecen a la empresa Haulmer y los estudiantes son universitarios de la Universidad de Talca.

Finalmente cabe mencionar que existen muchas mejoras por realizar, sin embargo los usuarios de prueba se sintieron cómodos con la navegación en la plataforma y les resultó muy intuitiva. Finalmente el desarrollo de la experimentación arrojó resultados favorables, los cuales se exponen en la siguiente sección.

#### 4.1.5. Análisis

Luego de la ejecución de la experimentación, se elabora un análisis de los resultados obtenidos en las encuestas realizadas. Para la presentación de la evaluación se crean gráficos que muestran los resultados generados por tipo de usuario. Es importante mencionar que los resultados fueron recopilados en escala de Likert. A continuación se muestran los resultados obtenidos.

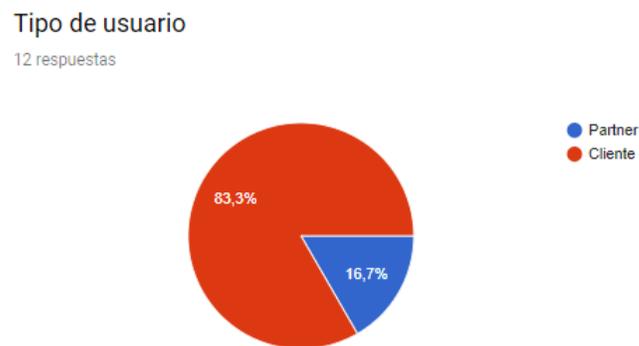


FIGURA 4.1: Usuarios de prueba.

En el gráfico anterior se muestra el porcentaje equivalente al tipo de usuario que realiza la encuesta. El 82,3 % corresponde a los usuarios de tipo partner y el 16,7 % corresponde a usuarios de tipo cliente.

Por otra parte, los usuarios de prueba se caracterizaban por tener la profesión de ing. civil en computación o la actividad de estudiante. Estos se dividían en porcentajes iguales, tal como se muestra en el siguiente gráfico:

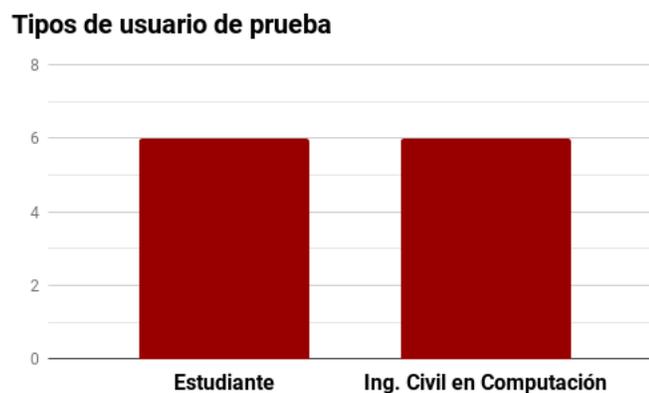


FIGURA 4.2: Caracterización de los usuarios de prueba.

#### 4.1.5.1. Resultados Usuario Cliente

En esta sección se entregan los resultados obtenidos de la experimentación realizada por los usuarios clientes, los cuales evaluaron las características “funcionalidad” y “utilidad” de la plataforma.

- **Funcionalidad:** Para evaluar esta característica se consulta por cinco sentencias:
  1. La plataforma permite verificar la disponibilidad de un nombre de dominio de manera exitosa.
  2. La plataforma permite comprar un dominio de manera exitosa.
  3. La plataforma permite renovar el periodo de caducidad de un dominio de manera exitosa.
  4. La plataforma permite ver las ordenes generadas por compras realizadas de manera exitosa.
  5. La plataforma permite crear contactos para añadirlos a los dominios de manera exitosa.

Luego de realizar la experimentación se obtuvo los resultados que se muestran en la Figura 4.3.

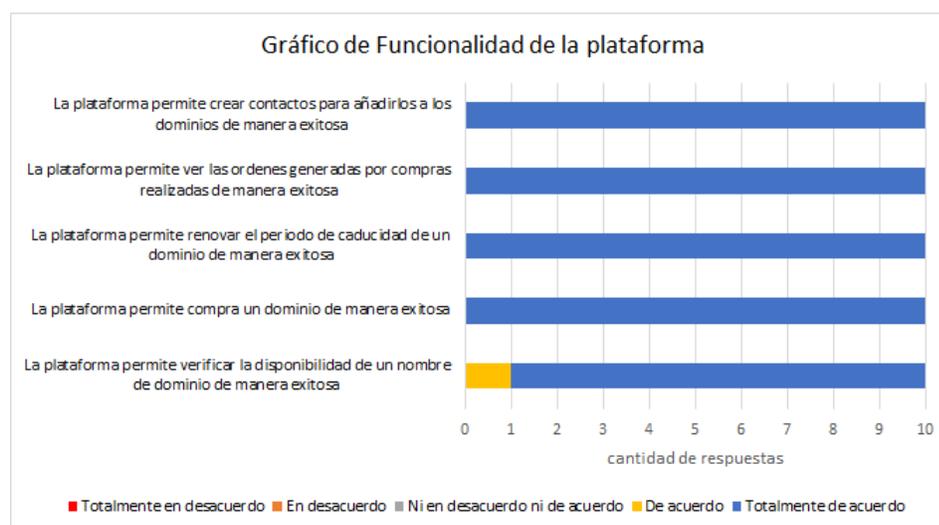


FIGURA 4.3: Resultado de evaluación de la característica funcionalidad.

EVALUACIÓN FUNCIONALIDAD CLIENTES						
	Pregunta 1	Pregunta 2	Pregunta 3	Pregunta 4	Pregunta 5	% de la suma de respuestas
 Totalmente en desacuerdo	0	0	0	0	0	0%
 En desacuerdo	0	0	0	0	0	0%
 Ni en desacuerdo ni de acuerdo	0	0	0	0	0	0%
 De acuerdo	1	0	0	0	0	2%
 Totalmente de acuerdo	9	10	10	10	10	98%

FIGURA 4.4: Tabla de datos de la funcionalidad de la plataforma.

Para la compra de dominios, en general fue bastante aceptada por parte de los clientes de prueba ya que se obtuvo un 98 % como respuesta “*Totalmente de acuerdo*” con que el sistema manejaba bien el flujo de compra de dominios. Apenas un 2 % indicó que sólo estaba de acuerdo con la sentencia “La plataforma permite verificar la disponibilidad de un nombre de dominio de manera exitosa”, lo cual muestra que pueden haber detalles por mejorar en esta función en específico. Por otra parte, para la gestión de transacciones todos los clientes de prueba estaban totalmente de acuerdo con la sentencia “La plataforma permite ver las ordenes generadas por compras realizadas de manera exitosa” por lo cual el objetivo se cumple satisfactoriamente.

■ **Utilidad:** Para evaluar esta característica se realizaron dos sentencias:

1. La plataforma envía las credenciales de acceso vía email de tu cuenta creada de manera exitosa.
2. La plataforma permite utilizar el servicio “whois” sobre un dominio registrado de manera exitosa.

Las Figuras 4.5 y 4.6 se presentan los resultados obtenidos en forma de gráfico y tabla.

