



**Universidad de las Ciencias  
Informáticas**



**Empresa de Tecnologías de la  
Información para la Defensa**

**Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas**

# **Sistema de gestión de productos y servicios para proveedores y consumidores empresariales utilizando la arquitectura de microservicios.**

**Autor:** Roberto Camejo Domínguez.

**Tutores:** MSc. Darling Darías Pérez.

Ing. Andrés Sellés González.

**La Habana, junio de 2019**

**Año 61 de la Revolución**

## **DECLARACIÓN DE AUTORÍA:**

Declaro ser el único autor de la tesis “Sistema de gestión de productos y servicios para proveedores y consumidores empresariales utilizando la arquitectura de microservicios”, realizado con la colaboración de la Universidad de las Ciencias Informáticas(UCI) y la Empresa de Tecnologías de la Información para la Defensa (XETID), a las cuales le reconozco los derechos patrimoniales de la investigación, con carácter exclusivo.

Para que así conste se firma la presente a los \_\_\_\_\_ días del mes junio de 2019.

---

Roberto Camejo Domínguez

**Autor**

---

MSc. Darling Darías Pérez.

**Tutor**

---

Ing. Andrés Sellés González.

**Tutor**

## **Agradecimientos**

**Sin duda las primeras personas que vienen a mi mente son mis padres Belkis y Luis por haberme apoyado durante este largo trayecto en busca de lograr mi sueño. Por darme su amor y cariño en los buenos momentos y en las más difíciles circunstancias, por su educación, por sus valores y sus consejos.**

**A mis abuelos mima y pipo, que son lo espiritual que hay detrás de todo lo que hago.**

**A mi gordo, por estar ahí más veces de las que puedo recordar ofreciéndome su amistad incondicional y a Franquito, el tercer mosquetero, ya son 12 años.**

**A mi hermano, por ser un ejemplo a seguir.**

**A Manolo, que más que vecino es un amigo, por toda la ayuda que me ha brindado, sin la cual esta investigación hubiese estado llena de obstáculos.**

**A Yanet, por ser la calma de mi madre y a su vez la calma mía.**

**A Patricia, por su comprensión y apoyo desde que la conocí.**

**A Mauri y Karen, por haberme apoyado, ayudado y por haberme permitido formar parte de sus vidas.**

**A Alejandro, Chang, Daniela, Julio, Suny, William, Jorge Ernesto, Henry, Michel, Jorgito, los integrantes de la Canoa y en general mi grupo 2501 y compañeros del 2502 por todos esos momentos que hemos compartido.**

**A mis tutores, por su ayuda a pesar de todos mis tropiezos.**

**Al tribunal, por su ayuda, críticas constructivas y consejos que hicieron posible el desarrollo de esta investigación.**

**A todos mis profesores, por haberme formado como lo que hoy aspiro a oficializar.**

**A todas esas personas, que no mencioné, pero, que saben que siempre formarán parte de mi vida.**

**Gracias.**

## *Dedicatoria*

*A mi madre por todo su esfuerzo, sacrificio y abnegación.*

*A ese hombre que no necesito mencionar pues no comparto su  
sangre, pero sí sus valores,*

*A mi gordo y amigo del alma.*

## Resumen

Muchas veces las empresas no están conscientes de la disponibilidad de productos a la hora de adquirirlos de otras empresas. Es un cúmulo de información que no está organizada, dificulta la actualización y el estado de las mismas generando desorden, errores, pérdidas, entre otros aspectos.

En la presente investigación se propone un Sistema web de gestión de productos y servicios entre proveedores y consumidores empresariales en Cuba utilizando la arquitectura de microservicios. Este permite que las cuentas registradas, antes de darle acceso al sistema, sean verificadas por un administrador. Una vez autenticado, en calidad de proveedor, admite la publicación de productos y de servicios, así como la posibilidad de recibir notificaciones de las reservas realizadas a los productos y servicios. Al acceder como consumidor puede realizar las reservas de estos productos y servicios publicados, otorgándoles una evaluación.

La investigación está regida por la metodología de desarrollo Extreme Programming (XP). Se empleó el lenguaje JavaScript y HTML5 para su implementación. Se utilizó los frameworks NodeJs, Express, MongoDB con su gestor Robo 3t y Postman en las consultas a la API para el caso de la parte trasera del sistema o backend, y AngularJS para la parte visual o frontend. Como herramientas de prueba se emplean Jasmine y Karma para probar el frontend y Mocha para el backend, las cuales permitieron conocer los errores arrojados por la interfaz y la velocidad de respuesta de las consultas a las APIs.