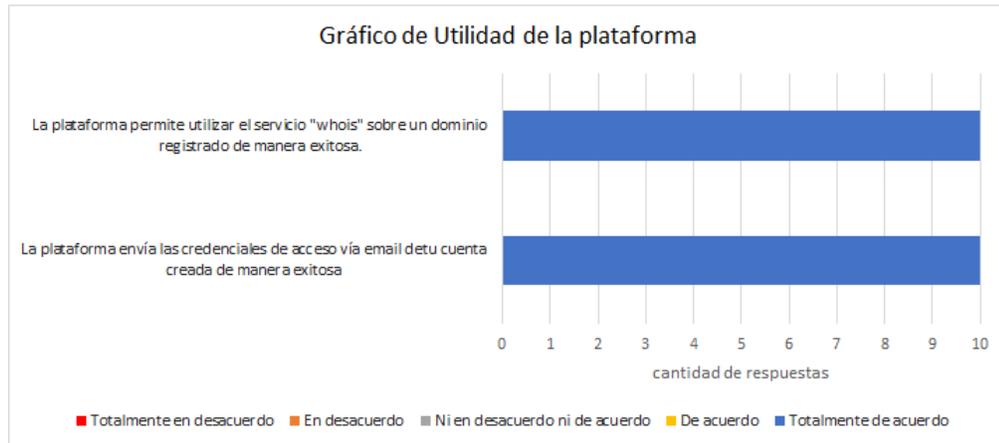


FIGURA 4.5: Resultado de evaluación de la característica utilidad.

EVALUACIÓN UTILIDAD CLIENTES			
	Pregunta 1	Pregunta 2	% de la suma de respuestas
Totalmente en desacuerdo	0	0	0%
En desacuerdo	0	0	0%
Ni en desacuerdo ni de acuerdo	0	0	0%
De acuerdo	0	0	0%
Totalmente de acuerdo	10	10	100%

FIGURA 4.6: Tabla de datos de la utilidad de la plataforma.

De acuerdo con los datos obtenidos de la experimentación, el 100 % de los clientes de prueba están totalmente de acuerdo con que el sistema es útil. Adicionalmente, con la ayuda de la información obtenida a partir de la observación de la experimentación (instrumento de medición), se obtuvo que a un bajo porcentaje de los clientes de prueba no les costó entender el flujo de compra de un dominio en particular.

Finalmente, se concluye que existe una oportunidad de mejora en relación con el flujo de compra de dominio y el servicio de “whois”.

#### 4.1.5.2. Resultados Usuario Partner

En esta sección se entregan los resultados obtenidos de la experimentación realizada por los usuarios partners, los cuales evaluaron las características “funcionalidad” y “utilidad” de la plataforma.

- **Funcionalidad:** Para evaluar esta característica se consulta por ocho sentencias:

1. La plataforma permite registrarse como partner de manera exitosa.
2. La plataforma crea los clientes de forma exitosa.
3. La plataforma permite visualizar a los clientes registrados.
4. La plataforma permite visualizar los nombres de dominios registrados por sus clientes.
5. La plataforma permite renovar el periodo de validez de un nombre de dominio de manera exitosa.
6. La plataforma permite ver las transacciones de manera exitosa.
7. La plataforma carga los fondos de forma exitosa.
8. La plataforma permite ver los fondos disponibles de su cuenta de manera exitosa.

Luego de realizar la experimentación se obtuvo los siguientes resultados mostrados en las Figuras 4.7 y 4.8.



FIGURA 4.7: Resultado de evaluación de la característica funcionalidad.

EVALUACIÓN FUNCIONALIDAD PARTNERS									
	Pregunta 1	Pregunta 2	Pregunta 3	Pregunta 4	Pregunta 5	Pregunta 6	Pregunta 7	Pregunta 8	% de la suma de respuestas
 Totalmente en desacuerdo	0	0	0	0	0	0	0	0	0%
 En desacuerdo	0	0	0	0	0	0	0	0	0%
 Ni en desacuerdo ni de acuerdo	0	0	0	0	1	0	0	0	6%
 De acuerdo	0	0	2	0	0	0	0	0	13%
 Totalmente de acuerdo	2	2	0	2	1	2	2	2	81%

FIGURA 4.8: Tabla de datos de la funcionalidad de la plataforma.

De acuerdo con los datos obtenidos, el 81 % de los clientes de prueba están totalmente de acuerdo que el sistema es funcional. Luego, para la sentencia “*La plataforma permite visualizar a los clientes registrados*” se reflejó un 13 % que estaban de acuerdo con la gestión de usuarios, lo cual se puede apreciar detalles por mejorar en este caso específico. Por otra parte, podemos visualizar que un 6 % de los partners de prueba no está ni de acuerdo ni en desacuerdo con la sentencia “*La plataforma permite renovar el periodo de validez de un nombre de dominio de manera exitosa*”, en donde se encontraron oportunidades de mejoras en esta funcionalidad.

Finalmente en cuanto a la gestión de transacciones se puede apreciar una gran aceptación por parte de los partners de pruebas, debido a que la mayoría de las respuestas asociadas a este objetivo fueron “totalmente de acuerdo”.

■ **Utilidad:** Para evaluar esta característica se realizaron tres sentencias:

1. La plataforma envía las credenciales de acceso vía email de tu cuenta creada de manera exitosa.
2. La plataforma envía las credenciales de acceso vía email de una nueva cuenta de cliente creada de manera exitosa.
3. La plataforma permite utilizar el servicio “whois” sobre un dominio registrado de manera exitosa.

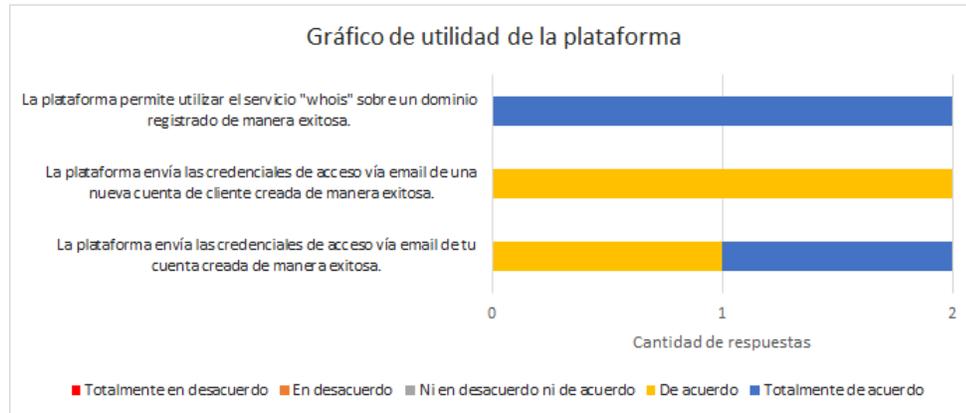


FIGURA 4.9: Resultado de evaluación de la característica utilidad.

EVALUACIÓN UTILIDAD PARTNERS				
	Pregunta 1	Pregunta 2	Pregunta 3	% de la suma de respuestas
Totalmente en desacuerdo	0	0	0	0%
En desacuerdo	0	0	0	0%
Ni en desacuerdo ni de acuerdo	0	0	0	0%
De acuerdo	1	2	0	50%
Totalmente de acuerdo	1	0	2	50%

FIGURA 4.10: Tabla de datos de la utilidad de la plataforma.

Los resultados obtenidos de la experimentación que midieron los indicadores de utilidad del área de usuarios partners de la plataforma que se ven en las Figuras 4.9 y 4.10 arrojó opiniones divididas, por lo cual no podemos tener una clara confianza en que el sistema es de gran utilidad para los partners. Debido a los resultados obtenidos nacen oportunidades de mejorar la utilidad de la plataforma, específicamente en el área de usuarios partners.

Finalmente se concluye que la plataforma puede mejorar en el envío de credenciales de usuarios vía email, ya que principalmente fueron las dos sentencias que involucran esta características, las que obtuvieron una baja aceptación por parte de los partner de prueba.

#### 4.1.5.3. Resultados Evaluación General

Con la finalidad de conocer la percepción general de la plataforma por parte de los usuarios de prueba, se consideraron algunas preguntas que se presentan a continuación:

- **¿La plataforma presentó algún error al momento de ser utilizado?**

Las respuestas a esta pregunta se muestran en las Figuras 4.11



FIGURA 4.11: Resultados evaluación de fallos de los clientes en el sistema.

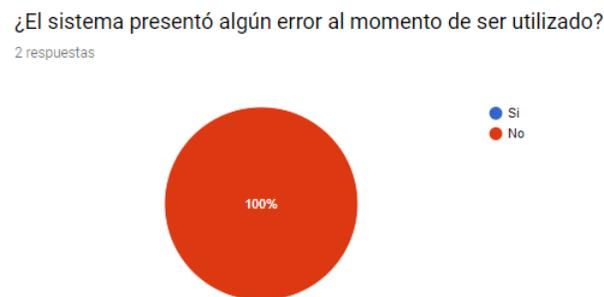


FIGURA 4.12: Resultados evaluación de fallos de los partners en la plataforma.

- ¿Utilizaría la plataforma de Partner para comprar o revender nombre de dominios .CL ?

*Respuestas Clientes:*

- Si, lo usaría.
- Si, pero aún falta por mejorar algunos aspectos.
- Puede ser.
- Si, es una plataforma muy intuitiva.
- Por supuesto, además lo recomendaría.
- Si, es algo novedoso.
- Si, me gustó bastante la idea.
- Si, obvio.
- Sin duda lo haría.
- Si, muy interesante la plataforma.

En síntesis, de las 10 respuestas de los clientes de prueba; nueve fueron respuestas positivas y una respuesta parcial (“*puede ser*”). Es posible que el cliente que contestó parcialmente haya pensado que la plataforma aún no estaba del todo funcional.

*Respuestas Partners:*

- Si, me parece una plataforma fácil de usar.
- Es una plataforma simple de usar, sin duda lo usaría.

En resumen, las respuestas de los partners de prueba fueron bastante positivas.

▪ **¿Qué cambios o mejoras le haría usted a la plataforma?**

Respuestas clientes:

- La plataforma podría enviar una notificación cuando el registro de un nombre de dominio haya completado.
- Resaltar botón regresar atrás y validar caracteres especiales.
- Agregaría una vista predeterminada de los periodos, listando los años y valores en una sola función, no seleccionando el periodo manual, para que el cliente al hacer clic en el periodo de tiempo tenga el valor inmediato.
- Agregar la opción de transferencia de nombre de dominios a otro cliente.
- Ningún cambio.
- Mejorar la familia fonética del sitio o ayudar con *tooltips*.
- Añadir la opción de cambiar o recuperar contraseña.
- Ninguno cambio por el momento.
- Agregar “Listados” a los input de comuna, ciudad y región.
- Podría entregar más información acerca del dominio al finalizar la compra.

Respuestas Partners:

- Se podría asociar un medio de pago más completo, por ejemplo; una plataforma externa como WebPay.
- Más información acerca los dominios y clientes en el área administrativa de partners.

■ **Cómo le ha parecido PartnerSite respecto a:**

1. Funcionalidad en general de servicio.
2. Utilidad en general de servicios.

Los resultados de esta pregunta se ven en las Figuras 4.13 y 4.14.

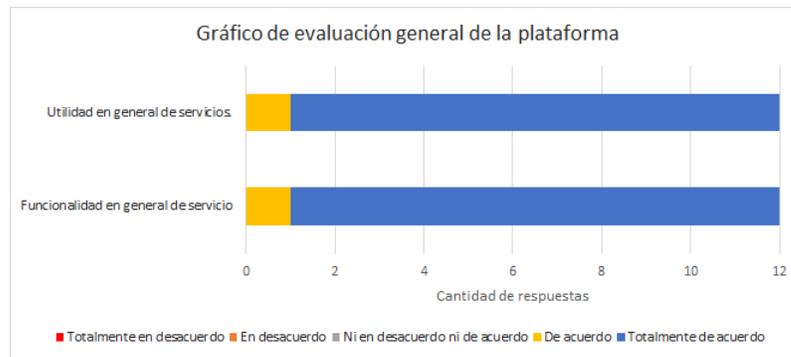


FIGURA 4.13: Resultados evaluación general del sistema.

EVALUACIÓN GENERAL			
	Pregunta 1	Pregunta 2	% de la suma de respuestas
Totalmente en desacuerdo	0	0	0%
En desacuerdo	0	0	0%
Ni en desacuerdo ni de acuerdo	0	0	0%
De acuerdo	1	1	8%
Totalmente de acuerdo	11	11	92%

FIGURA 4.14: Tabla de datos de la evaluación general del sistema.

Como síntesis del capítulo y luego de analizar los resultados obtenidos en la evaluación de la plataforma se puede deducir que se cumplió con los objetivos experimentales propuestos.

Para el Objetivo Experimental 1: “*Gestión de Usuarios*”, en promedio el 71,4% está *totalmente de acuerdo* con que se cumplió este objetivo. Para el Objetivo Experimental 2: “*Compra de un dominio*”, en promedio el 93% de los usuarios de prueba está *totalmente de acuerdo* con que se cumplió este objetivo. Finalmente para el Objetivo Experimental 3: “*Gestión de Facturación*”, podemos deducir que todos los usuarios de prueba están *totalmente de acuerdo* con que este objetivo se cumplió satisfactoriamente.

## 5. Conclusiones y Trabajo Futuro

---

En este último capítulo se presentan las conclusiones generales obtenidas del proceso de desarrollo de la propuesta solución. También se expone el cumplimiento de los objetivos y se analiza si la solución es realmente útil para el negocio. Por otra parte, se redactan las lecciones aprendidas tras finalizar este proceso de desarrollo y finalmente se especifica los trabajos a futuro que puedan existir.

### 5.1. Conclusiones

Al comenzar el desarrollo de este proyecto se planteó el siguiente objetivo principal; *“Desarrollar un ecosistema basado en microservicios que permita la integración de sistemas externos que involucra la administración y reventa de dominios .CL, así como también la gestión de Reseller y sus clientes”*. Luego de analizar los resultados obtenidos en la evaluación de la plataforma, se puede deducir que se cumple con las características funcionales para lograr el objetivo principal propuesto. Como síntesis, en promedio el 84% de las respuestas obtenidas en las encuestas realizadas a los usuarios cliente y partner indican que están *“totalmente de acuerdo”* con la funcionalidad, utilidad y veracidad de los datos adquiridos.

Durante la implementación del proyecto se logró llevar a cabo las actividades necesarias para el cumplimiento de los objetivos específicos. En particular, para el primer objetivo *“Especificar los requerimientos que satisfacen la reventa y administración de dominios .CL”*, se generó un documento de especificación de requisito en una plataforma de la empresa Haulmer. Este documento generado contiene los requisitos formalizados bajo los estándares que la empresa maneja y se encuentran disponibles en el **Anexo A**.

Para el objetivo “*Generar un diseño de la capa lógica para la gestión y reventa de dominios .CL basado en una arquitectura de microservicios*”, se diseñó un diagrama que cubría las necesidades del sistema, y que se encuentra disponible en la **Sección 3.2.2** del presente documento.

Por otra parte, para los objetivos “*Implementar la interacción para la reventa de dominios .CL a través de un proveedor externo*”, “*Desarrollar back-end del microservicio para la gestión y reventa de dominios .CL*”, “*Desarrollar back-end del microservicio para la gestión de Reseller y sus clientes finales*”, “*Desarrollar capa de presentación que nos permita visualizar el flujo que tiene la administración y reventa de un dominio .CL*”, se observa el cumplimiento de los objetivos dado los resultados obtenidos en la etapa de implementación de la solución del sistema, el cual queda en completa evidencia en el **Capítulo 3**.