**Palabras claves:** comercio electrónico, consumidor, proceso, producto, proveedor, reserva, servicio, sistema.

## **Abstract**

In this research, a system to interrelate providers and customers at a business level called SICOVA is proposed, using the microservice architecture. This allows the accounts registered, before giving access to the system, to be verified by an administrator and, once authenticated, as a provider, allows the publication of products and services, as well as the possibility of receiving notifications of reservations made to the products and services; and when accessing as a consumer it allows to make the reservations of these published products and services and the possibility of granting them an evaluation.

This work is governed by the Extreme Programming (XP) development methodology. JavaScript and HTML5 were used, the NodeJs, Express, MongoDB frameworks with their manager Robo3t and Postman were used in the API queries for the case of the back of the system or backend and AngularJS for the visual or frontend part. As test tools Jasmine and Karma are used to test the frontend and Mocha for the backend, which allowed to know the errors thrown by the interface and the speed of response of the queries to the APIs.

**Keywords:** trade electronic, customer, provider, process, reservation, system, service, supplier.

## Índice

Introducción	12
CAPITULO I Fundamentación teórica de gestión de productos y servicios entre proveedores y consumidores empresariales	16
1.1 Conceptos asociados	16
1.2 Análisis de soluciones similares	17
1.2.1 Internacionales	17
1.2.2 Nacionales	18
1.2.3 Resultados obtenidos por el estudio	20
1.3 Metodología, lenguajes, herramientas y tecnologías empleadas en el desarrollo	20
1.3.1 Metodologías de desarrollo	20
1.3.2 Lenguajes y frameworks utilizados	21
1.3.3 Entornos de desarrollo	22
1.3.4 Sistemas gestores de bases de datos	24
1.3.5 Herramientas case	24
1.3.6 Herramientas de pruebas del sistema	24
1.3.7 Conclusiones parciales	24
CAPITULO II Análisis y diseño del sistema	26
2.1.1 Modelo conceptual	27
2.1.2 Descripción de clases del modelo del dominio.	27
2.2 Roles y responsabilidad.	28
2.3 Fase de planeación.	29
2.3.1 Disciplina de requisitos.	29
2.3.2 Historias de usuario.	30
2.3.3 Requisitos no funcionales.	34
2.4 FASE DE DISEÑO	36
2.4.1 Arquitectura de software.	36
2.4.2 Tarjeta CRC.	37
2.5 Representación de los datos	39
2.6 Patrones GRASP	40
2.7 Conclusiones parciales.	42
CAPITULO 3 Implementación y pruebas del software	43
3.1 Tarea de ingeniería	43
3.2 Estrategia de pruebas	49



3.2.1 Pruebas unitarias	50
3.2.2 Pruebas de aceptación	53
3.3 Conclusiones parciales	61
Conclusiones generales	62
Recomendaciones	63
Referencias bibliográficas	64
Bibliografía	66
Anexos	68

## Índice de tablas

Tabla 1 HU_1 Gestionar producto.....	31
Tabla 2 HU_2 Gestionar servicio .....	31
Tabla 3 HU_3 Autorizar registro.....	32
Tabla 4 HU_4 Reservar producto.....	32
Tabla 5 HU_5 Reservar servicio.....	32
Tabla 6 HU_6 Evaluar producto .....	33
Tabla 7 HU_7 Evaluar servicio .....	33
Tabla 8 HU_8 Notificar reserva .....	33
Tabla 9 HU_9 Gestionar Usuario.....	33
Tabla 10 HU_10 Autenticar usuario .....	34
Tabla 11 Tarjeta CRC Usuario.....	37
Tabla 12 Tarjeta CRC Producto.....	37
Tabla 13 Tarjeta CRC Servicio .....	38
Tabla 14 Tarea 1 Insertar Producto.....	43
Tabla 15 Tarea 2 Modificar Producto.....	43
Tabla 16 Tarea 3 Listar Productos.....	44
Tabla 17 Tarea 4 Eliminar Producto.....	44
Tabla 18 Tarea 5 Insertar Servicio .....	44
Tabla 19 Tarea 6 Modificar Servicio.....	45
Tabla 20 Tarea 7 Listar Servicios.....	45
Tabla 21 Tarea 8 Eliminar Servicio .....	45
Tabla 22 Tarea 9 Autorizar registro .....	46
Tabla 23 Tarea 10 Requerir Reserva de Producto .....	46
Tabla 24 Tarea 11 Requerir Alquiler de Servicio.....	46
Tabla 25 Tarea 12 Evaluar Producto.....	47
Tabla 26 Tarea 13 Evaluar Servicio.....	47
Tabla 27 Tarea 14 Notificar reserva .....	47
Tabla 28 Tarea 15 Registrar usuario.....	48
Tabla 29 Tarea 16 Modificar Usuario .....	48
Tabla 30 Tarea 17 Listar usuario.....	48
Tabla 31 Tarea 18 Eliminar usuario .....	48
Tabla 32 Tarea 19 Autenticar Usuario .....	49
Tabla 33 Prueba Registrar Usuario.....	53
Tabla 34 Prueba Modificar Usuario.....	53
Tabla 35 Prueba Listar Usuario.....	54
Tabla 36 Prueba Eliminar Usuario .....	54
Tabla 37 Prueba Autenticar Usuario.....	55
Tabla 38 Prueba Registrar Producto .....	55
Tabla 39 Prueba Modificar Producto .....	55
Tabla 40 Prueba Listar Producto .....	56
Tabla 41 Prueba Eliminar Producto.....	56
Tabla 42 Prueba Registrar Servicio .....	57
Tabla 43 Prueba Modificar Usuario.....	57
Tabla 44 Prueba Listar Servicio .....	58
Tabla 45 Prueba Eliminar Servicio.....	58
Tabla 46 Prueba Autorizar Registro.....	58