Respecto al objetivo “*Aplicar evaluación del ecosistema de microservicios en conjunto con el cliente para comprobar una completa integración tanto con la plataforma de venta de los Reseller como con sistemas externos de la empresa*”, se aplicó la metodología de evaluación seleccionada y especificada en la **Sección 2.5**, la que durante su aplicación se aprobó la utilidad y funcionamiento del sistema ya que se obtuvo resultados satisfactorios. Para realizar este objetivo se encuestó a los usuarios de prueba según su tipo, dicha encuesta está disponible en el **Anexo B.2**.

Finalmente, se determinó que la implementación del sistema puede ser aplicada al mundo real, pero teniendo la consideración de incluir algunas mejoras que son relevantes para un funcionamiento apropiado. Sin embargo, quedó en evidencia el cumplimiento de los objetivos iniciales dado que los resultados entregados por los usuarios fue bastante positivo.

## 5.2. Lecciones Aprendidas

Luego de realizar el proceso de desarrollo de la propuesta solución, cabe mencionar que se obtuvieron una serie de lecciones que permiten el mejoramiento en cada fase involucrada durante la implementación. Es importante aprender de cada error o equivocación en el desarrollo de software, pero también hay que considerar las fortalezas manifestadas en el proceso. Básicamente podemos encontrar lecciones del tipo metodológicas, tecnológicas y de mercado, entre otras.

En cuanto a las lecciones de tipo metodológica son aquellas que se involucran con la aplicación de la metodología escogida. En estas se puede mencionar que el mayor desafío fue aprender acerca la metodología híbrida ya que es la que utiliza la empresa Haulmer para el desarrollo de sus productos. Otra lección de gran valor fue la experiencia adquirida al aplicar las metodologías tanto de desarrollo como de evaluación en el proyecto, ya que cada fase fue un reto con sus respectivas dificultades.

Las lecciones tecnológicas son las que se obtienen tras la utilización de diferentes tecnologías. Dentro de estas lecciones está el aprendizaje obtenido al utilizar el protocolo EPP, ya que es un protocolo que lo utilizan todos los proveedores de dominios para su gestión, por lo tanto era un aprendizaje crucial para el desarrollo del sistema. Otro aprendizaje fue el conocimiento obtenido al utilizar una arquitectura basada en microservicios.

Las lecciones de mercado son todas aquellas relacionadas con el impacto de la solución desarrollada en el negocio. La principal lección fue aprender el flujo de compra y reventa de nombres de dominio que existe actualmente en el mercado. Adicionalmente gracias a los comentarios y críticas por parte de los usuarios de prueba se dio a entender que esta solución puede tener un impacto positivo en el mercado objetivo.

### 5.3. Trabajos a Futuro

Dentro de las oportunidades de mejoras encontradas por los usuarios de prueba en la evaluación de la solución se destacan las siguientes:

- Resaltar botones para volver atrás durante la compra.
- Validar campos y caracteres especiales.
- Agregar una lista con los años predeterminados cuando se renueve un dominio, en vez de realizar la acción manualmente.
- Agregar un listado con las regiones y sus ciudades respectivamente, para así evitar que el usuario ingrese ese tipo de datos manualmente.
- Entregar más información del dominio solicitado al momento de realizar el último paso en el proceso de compra.
- Visualizar más información acerca de los clientes y dominios en el área de partners.
- Notificación vía correo electrónico del proceso de compra.

Finalmente es importante mencionar que la solución lograda puede mejorar y ser aún más útil para los usuarios que la utilicen. Sin embargo es necesario decir que hay mejoras que se deben realizar con alta prioridad, como por ejemplo; incorporar sistemas de pagos, validar caracteres especiales y tipos de datos e incluir la funcionalidad de transferencia de nombres de dominio entre usuarios registrados.

# Bibliografía

- [1] Gerard Briscoe y Philippe De Wilde. «Digital Ecosystems: Evolving Service-Orientated Architectures». En: *CoRR* abs/0712.4102 (2007). URL: <http://arxiv.org/abs/0712.4102>.
- [2] Sam Newman. *Building microservices: designing fine-grained systems*. O'Reilly Media, Inc., 2015.
- [3] Gavin Brown. *The Extensible Provisioning Protocol*. [www.uknof.org.uk](http://www.uknof.org.uk). Disponible en el siguiente enlace <https://www.uknof.org.uk/uknof5/Brown-EPP/>. Consultado el 23 de Abril de 2018.
- [4] Scott Hollenbeck. *Extensible Provisioning Protocol*. [ietf.org](http://ietf.org). Disponible en el siguiente enlace <https://tools.ietf.org/html/rfc3730>. Consultado el 24 de Abril de 2018.
- [5] Bass, Clements y Kazman. *Software Architecture in Practice: Views and Beyond*. Addison-Wesley Professional, 19 de Abril, 2003.
- [6] May ISO. *Systems and software engineering—architecture description*. Inf. téc. ISO/IEC/IEEE 42010, 2011.
- [7] Eric J. Evans. *Domain-Driven Design: Tackling Complexity in the Heart of Software*. Addison Wesley Professional, August 01, 2003.
- [8] James Lewis y Martin Fowler. «Microservices, a Definition of This New Architectural Term.» En: URL: <https://martinfowler.com/articles/microservices.html> (2014).
- [9] *Back-End Web Architecture*. <https://www.codecademy.com>. Disponible en el siguiente enlace <https://www.codecademy.com/articles/back-end-architecture>. Consultado el 05 de Octubre de 2018.

- [10] *What is backend as a service(BaaS)?* <https://www.quora.com>. Disponible en el siguiente enlace <https://www.quora.com/What-is-backend-as-a-service-BaaS>. Consultado el 05 de Octubre de 2018.
- [11] *Definition of: back end.* [www.pcmag.com](http://www.pcmag.com). Disponible en el siguiente enlace <https://www.pcmag.com/encyclopedia/term/38340/back-end>.
- [12] MA Awad. «A comparison between agile and traditional software development methodologies». En: *University of Western Australia* (2005).
- [13] C Wohlin y col. «Experimentation in Software Engineering-An Introduction. Kluwer Academic Publishers». En: *Doedrecht the Netherlands* (2000).
- [14] Marcela Genero Bocco y José A. Cruz-Lemus. «Experimentación en Ingeniería de Software». En: *Universidad de Castilla-La Mancha* (2010).
- [15] *DB-Engines Ranking Top Ten.* <https://db-engines.com>. Disponible en el siguiente enlace <https://db-engines.com/en/ranking>. Consultado el 08 de Octubre de 2018.
- [16] *World's Most Popular Open Source Database.* <https://www.oracle.com>. Disponible en el siguiente enlace <https://www.oracle.com/mysql/>.
- [17] Guido Van Rossum y Fred L Drake. *Python language reference manual.* Network Theory United Kingdom, 2003.
- [18] Stig Saether Bakken, Zeev Suraski y Egon Schmid. *PHP Manual: Volume 1.* iUniverse, Incorporated, 2000.
- [19] Martin Bean. *Laravel 5 essentials.* Packt Publishing Ltd, 2015.
- [20] Joseph A Gliem y Rosemary R Gliem. «Calculating, interpreting, and reporting Cronbach's alpha reliability coefficient for Likert-type scales». En: Midwest Research-to-Practice Conference in Adult, Continuing, y Community .... 2003.