<b>Tabla 47 Prueba Requerir Compra de Producto .....</b>	<b>59</b>
<b>Tabla 48 Prueba Requerir Alquiler de Servicio .....</b>	<b>59</b>
<b>Tabla 49 Prueba Evaluar Producto .....</b>	<b>60</b>
<b>Tabla 50 Prueba Notificar reserva.....</b>	<b>60</b>

## Introducción

En el mundo se impulsa cada vez más, el desarrollo del E-commerce o comercio electrónico. En la actualidad se ha convertido en una herramienta con gran éxito para el mundo de los negocios gracias a la apertura y facilidad de acceso al Internet.

En tal sentido, adquirir un artículo o pagar un servicio del hogar desde la comodidad de un teléfono celular o una pc para el comercio electrónico, cobran cada vez mayor auge y aceptación en el mundo: Son reflejos precisamente, de lo que puede hacerse para el bienestar de las personas.

Es importante conocer que el comercio electrónico funciona como una tienda en línea. En esta tienda tanto las personas naturales como las entidades pueden comprar o vender bienes y servicios utilizando medios electrónicos como redes sociales y páginas web.

Las transacciones, si de comercio electrónico se refiere, pueden ser de diferentes formas. Lo mismo, de empresas a consumidores (B&C), de empresas a empresas(B&B) o de plataformas donde los consumidores compran y venden entre ellos asumiendo, en un solo usuario tanto el rol de proveedores como de consumidores(C&C). (Ruiz, 2016)

En Cuba, el comercio electrónico es un ámbito fundamental que contempla la política de informatización. (Díaz-Canel, 2019)

En el país los procesos que intervienen con proveedores y consumidores, a nivel empresarial, como la reserva, el almacenamiento y calificación de los productos y servicios brindados, generalmente, tienen como características ser lentos, realizarse manualmente y que el cliente no posea control de la ejecución de los trámites sin depender de algún funcionario que lo gestione. La burocracia, el exceso de pasos y las demoras, tienen como resultado la inconformidad y molestias de los clientes. Además, ocurren retrasos debido a la forma en que se ejecuta el proceso actualmente.

La colaboración de grandes entidades como el Banco Nacional de Cuba, el Banco Popular de Ahorros, la empresa de telecomunicaciones ETECSA, la Empresa de Correos de Cuba, las empresas de desarrollo de software y el Gobierno Cubano están enfocadas en impulsar proyectos para incrementando el desarrollo del comercio electrónico en Cuba. Dirigidos inicialmente a la relación entre empresas comercializadoras, de productos y servicios, y la población.

La gestión de dicho proceso se realiza, en su mayoría, de forma manual, y muchas veces las empresas no están conscientes de la disponibilidad de productos a la hora de adquirirlos de otras empresas. Es un cúmulo de información que no está debidamente organizada, dificulta la

actualización y el estado de las mismas generando desorden, errores, pérdidas, entre otros aspectos.

Actualmente, al solicitar un servicio no se dispone de la información necesaria, como la calidad y los plazos de disponibilidad, propiciando tardanzas en la respuesta e inconformidad en las prestaciones.

Por lo antes expuesto se identifica el siguiente **problema a resolver**: ¿Cómo facilitar la gestión de productos y servicios entre proveedores y consumidores empresariales en Cuba?

La presente investigación tiene como **objeto de estudio**: Procesos de compra-venta de productos y servicios entre consumidores y proveedores a nivel empresarial.

Por lo que se define como **objetivo general**: Desarrollar un sistema que facilite la gestión de productos y servicios entre consumidores y proveedores empresariales, empleando la arquitectura de microsistemas.

El **campo de acción** se centra en: la gestión de servicios y productos entre consumidores y proveedores en las empresas estatales en Cuba.