# ANEXOS

## A. Documentos de Apoyo

---

### A.1. Documento de Requisitos

# FNC: 1. Definición del Proyecto

<b>Project name</b>	NIC Domain Registrar
<b>JIRA Main Issue</b>	
<b>Document status</b>	<b>DRAFT</b>
<b>Requirement status</b>	<b>TO DO</b>
<b>Document owner</b>	Hernan Arturo Galvez Neculman
<b>Assignee</b>	Hernan Arturo Galvez Neculman
<b>Stakeholders</b>	Felipe Rivas, Fernando Guerra

Related links	
Drive Folder	<ul style="list-style-type: none"><li><a href="https://drive.google.com/drive/folders/1lq15SE_uLS2lrKEC_tNS1aoDDMwBPr_h?usp=sharing">https://drive.google.com/drive/folders/1lq15SE_uLS2lrKEC_tNS1aoDDMwBPr_h?usp=sharing</a></li></ul>

## Related Process Diagrams (BPMN 2.0)

### Related Process

There are no images attached to this page.

## Goals

- **Objetivo General:**
  - Desarrollar un ecosistema basado en una arquitectura de microservicios que permita la gestión de dominios punto CL.
- **Objetivos Específicos:**
  - Especificar los requerimientos que satisfacen la reventa y administración de dominios punto CL.
  - Generar un diseño de la capa lógica para la gestión de dominios punto CL basado en una arquitectura de microservicios .
  - Implementar una conexión para la reventa de dominios punto CL a través de un proveedor externo.
  - Desarrollar back-end de microservicios para la gestión de dominios punto CL.
  - Aplicar evaluación del ecosistema de microservicios en conjunto con el cliente para comprobar una completa integración tanto con sistemas externos como internos de la empresa.

## Background and strategic fit

La problemática pasa principalmente por la manera en que la empresa realiza el proceso de venta de dominios debido a la falta de requerimientos básicos que exige un proveedor externo para gestionar los dominios punto CL. Por otra parte, la empresa debe cumplir con estándares y protocolos de venta de dominios que actualmente le exigen sus proveedores de servicio. Cabe mencionar la carencia de la infraestructura tecnológica que posee la empresa para gestionar dominios punto CL, ya que actualmente sus clientes no pueden obtener este servicio.

La propuesta esta sujeta a los términos y condiciones que exija una entidad externa para gestionar dominios punto CL, que en este caso será NIC Chile, único proveedor que actualmente ofrece este servicio en Chile. Para la reventa de dominios, la empresa requiere implementar una conexión e integración con NIC Chile, la cual le otorgar´ a un certificado que lo autorice. Sin embargo, para realizar dicha implementación se requiere desarrollar una cantidad finita de microservicios, que podr´ a variar entre 20 a 30 en total, el cual llamaremos ecosistema de microservicios.

Por otra parte, la empresa otorga la opción de tener reseller(Revendedor: adj. Que revende), quienes podrán revender este servicio como ellos estimen conveniente. Cabe destacar que la propuesta sólo contempla el desarrollo back-end del ecosistema de microservicios que realiza la gestión de dominios y reseller, la cual permite realizar las siguientes acciones:

- Gestión de dominios.
- Gestión de usuarios(Sólo de reseller).
- Interacción con sistemas o servicios internos y externos de la empresa.

Finalmente, es necesario mencionar que este ecosistema de microservicios se tendrá que comunicar con otros servicios internos de la empresa, dentro de los cuales encontramos el keycloak, whmcs.

## Assumptions

- En este proyecto se espera implementar un ecosistema de microservicios, por lo tanto se propone desarrollar la capa lógica del sistema que permita la interacción con un proveedor externo de venta de dominios punto CL y además una integración con sistemas internos de la empresa.
- En este proyecto no se desarrollará interfaz de usuario o similares que permita la gestión de dominios. Quedará como trabajo futuro la implementación de Front-End del proyecto.
- Este proyecto se limita a utilizar el protocolo de comunicación EPP para la interacción con NIC Chile.

## Requirements

Conexión NIC Chile				
	Item	Descripción Textual	Descripción Gráfica	Enlace
1	<input type="checkbox"/> General	Realizar conexión con los servicios de NIC Chile a través de microservicios, los cuales deben incluir un login básico y el típico "Hello" mediante el protocolo EPP.		<a href="#">FNC: Conexión NIC Chile</a>
2	<input type="checkbox"/> Usuario	Realizar microservicios que cree y actualice un usuarios mediante el protocolo EPP.		<a href="#">FNC: Conexión NIC Chile</a>
3	<input type="checkbox"/> Dominio	Realizar microservicios que gestionen dominios, osea que realice acciones como revisar, crear, actualizar, renovar, borrar, restaurar un dominio mediante el protocolo EPP.		<a href="#">FNC: Conexión NIC Chile</a>
Extranet Web				
	Item	Descripción Textual	Descripción Gráfica	Enlace
4	<input type="checkbox"/> Usuarios	Realizar microservicios que gestione usuarios realizando acciones como login, cambiar contraseña, crear administrador, crear lector, cambio de permisos de usuario admin a lector y viceversa, eliminar usuario mediante el protocolo EPP.		<a href="#">FNC: Extranet Web NIC Chile</a>
5	<input type="checkbox"/> Transacciones	Implementar un microservicio que realice operaciones de búsqueda mediante el protocolo EPP.		<a href="#">FNC: Extranet Web NIC Chile</a>
6	<input type="checkbox"/> Facturas	Desarrollar un microservicio que notifique el pago(se procesa para aceptarlo) y otro microservicio que notifique el pago (se procesa para rechazar y el Registrador tiene que volver a notificar el error de evento según se le asigne. Por ejemplo, importe incorrecto)		<a href="#">FNC: Extranet Web NIC Chile</a>
Manager Microservicios				
	Item	Descripción Textual	Descripción Gráfica	Enlace
7	<input type="checkbox"/> Administrador de Peticiones	Implementar un microservicio que administre las peticiones que realicen los usuarios al sistema, a su vez dando respuesta de parte del sistema.		<a href="#">FNC: Manager Microservicios</a>
8	<input type="checkbox"/> Manejo de Logs	Implementar un microservicio que maneje los logs de acciones que se realizan en el sistema.		<a href="#">FNC: Manager Microservicios</a>
Administración de Usuarios				
	Item	Descripción Textual	Descripción Gráfica	Enlace
9	<input type="checkbox"/> Manejo de Reseller	Implementar un microservicio que administre a los usuarios del sistema que en este caso corresponde a los Reseller.		<a href="#">FNC: Administración de Usuarios</a>
10	<input type="checkbox"/> Administrar Fondos	Implementar un microservicio que maneje los fondos de las cuentas de cada usuarios, así poder conocer los fondos disponibles, cargar fondos, descontar fondos para un usuario en particular.		<a href="#">FNC: Administración de Usuarios</a>

## Related documents

## Questions

Below is a list of questions to be addressed as a result of this requirements document:

Question	Outcome

## Pending

## Not Doing

# FNC: Administración de Usuarios

Project name	NIC Domain Registrar
JIRA Main Issue	
Document status	DRAFT
Requirement status	TO DO
Document owner	Hernan Arturo Galvez Neculman
Assignee	Hernan Arturo Galvez Neculman
Stakeholders	Felipe Rivas, Fernando Guerra

Related links	
Drive Folder	<ul style="list-style-type: none"><li><a href="https://drive.google.com/drive/folders/1lq15SE_uLS2lrKEC_tNS1aoDDMwBPr_h?usp=sharing">https://drive.google.com/drive/folders/1lq15SE_uLS2lrKEC_tNS1aoDDMwBPr_h?usp=sharing</a></li></ul>

## Related Process Diagrams (BPMN 2.0)

### Related Process

There are no images attached to this page.

## Goals

- Entender la interacción que tienen los usuarios en el sistema.

## Background and strategic fit

**Manejo de Reseller:** Se debe administrar a los usuarios del sistema, estos usuarios son los Reseller que interactúan en el sistema.