Para dar cumplimiento al objetivo general se definen los siguientes **objetivos específicos**:

1. Elaborar el marco teórico conceptual de la investigación.
2. Realizar el análisis y diseño de la propuesta de solución.
3. Implementar los elementos de servicios, productos, clientes y evaluación, para los consumidores y/o proveedores empresariales.
4. Realizar pruebas unitarias y de aceptación al sistema desarrollado.
5. Validar el resultado obtenido.

Con el fin de dar cumplimiento a los objetivos propuestos en la investigación se tomaron como **tareas científicas**:

1. Elaboración de los referentes teóricos y metodológicos de la investigación, asociada a los procesos de gestión de la interrelación entre proveedores y consumidores a nivel empresarial.
2. Análisis de sistemas que intervienen en los procesos entre proveedores y consumidores empresariales a nivel nacional e internacional para sentar las bases de la investigación.
3. Breve explicación de las herramientas, tecnologías y modelo de desarrollo de software establecido, para lograr un entendimiento de estas.

4. Identificación de las Historias de usuario y de los requisitos no funcionales establecidos por la metodología, para plasmar las funcionalidades que debe cumplir la propuesta de solución, así como dejar precisas las condiciones bajo las cuales debe operar.
5. Diseño de la propuesta de solución teniendo en cuenta la metodología de desarrollo de software determinada.
6. Implementación de un sistema para contribuir a la gestión de servicios y productos entre proveedores y consumidores empresariales en Cuba.
7. Realización de pruebas para validar el código y el correcto funcionamiento de la aplicación, teniendo en cuenta la satisfacción del cliente.

Para la realización de la investigación se utilizaron los siguientes **métodos de investigación**:

#### **Métodos teóricos:**

- **Analítico-sintético:** El empleo de este método se evidencia en el análisis de la documentación del desarrollo del comercio electrónico en el país y en los procesos de compra-venta de productos y servicios, este permite sintetizar la información necesaria para el desarrollo de un sistema que contribuya a facilitar la relación entre empresas en calidad de consumidores y proveedores.
- **Histórico-lógico:** Utilizado al tener en cuenta la caracterización de la evolución histórica de los sistemas de compra-venta de productos y servicios entre consumidores y proveedores empresariales en Cuba y el mundo, como base para el desarrollo del sistema actual.

#### **Métodos empíricos:**

- **Medición:** el sistema obtenido como resultado de la investigación será medido a través de las pruebas unitarias y de validación garantizando el correcto funcionamiento del mismo.
- **Modelación:** este método permite la creación de modelos, (propuestas, alternativas, etc.). En esta investigación a través del método empírico se realiza el modelado mediante diagramas. Esto permite reflejar la estructura, las relaciones y características de la solución propuesta.

El documento consta de tres capítulos en los que se abordan diferentes temáticas. Se estructuran de la siguiente forma:

**Capítulo 1. Fundamentación teórica:** En este capítulo se elabora el marco teórico conceptual de la investigación para un mejor entendimiento de la problemática. Se realiza un análisis de las soluciones existentes sobre la gestión de productos y servicios entre consumidores y proveedores

empresariales. Se caracteriza la metodología de desarrollo XP para tener en cuenta los artefactos a generar en cada fase. Además, se describen las arquitecturas de software definidas en el desarrollo de la solución propuesta. Se explican las herramientas y tecnologías definidas para el desarrollo de la propuesta de solución.

**Capítulo 2. Análisis y diseño:** En este capítulo se describen inicialmente la propuesta de solución del sistema web a desarrollar y los procesos que serán informatizados, así como las historias de usuario especificadas por el cliente y los requisitos no funcionales que debe cumplir el sistema de software a desarrollar. Finalmente se generan los artefactos correspondientes según la metodología de desarrollo de software empleada.

**Capítulo 3. Implementación y Validación:** En este capítulo se abordan los aspectos relacionados con la construcción de la aplicación, dentro del cual se destacan los procesos de implementación y el de prueba para su validación.

# **CAPITULO I Fundamentación teórica de gestión de productos y servicios entre proveedores y consumidores empresariales**

En el presente capítulo se realiza un estudio de temas relacionados con la conceptualización de consumidores y proveedores. Así como los conceptos asociados al objeto de estudio y al campo de acción y otros sistemas a fin de realizar una investigación más detallada. Además, se analizan y explican las herramientas, tecnologías, lenguajes y metodología de software a utilizar para el desarrollo de la propuesta de solución.

## **1.1 Conceptos asociados**

### **Consumidores:**

Según la Real Academia de la Lengua Española consumidor es aquel que consume bienes y productos en una sociedad de mercado. (RAE-ASALE, 2019)

Se define, para la investigación, como **consumidores** a las empresas que, como personas jurídicas, son responsables de adquirir, reservar y evaluar productos y servicios.