**Administrar Fondos:** Cada Reseller tendrá fondos asociados. Los fondos son dinero que se puede usar dentro del sistema para comprar dominios. Es por esto que es necesario tener un microservicio que se dedique a administrar estos fondos, para saber si un Reseller puede comprar dominios.

## Assumptions

- Debemos asumir que un Reseller solo podrá comprar un dominio si tiene fondos disponible en su cuenta.

## Requirements

Administración de Usuarios		
Manejo de Reseller		
1	Crear Reseller	Implementar un microservicio que cree un Reseller en el sistema.

2	Actualizar Reseller	Implementar un microservicio que actualice información de un Reseller.
3	Eliminar Reseller	Implementar un microservicio que eliminé a un Reseller del sistema.
	<b>Administrar Fondos</b>	
4	Consultar Fondos	Implementar microservicio que consulte los fondos disponibles para un Reseller en particular.
5	Cargar Fondos	Implementar un microservicio que agregue fondos a un Reseller en particular.

## Related documents

1. [Contexto del Proyecto](#)

## Questions

Below is a list of questions to be addressed as a result of this requirements document:

Question	Outcome

## Pending

## Not Doing

# FNC: Conexión NIC Chile

Project name	NIC Domain Registrar
JIRA Main Issue	
Document status	DRAFT
Requirement status	TO DO
Document owner	Hernan Arturo Galvez Neculman
Assignee	Hernan Arturo Galvez Neculman
Stakeholders	Felipe Rivas, Fernando Guerra

Related links	
Drive Folder	<ul style="list-style-type: none"><li><a href="https://drive.google.com/drive/folders/1lq15SE_uLS2lrKEC_tNS1aoDDMwBPr_h?usp=sharing">https://drive.google.com/drive/folders/1lq15SE_uLS2lrKEC_tNS1aoDDMwBPr_h?usp=sharing</a></li></ul>

## Related Process Diagrams (BPMN 2.0)

### Related Process

There are no images attached to this page.

## Goals

- Conocer los distintos micro servicios que nos permita gestionar dominios punto CL y entender como trabajar en conjunto con el proveedor NIC Chile.

## Background and strategic fit

La idea principal de describir estos requisitos, es poder entender como el proveedor de dominios NIC Chile trabaja con sus clientes para que funcionen como *Registrar*(Registrador: Encargado del registro). También es necesario conocer el protocolo de comunicación que tienen para interactuar con su sistema de manera eficiente.

## Assumptions

- Solo los *Registrar* que estén registrado en el sistema podrán utilizar estas funcionalidades, y siempre teniendo en cuenta que necesitan contar con fondos en sus cuentas para poder registrar un dominio.

## Requirements

General	
Item	Descripción

1	<input type="checkbox"/> login	Desarrollar microservicio que nos permita acceder a los servicios que Nic Chile, mediante el protocolo EPP.
2	<input type="checkbox"/> hello	Implementar función que nos permita obtener el típico "Hello World" desde los servicios de Nic Chile, mediante el protocolo EPP.
<b>Usuario</b>		
3	<input type="checkbox"/> create contact	Implementar un microservicio que nos permita crear un contacto que tenga la facultad de acceder a los servicios de compra de NIC Chile, mediante el protocolo EPP.
4	<input type="checkbox"/> update contact	Implementar un microservicio que nos permita actualizar información de un contacto que se encuentre registrado en el sistema y tenga la capacidad de acceder a servicios de NIC Chile, mediante el protocolo EPP.
<b>Dominio</b>		
5	<input type="checkbox"/> check	Implementar un microservicio que permita verificar en los servicios de NIC Chile si un dominio en particular esta disponible para ser usado por algún contacto.
6	<input type="checkbox"/> create	Desarrollar un microservicio que permitir poder registrar un dominio punto CL en los servicios de NIC Chile para un contacto en particular, mediante el protocolo EPP.
7	<input type="checkbox"/> update	Implementar un microservicio que nos permita actualizar un dominio en particular que este registrado dentro del sistema de NIC Chile, mediante el protocolo EPP.
8	<input type="checkbox"/> renew	Implementar un microservicio que nos permita renovar un dominio en particular que este registrado dentro del sistema de NIC Chile, mediante el protocolo EPP.
9	<input type="checkbox"/> delete	Implementar un microservicio que nos permita eliminar un dominio en particular que este registrado dentro del sistema de NIC Chile, mediante el protocolo EPP.
10	<input type="checkbox"/> restore	Implementar un microservicio que nos permita restaurar un dominio en particular que este registrado dentro del sistema de NIC Chile, mediante el protocolo EPP.

## Related documents

1. [Contexto del Proyecto](#)

## Questions

Below is a list of questions to be addressed as a result of this requirements document:

Question	Outcome

## Pending

## Not Doing

# FNC: Extranet Web NIC Chile

Project name	NIC Domain Registrar
JIRA Main Issue	
Document status	DRAFT
Requirement status	TO DO
Document owner	Hernan Arturo Galvez Neculman
Assignee	Hernan Arturo Galvez Neculman
Stakeholders	Felipe Rivas, Fernando Guerra

Related links	
Drive Folder	<ul style="list-style-type: none"><li><a href="https://drive.google.com/drive/folders/1lq15SE_uLS2lrKEC_tNS1aoDDMwBPr_h?usp=sharing">https://drive.google.com/drive/folders/1lq15SE_uLS2lrKEC_tNS1aoDDMwBPr_h?usp=sharing</a></li></ul>

## Related Process Diagrams (BPMN 2.0)

### Related Process

There are no images attached to this page.

## Goals

- Gestionar el acceso y roles de los clientes que se involucran en la plataforma. Por otra parte también se requiere gestionar las transacciones

## Background and strategic fit

- **Gestión de Acceso:** Se debe gestionar el acceso que tiene cada cliente, administrando las acciones que puede realizar, como por ejemplo, acceder a la plataforma con un nombre de usuario y contraseña, cambiar contraseña, crear administrar clientes administradores y lectores, cambiar roles de administrador a lector o viceversa, borrar clientes.
- **Gestionar Transacciones:** Se debe gestionar las transacciones que ocurren en el sistema, realizando acciones de búsqueda de alguna transacción en particular.
- **Gestionar Facturas:** Se debe gestionar las facturas que se realizan durante el proceso de pago, específicamente se requiere notificar si un pago fue aceptado o rechazado.

## Assumptions

## Requirements

Extranet Web
Usuarios

1	<input type="checkbox"/> Login	Implementar un microservicio que gestione el acceso de los clientes al sistema, mediante un nombre de usuario y contraseña.
2	<input type="checkbox"/> Cambiar Contraseña	Desarrollar un microservicio que permita a un cliente en particular, cambiar.
3	<input type="checkbox"/> Usuario Admin	Desarrollar un microservicio que permita al sistema poder crear usuarios administrador.
4	<input type="checkbox"/> Usuario Lector	Desarrollar un microservicio que permita al sistema poder crear usuarios lectores.
5	<input type="checkbox"/> Cambiar Roles	Implementar un microservicio que permita a un usuario poder cambiar de rol, ya sea de un rol de administrador a lector o viceversa.
6	<input type="checkbox"/> Borrar Usuario	Implementar un microservicio que permita al sistema poder borrar un usuario en particular.
<b>Transacciones</b>		
7	<input type="checkbox"/> Búsqueda	Se debe implementar un microservicio que permita la búsqueda de alguna transacción realizada.
<b>Facturas</b>		
8	<input type="checkbox"/> Pago Aceptado	Se debe implementar una funcionalidad que notifique a un cliente cuando el pago fue realizado con éxito.
9	<input type="checkbox"/> Pago Rechazado	Se debe implementar una funcionalidad que notifique a un cliente cuando un pago fue rechazo, informándole el motivo por el cual fue rechazado.

## Related documents

1. [Contexto del Proyecto](#)

## Questions

Below is a list of questions to be addressed as a result of this requirements document:

Question	Outcome

## Pending

## Not Doing

# FNC: Manager Microservicios

Project name	NIC Domain Registrar
JIRA Main Issue	
Document status	DRAFT
Requirement status	TO DO
Document owner	Hernan Arturo Galvez Neculman
Assignee	Hernan Arturo Galvez Neculman
Stakeholders	Felipe Rivas, Fernando Guerra

Related links	
Drive Folder	<ul style="list-style-type: none"><li><a href="https://drive.google.com/drive/folders/1lq15SE_uLS2lrKEC_tNS1aoDDMwBPr_h?usp=sharing">https://drive.google.com/drive/folders/1lq15SE_uLS2lrKEC_tNS1aoDDMwBPr_h?usp=sharing</a></li></ul>

## Related Process Diagrams (BPMN 2.0)

### Related Process

There are no images attached to this page.

## Goals

- Entender las funcionalidades que tiene el Administrador de Microservicios.

## Background and strategic fit

Con el propósito de entregar solución al problema, se requiere implementar un administrador de microservicios. Este administrador contará con dos funciones, las cuales son:

- Administrador de Peticiones:** Esta funcionalidad permite administrar las distintas peticiones que llegan al sistema, las cuales posteriormente ejecutan uno o varios microservicios para realizar la acción solicitada por el cliente.
- Manejo de Incidencias:** Esta funcionalidad permite almacenar un historial de todas las acciones que se ejecutan en el sistema, con el fin de tener un logs de acciones del sistema.

## Assumptions

## Requirements

Manager Microservicios
Administrador de Peticiones

1	<input type="checkbox"/> Peticiones	Se debe implementar una funcionalidad que permita administrar todas las funcionalidades que gestione las peticiones que realizan los clientes al sistema. También debe involucrar ejecutar los microservicios para realizar la acción requerida.
<b>Manejo de Logs</b>		
2	<input type="checkbox"/> Incidencias	Se debe desarrollar una funcionalidad que mantenga y almacene un historial de todas las incidencias o acciones que realiza un cliente en el sistema, con el fin de tener un log de acciones.

## Related documents

1. [Contexto del Proyecto](#)

## Questions

Below is a list of questions to be addressed as a result of this requirements document:

Question	Outcome

## Pending

## Not Doing

## B. Complementos de Evaluación

---

### B.1. Lista de Actividades de Usuarios

# Actividades Clientes

A continuación se presenta una lista de actividades a realizar para los usuarios de tipo cliente:

Lista de Actividades	
1	Pedir a un <b>partner</b> que registre a mi persona como su cliente en la plataforma.
2	Verificar mi cuenta de <b>correo electrónico</b> para obtener credenciales de acceso a la plataforma.
3	Acceder a la plataforma con las <b>credenciales</b> obtenidas.
4	Comprar un dominio, ingresa al menú " <b>Comprar</b> " en " <b>Mis dominios</b> ".
5	Verificar la disponibilidad del dominio " <b>haulmer.cl</b> ", ingresa el nombre del dominio sin el .CL
6	Verificar la disponibilidad del dominio " <b>Dell.cl</b> ", ingresa el nombre del dominio sin el .CL
7	Comprobar la disponibilidad de un dominio personalizado cuya lógica es la siguiente: " <i>Primera letra de tu nombre seguido de tu apellido( Por ej: Si tu nombre es Hernan Galvez, el nombre de dominio sería <b>hgalvez</b>)</i> "
8	Comprar el dominio creado anteriormente saltando al siguiente paso con el botón " <b>Comprar</b> ". si no está disponible agrega al final del dominio tu año de nacimiento( por ej: hgalvez1992).
9	Llenar el campo " <b>Periodo de Registro</b> " con <b>2</b> años.
10	Llenar el campo <b>NameServer</b> siguiendo la siguiente lógica: " <i>si el nombre de dominio es "hgalvez", entonces el nameserver completalo como: <b>ns.hgalvez.cl</b></i> "
11	Marcar la casilla " <b>Si</b> " en la opción <b>Contacto Administrador</b> , para que tú quedes registrado como contacto administrativo en el dominio a registrar.
12	Marcar la casilla " <b>Si</b> " en la opción <b>Contacto Financiero</b> , para que tú quedes registrado como contacto financiero en el dominio a registrar.
13	Marca la casilla " <b>Si</b> " en la opción <b>Contacto Técnico</b> , para que tú quedes registrado como contacto técnico en el dominio a registrar. Finalmente presionar al botón " <b>siguiente</b> ".

14	Verificar que la información esté correcta y <b>finalizar</b> la solicitud de compra presionando el botón " <i>Finalizar Compra</i> ".
15	Verificar que la orden de <b>compra</b> está registrada con estado " <b>Pendiente</b> "
16	Verificar la disponibilidad del dominio " <b>Monkey.cl</b> ", ingresa el nombre del dominio sin el .CL
17	Comprobar la disponibilidad del siguiente dominio cuya lógica es la siguiente: " <i>Tu apellido seguido de la palabra "store"</i> ( <i>Por ej: Si tu apellido es Galvez, el nombre de dominio sería galvezstore</i> )"
18	Proceder a comprar saltando al siguiente paso con el botón " <b>Comprar</b> "
19	Llenar el campo " <b>Periodo de Registro</b> " con 1 año
20	Llenar el campo NameServer cuya siguiente lógica es: " <i>si el nombre de dominio es "galvezstore", entonces el nameserver completalo como ns.hgalvez.cl</i> "
21	Marcar la casilla " <b>Si</b> " en la opción Contacto Administrador, para que tú quedes registrado como contacto administrativo en el dominio a registrar.
22	Marcar la casilla " <b>No</b> " en la opción Contacto Financiero para crear un nuevo contacto.
23	Llenar el formulario de contacto financiero con los datos solicitados para registrar al nuevo contacto.
24	Marca la casilla " <b>Si</b> " en la opción Contacto Técnico, para que tú quedes registrado como contacto administrativo en el dominio a registrar. Finalmente presiona al botón siguiente.
25	Verificar que la información esté correcta y <b>finalizar</b> la solicitud de <b>compra</b> presionando el botón " <i>Finalizar Compra</i> ".
26	Verificar que la <b>orden de compra</b> está registrada con estado " <b>Pendiente</b> "
27	Esperar que las solicitudes de registro de dominio estén resueltas
28	Luego que las solicitudes estén resueltas, verificar el estado de las órdenes pasen a estado " <b>Activa</b> "
29	Luego que los dominios fueron registrados, procede a renovar su periodo de caducidad. Para esto, dirígete a pestaña " <i>Mis Dominio</i> " y luego pincha en " <i>Ver todos</i> ". Se desplegará una tabla con tus dominios registrados.
30	Para renovar, pincha en el botón "Renovar" perteneciente a la fila del primer dominio registrado. Finalmente ingresa como periodo de renovación <b>dos años</b> y presiona el botón " <i>Si, renovar</i> ".