### **Proveedores:**

Según la Real Academia de la Lengua Española proveedor es una persona o una empresa que provee o abastece todo lo necesario para un fin a grandes grupos, asociaciones, comunidades, etc. (RAE-ASALE, 2019)

Se define, para la investigación, como **proveedores** a las empresas que, en calidad de personas jurídicas, ofertan y suministran productos o prestan servicios de cualquier naturaleza a los consumidores.

### **Relación entre consumidores y proveedores:**

Se define como los procesos que hacen a uno depender del otro. La relación entre un consumidor y un proveedor está dada por la oferta, demanda, reservación, comercialización y adquisición.

### **REST (Estado representacional de transferencia):**

Un servicio REST se define como una agregación de diferentes recursos que pueden ser alcanzados desde un identificador universal de recurso (URI) base. Un recurso representa a una entidad del mundo real cuyo estado está expuesto y puede cambiarse accediendo a un URI. Una Representación es la descripción de los mensajes enviados o recibidos de un Recurso en términos de un lenguaje tecnológico. Actualmente XML y JSON son los idiomas más populares para describir estos mensajes. (González, 2015)



## **API (Interfaz de Programación de Aplicaciones)**

Una API es un conjunto de funciones, procedimientos y subrutinas que ofrece una “biblioteca” para ser utilizadas por otro software. API es un conjunto de acciones que dan acceso a determinadas tareas de un software, como pueden ser tareas de creación, actualización o borrado de elementos. Como se percibe en su propia descripción, las APIs se utilizan para hacer uso de funciones existentes en otro software. A su vez, distintas aplicaciones pueden hacer uso de las API de cada una para mantener una comunicación de datos entre ellas, de manera transparente para el usuario (Mattson, 2010)

### **NoSQL o no relacional:**

NoSQL (a veces llamado "no sólo SQL") es una amplia clase de sistemas de gestión de bases de datos que difieren del modelo clásico de SGBDR (Sistema de Gestión de Bases de Datos Relacionales) en aspectos importantes, siendo el más destacado que no usan SQL como lenguaje principal de consultas (Strozzi,2007). Los datos almacenados no requieren estructuras fijas como tablas, normalmente no soportan operaciones JOIN, ni garantizan completamente ACID (atomicidad, consistencia, aislamiento y durabilidad) y habitualmente escalan bien horizontalmente. Los sistemas NoSQL se denominan a veces "no sólo SQL" para subrayar el hecho de que también pueden soportar lenguajes de consulta de tipo SQL (Antiñanco, 2017)

## **1.2 Análisis de soluciones similares**

A continuación, se realiza el análisis de los sistemas de compra-venta de servicios y productos, tanto internacionales como nacionales, que están enfocados en el desarrollo del comercio electrónico y a su vez, algunos, en el empleo de la arquitectura de microservicios.

### **1.2.1 Internacionales**

#### **Amazon**

Amazon, es una empresa de comercio electrónico y servicios de computación en la nube a todos los niveles con sede en la ciudad estadounidense de Seattle, Estado de Washington. Su lema es: From A to Z (traducido al español: «De la A a la Z»). El sistema de ventas de Amazon, consultable a través de la dirección amazon.com, ofrece la posibilidad de informarse de productos de cualquier denominación incluyendo un sistema de pagos electrónicos mediante PayPal. Así como la posibilidad de entregas a domicilio en varios países del globo y también ofrece servicios de asistencia técnica remota y de streaming. Hace unos años la plataforma migró de la arquitectura monolítica a la arquitectura de microservicios pues desarrollaron una API que ofrece la consulta o almacenamiento de productos. (A, Essau, 2016)

Amazon no permite la publicación de cualquier tipo de servicio, solo se pueden ver los que la propia empresa ofrece y no se le da a conocer al consumidor la disponibilidad de los mismos.

## **eBay**

eBay es un sitio destinado a la subasta y comercio electrónico de productos a través de Internet. Desde el año 1995 permite la comercialización de productos. Este sitio realmente es un sistema de intermediación automático, en el que los usuarios pueden calificar al otro usuario mediante un sistema de puntos positivos o negativos, dependiendo del éxito de la operación. La compañía es pionera en la adopción de tecnologías como Docker y el empleo de microservicios. Su aplicación principal comprende varios servicios autónomos, y cada uno ejecuta la lógica propia de cada área funcional que se ofrece a los clientes. (Nielsen, 2016)

eBay no es un sitio destinado a la venta mayorista de productos, más bien permite que un usuario coloque en venta sus productos y le da la posibilidad a otros que pujen por su adquisición.