<b>31</b>	Realiza el procedimiento anterior con el segundo dominio registrado, pero en el periodo de renovación ingresa <b>tres años</b> .
<b>32</b>	Verifica que los dominios tengan la nueva fecha de renovación en la sección donde se encuentran tus dominios consultada previamente.
<b>33</b>	Dirigirse a la página principal de la plataforma pinchando en el nombre "Partner Site"( Esquina superior izquierda)
<b>34</b>	Verificar en el servicio <b>Whois</b> ubicado en la parte inferior de la página la información de los dos dominio solicitados previamente, escribiendo en el cuadro de texto el nombre de dominios añadiendo al final ".cl"( ej: "hgalvez.cl"), finalmente presiona el botón "buscar".
<b>35</b>	Cierra tu sesión en la plataforma

# Actividades Partners

A continuación se presenta una lista de actividades a realizar para los usuarios de tipo partner:

Lista de Actividades	
1	Registrarse como un partner en la plataforma.
2	Revisar tu correo electrónico para obtener la contraseña de tu cuenta.
3	Ingresar al sistema con las credenciales enviadas a tu correo electrónico.
4	Cargar \$5.000 a tu cuenta personal.
5	Verifica que la transacción se haya creado correctamente.
6	Crear un usuario cliente #1
7	Verificar que el cliente se creó correctamente en el menú "Ver Todos"
8	Esperar que el cliente acceda a la plataforma y compré un dominio.
9	Posteriormente, verificar que la transacción se realizó por la compra.
10	Verificar los fondos disponibles en el menú "Resumen".
11	Crear un usuario cliente #2
12	Verificar que el cliente se creó correctamente en el menú "Ver Todos"
13	Esperar que el cliente acceda a la plataforma y compré un dominio.
14	Posteriormente, verificar que la transacción se realizó por la compra.
15	Verificar los fondos disponibles en el menú "Resumen".
16	Crear un usuario cliente #3
17	Verificar que el cliente se creó correctamente en el menú "Ver Todos"
18	Esperar que el cliente acceda a la plataforma y compré un dominio.
19	Posteriormente, verificar que la transacción se realizó por la compra.
20	Verificar los fondos disponibles en el menú "Resumen".
21	Cargar \$3.000 a tu cuenta personal.
22	Verificar los fondos disponibles en el menú "Resumen".

<b>23</b>	Crear un usuario cliente #4
<b>24</b>	Verificar que el cliente se creó correctamente en el menú " <i>Ver Todos</i> ".
<b>25</b>	Esperar que el cliente acceda a la plataforma y compré un dominio.
<b>26</b>	Posteriormente, verificar que la transacción se realizó por la compra.
<b>27</b>	Verificar los fondos disponibles en el menú " <i>Resumen</i> ".
<b>28</b>	Crear un usuario cliente #5
<b>29</b>	Verificar que el cliente se creó correctamente en el menú " <i>Ver Todos</i> ".
<b>30</b>	Esperar que el cliente acceda a la plataforma y compré un dominio.
<b>31</b>	Posteriormente, verificar que la transacción se realizó por la compra.
<b>32</b>	Verificar que los dominios registrados se encuentren en el sistema( Ver en el menú " <i>Resumen</i> ")

## **B.2. Encuesta Realizada**

# Encuesta de evaluación de la plataforma Partner Site para la gestión y reventa de nombre de dominios .CL

Se ha desarrollado una plataforma que permite la compra y reventa de nombre de dominios .CL, además permite la gestión de usuarios que involucran el negocio, como lo son los usuarios que revenden( Partners) y los usuarios que compran( Clientes). Nos encantaría conocer tu opinión sobre cómo podemos mejorar tu experiencia.

\* Required

*Skip to question 1.*

## Datos de usuario

### 1. Tipo de usuario \*

*Mark only one oval.*

- Partner *After the last question in this section, skip to question 4.*
- Cliente *After the last question in this section, skip to question 5.*

### 2. ¿Qué Profesión tienes? \*

---

### 3. ¿Cuál es tu Género? \*

*Mark only one oval.*

- Masculino
- Femenino
- Prefiero no decirlo

*Skip to question 4.*

## Evaluación de funcionalidades del Partner

En esta sección, debe seleccionar el grado de acuerdo o desacuerdo referente a la utilización de la plataforma.

4. \*

*Mark only one oval per row.*

	Totalmente en desacuerdo	En desacuerdo	Ni en desacuerdo ni de acuerdo	De acuerdo	Totalmente de acuerdo
La plataforma permite registrarse como partner de manera exitosa.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
La plataforma crea los clientes de forma exitosa.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
La plataforma permite visualizar a los clientes registrados.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
La plataforma permite visualizar los nombres de dominios registrados por sus clientes.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
La plataforma permite renovar el periodo de validez de un nombre de dominio de manera exitosa.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
La plataforma permite ver las transacciones de manera exitosa.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
La plataforma carga los fondos de forma exitosa.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
La plataforma permite ver los fondos disponibles de su cuenta de manera exitosa.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

*Skip to question 7.*

## Evaluación de funcionalidades del Cliente

En esta sección, debe seleccionar el grado de acuerdo o desacuerdo referente a la utilización de la plataforma.

5. \*

*Mark only one oval per row.*

	Totalmente en desacuerdo	En desacuerdo	Ni en desacuerdo ni de acuerdo	De acuerdo	Totalmente de acuerdo
La plataforma permite verificar la disponibilidad de un nombre de dominio de manera exitosa.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
La plataforma permite compra un dominio de manera exitosa.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
La plataforma permite renovar el periodo de caducidad de un dominio de manera exitosa.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
La plataforma permite ver las ordenes generadas por compras realizadas de manera exitosa.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
La plataforma permite crear contactos para añadirlos a los dominios de manera exitosa	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

*Skip to question 6.*

## Utilidad del sistema del área cliente

En esta sección, debe seleccionar el grado de acuerdo o desacuerdo referente a la utilización de la plataforma.

6. \*

*Mark only one oval per row.*

	Totalmente en desacuerdo	En desacuerdo	Ni en desacuerdo ni de acuerdo	De acuerdo	Totalmente de acuerdo
La plataforma envía las credenciales de acceso vía email de tu cuenta creada de manera exitosa.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
La plataforma permite utilizar el servicio "whois" sobre un dominio registrado de manera exitosa.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

*Skip to question 8.*

## Utilidad del sistema del área partner

En esta sección, debe seleccionar el grado de acuerdo o desacuerdo referente a la utilización de la plataforma.

7. \*

Mark only one oval per row.

	Totalmente en desacuerdo	En desacuerdo	Ni en desacuerdo ni de acuerdo	De acuerdo	Totalmente de acuerdo
La plataforma envía las credenciales de acceso vía email de tu cuenta creada de manera exitosa.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
La plataforma envía las credenciales de acceso vía email de una nueva cuenta de cliente creada de manera exitosa.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
La plataforma permite utilizar el servicio "whois" sobre un dominio registrado de manera exitosa.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Skip to question 11.

## Preguntas para Clientes

En esta sección, se solicita responder las siguientes preguntas.

8. ¿La plataforma presentó algún error al momento de ser utilizado? \*

Mark only one oval.

- Si
- No

9. ¿Utilizaría la plataforma de Partner para comprar nombre de dominios .CL ? \*

---

10. ¿Qué cambios o mejoras le haría usted a la plataforma? \*

---



---



---



---



---

Skip to question 14.

## Preguntas para Partners

En esta sección, se solicita responder las siguientes preguntas.

**11. ¿La plataforma presentó algún error al momento de ser utilizado? \****Mark only one oval.* Si No**12. ¿Utilizaría la plataforma de Partner para revender nombre de dominios .CL a sus clientes? \***

---

**13. ¿Qué cambios o mejoras le haría usted a la plataforma? \***

---

---

---

---

---

*Skip to question 14.***Evaluación general de PartnerSite**

En esta sección, debe seleccionar el grado de deacuerdo o desacuerdo referente a la apreciación general de PartnerSite.

**14. Cómo le ha parecido PartnerSite respecto a: \****Mark only one oval per row.*

	Totalmente en desacuerdo	En desacuerdo	Ni en desacuerdo ni de acuerdo	De acuerdo	Totalmente de acuerdo
Funcionalidad en general de servicios.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Utilidad en general de servicios.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Powered by