### **1.2.2 Nacionales**

#### **Ofertas**

El sitio Ofertas brinda una plataforma, exclusivamente para publicar clasificados y publicidad de un medio de comunicación cubano. Facilita la compra o venta de un objeto personal, una casa, un automóvil, un celular, una computadora, permutar una vivienda, encontrar un empleo o sencillamente promocionar un negocio.

La plataforma de clasificados permite que el usuario pueda conectarse al sitio desde cualquier red nacional, sin requerir de acceso pleno a Internet. Así mismo, pretende mantenerse actualizada diariamente con el objetivo de brindar un servicio constante a personas naturales y jurídicas de todo el país.

Ofertas es un sistema de comercio electrónico enfocado al desarrollo de la relación Consumidor-Consumidor(C&C), permite que cualquier persona se registre en el sistema sin importar su rol y no ofrece la posibilidad de calificar ningún producto o servicio, estos últimos, diferenciables solo por la categoría, pero de cara al cliente se crean y publican de la misma manera.

#### **Súper Fácil**

Brinda la posibilidad de acceder a servicios y productos cubanos en el territorio nacional en sus propias tiendas. (Montoya, 2019)

A precios asequibles en CUP, pueden adquirirse en línea de contenidos digitales cubanos de variadas temáticas, idiomas y formatos, libros electrónicos, audiolibros, videos, documentales, películas, juegos y multimedia.

La oferta incluye soluciones informáticas para negocios propios y su pago de servicios, previamente contratados a CITMATEL en sus oficinas en todo el país, donde próximamente se incorporarán nuevas tiendas, según los anuncios del sitio.

Para efectuar el pago debe contarse con una tarjeta magnética en CUP del Banco Metropolitano, Banco Popular de Ahorro o Banco de Crédito y Comercio y, además, debe tenerla asociada al servicio de Telebanca.

Súper Fácil es un sistema bastante inestable en cuanto a disponibilidad. Al acceder a su tienda de Gran Comercial las búsquedas de todos los productos, sin especificar ningún criterio, generan un error de página no encontrada o 404, que no permite ir de vuelta al buscador. Ante cualquier error siempre se redirige a la página principal obligando a realizar todo el recorrido una vez más. No ofrece la posibilidad de publicar productos ni servicios por otras empresas que no sean propios de CITMATEL.

### **5ta y 42**

Luego de un período de pruebas que se extendió desde el verano pasado, cadena de tiendas Caribe publicó el 12 de diciembre de manera oficial el servicio de ventas a la población mediante el comercio electrónico en CUP que posibilita la compra de algunos de sus productos a través de internet.

La primera fase solo estará disponible en el centro comercial de 5ta y 42 en La Habana, aunque se prevé extenderlo, al menos, a una tienda en cada provincia durante el primer semestre del 2019. (Fernández, 2019)

El sitio permite la autenticación de usuarios para la realización de las compras, pero no permite la publicación de ofertas de ninguna procedencia que no sea la de la propia tienda.

Varias veces el sistema se ha encontrado sin funcionar por reparaciones, lo que denota una total dependencia entre todos sus componentes. Permite solo la compra y evaluación de los productos ofertados por la cadena de tiendas Caribe. No se brindan servicios, ni se permite la inserción de los mismos, ni de los productos de otras empresas.

### **CILMO:**

CILMO es un Sitio Web de la empresa CEDIPAD la cual según el decreto No.315 es responsable del tratamiento y gestión de los inventarios de productos de lento movimiento y ociosos de las empresas estatales cubanas. En esta plataforma se publica información relacionada con dichas mercancías y se gestiona su comercialización de forma mayorista.

## **SITRANS:**

El sitio de servicios de información del transporte brinda un conjunto de servicios a otras empresas entre los que destacan la consultoría, la asistencia técnica, la información y la comercialización de efectos, materiales, equipos y muebles de oficina. SITRANS posee una tienda virtual accesible mediante la dirección [www.tienda.virtual.cu](http://www.tienda.virtual.cu). Esta tienda contiene un conjunto de categorías de productos como adhesivos, corte y embalaje, medios electrónicos, informáticos y consumibles, y encuadernación y archivo. Además, cuenta con información de los productos más destacados y un carrito de compra que permite el pago de sus ofertas mediante la pasarela de pago cubana de Telebanca o transfermóvil.

El sitio solo posee en oferta los servicios y productos de la propia empresa, impidiendo la posibilidad de otras empresas de asociarse a este medio de comercialización.

### **1.2.3 Resultados obtenidos por el estudio**

A partir de la síntesis de cada uno de los sistemas de compra y venta de productos, se puede apreciar, que en el país los sistemas existentes solo están enfocados a mejorar el comercio electrónico entre consumidores, entre empresas, y empresa y consumidor. Los sistemas de Súper fácil y 5ta y 42 no admiten la publicación de otro producto que no sean los comercializados por las empresas de CIMEX y los aprobados por el Ministerio de Comercio Interior (MINCIN). Ninguno de los sistemas estudiados representa de manera adecuada el modo de brindar un servicio por un proveedor. No se diferencian los productos, excepto si el creador determina, como el caso de Ofertas, que su anuncio será publicado en el apartado de servicio. Esto provoca el desconocimiento, por parte de los consumidores, de la disponibilidad del servicio que van a requerir. Se tomará en cuenta la forma de representación de los productos del sitio 5ta y 42, la que permite que los consumidores conozcan de la disponibilidad del producto y a la vez su evaluación.

## **1.3 Metodología, lenguajes, herramientas y tecnologías empleadas en el desarrollo**

Las metodologías, lenguajes y herramientas son un conjunto de técnicas y recursos de software utilizados para el desarrollo de aplicaciones, haciendo posible el trabajo del equipo. A continuación, se brindan características de la metodología, los lenguajes y las herramientas de desarrollo estudiadas para emplear en cada una de las fases del proyecto.

### **1.3.1 Metodologías de desarrollo**

En este epígrafe se hace un breve análisis de la metodología **XP** (eXtreme Programming). Según Kent Beck, es una metodología ligera, eficiente y flexible para desarrollar software. Se basa en la simplicidad, la comunicación y la realimentación o reutilización del código desarrollado. Tiene

como objetivo la disminución del costo, la satisfacción del cliente y potenciar al máximo el trabajo en grupo (Joskowicz, 2008). Contiene cuatro fases: Análisis, Diseño, Implementación y Pruebas.

XP es uno de los procesos ágiles más destacados que pone énfasis en la adaptabilidad que en la previsibilidad. Los jefes de proyecto, los desarrolladores y el cliente son parte del equipo de desarrollo. Los desarrolladores deben conocer cada cambio que se realice durante el desarrollo del proyecto. (Pressman, 2010)

La metodología XP fue seleccionada para guiar el proceso ingenieril, porque facilita la documentación más discreta y propicia mayor dinamismo para el desarrollo, pues la codificación debe irse auto documentando. Para ello, se logra eligiendo adecuadamente los nombres de las variables, clases y métodos. Es apropiada para proyectos pequeños y entregas con ciclos rápidos, donde clientes y desarrolladores trabajan en conjunto. Fortalece el trabajo en equipo, facilitando el seguimiento a las tareas a partir de la inserción de procedimientos ágiles. La relación entre el cliente y el equipo de desarrollo evita errores del sistema por mala definición de los requisitos.

### **1.3.2 Lenguajes y frameworks utilizados**

#### **JavaScript:**

JavaScript es el lenguaje de programación común más utilizado para añadir interactividad a una página web. Es decir, es un lenguaje interpretado no compilado pues no genera ningún tipo de fichero objeto. Es un lenguaje moderno, sencillo, muy útil, barato pues solo se necesita un bloc de notas y un navegador. Permite la Programación Orientada a Objeto (POO) y además es visual debido a que permite la moderna programación visual. (Bray, 2014)

En el sistema se usa una variante llamada TypeScript que permite la conformación de métodos asíncronos, donde su ejecución no dependa de la respuesta de otros métodos.

#### **HTML 5:**

HTML, más que el lenguaje de marcado predominante usado para estructurar el contenido web, es realmente una tecnología web que interrelaciona varios estándares. En esta última revisión del lenguaje se incluyen varias características verdaderamente útiles, como son nuevos elementos semánticos y mejoras en los formularios (Mackdonald, 2016). Se emplea como complemento del desarrollo de la parte visual del sistema WEB derivado de la investigación.

#### **NodeJS:**

Node.js es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor (pero no limitándose a ello) basado en el lenguaje de programación ECMAScript, asíncrono, con I/O de datos en una arquitectura orientada a eventos y basado en el motor V8 de Google. Fue creado con el enfoque de ser útil en la creación de programas de red altamente

escalables, como, por ejemplo, servidores web. Fue creado por Ryan Dahl en 2009 y su evolución está apadrinada por la empresa Joyent, que además tiene contratado a Dahl en plantilla. (Finley, 2014)

NodeJS permite que tanto la base de datos, junto a los métodos que se encargan de gestionar su información, y la parte frontal del sistema compartan un único lenguaje y así facilitar la velocidad de las consultas y los tiempos de respuesta.

### **Express:**

Express.js, o simplemente Express, es un marco de aplicación web para NodeJS, lanzado como software gratuito y de código abierto bajo la Licencia MIT. Está diseñado para construir aplicaciones web y APIs. Se le ha llamado el marco de servidor estándar de facto para NodeJS.

Es un módulo de NodeJS y como tal funciona sobre esta plataforma; este módulo ofrece los métodos suficientes en JavaScript, para poder manejar las solicitudes o peticiones que se hacen por medio de los métodos del protocolo HTTP (GET, POST, etc.). También ofrece un sistema simple de enrutamiento, que dentro del mean stack es aprovechado en el backend o lado del servidor.

### **AngularJS 7:**

AngularJS (comúnmente llamado Angular.js), es un framework de JavaScript de código abierto, mantenido por Google, que se utiliza para crear y mantener aplicaciones web de una sola página. Su objetivo es aumentar las aplicaciones basadas en navegador con capacidad de Modelo Vista Controlador (MVC), en un esfuerzo para hacer que el desarrollo y las pruebas sean más fáciles.

La biblioteca lee el HTML (HyperText Model Language) que contiene atributos de las etiquetas personalizadas adicionales, entonces obedece a las directivas de los atributos personalizados, y une las piezas de entrada o salida de la página a un modelo representado por las variables estándar de JavaScript. Los valores de las variables de JavaScript se pueden configurar manualmente, o recuperados de los recursos JSON estáticos o dinámicos.

AngularJS se puede combinar con el entorno en tiempo de ejecución NodeJS, el framework para servidor Express.js y la base de datos MongoDB para formar el conjunto MEAN. (AngularJS, 2019)

Se utiliza para conformar la parte visual o frontend de la aplicación a través del empleo de servicios, modelos y controladores. Permite representar, de cara al usuario, la información brindada por las API REST que posee el sistema.

### **1.3.3 Entornos de desarrollo**

### **Visual Studio Code:**

Visual Studio Code es un editor de código fuente desarrollado por Microsoft para Windows, Linux y MacOS. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código. También es personalizable, por lo que los usuarios pueden cambiar el tema del editor, los atajos de teclado y las preferencias. Es gratuito y de código abierto, aunque la descarga oficial está bajo software propietario.

Visual Studio Code se basa en Electron, un framework que se utiliza para implementar aplicaciones Node.js para el escritorio, que se ejecuta en el motor de diseño Blink. Aunque utiliza el framework Electron, el software no usa Atom y en su lugar emplea el mismo componente editor (Mónaco) utilizado en Visual Studio Team Services. (ed, 2017)

En la herramienta se codifica tanto la parte del frontend como del backend de la aplicación. Gracias a los complementos de intérprete de lenguaje TypeScript y de poseer una terminal para compilar cada uno de los servicios ofrecidos por las API REST y por la parte visual desarrollada en Angular. Es un excelente entorno de desarrollo para el trabajo con la tecnología MEAN (Mongodb, Express js, Angularjs y NodeJs).

### **Robo 3T:**

Robomongo o Robo 3T es una herramienta multiplataforma con la que se puede administrar gráficamente las bases de datos que, pese a no tener versión definitiva actualmente (beta 0.8.4), se está convirtiendo en un importante aliado para los usuarios de MongoDB. (Schetnikovich, 2019)

### **Postman:**

Se trata de una herramienta dirigida a desarrolladores web que permite realizar peticiones HTTP a cualquier API. Postman es muy útil a la hora de programar y hacer pruebas. Además, ofrece la posibilidad de comprobar el correcto funcionamiento de los desarrollos. (Cuervo, 2019)

Postman no es una herramienta exclusiva para profesionales del entorno web. Es muy útil para todo aquel que interactúa con una API.

Permite generar documentación acerca del proceso de consultas a la API para un mejor entendimiento a la hora de compartir el servicio con otros desarrolladores, brindando información como los parámetros requeridos por las rutas de las consultas y los parámetros de respuesta.

### **1.3.4 Sistemas gestores de bases de datos**

#### **MongoDB:**

MongoDB es el sistema de base de datos desarrollada en 10gen por Geir Magnusson y Dwight Merriman. Es una base de datos orientada a documentos JSON, salvo que está diseñada para ser una verdadera base de datos de objetos, más que para un almacenamiento de clave/valor puro.

Es una base de datos no relacional, es decir, no utiliza SQL. El nombre viene del término inglés “humongous” (colosal) y puede ser definida como una base de datos documental sin esquema, escalable y de alto rendimiento. Algunos especialistas la han catalogado como la “MySQL de las bases de datos NoSQL”, al ver que es una base de datos rapidísima, sencilla en la funcionalidad ofrecida y que, como hizo MySQL hace años, logra la atención de la comunidad de desarrolladores. (Camargo-Ortega, 2015)

### **1.3.5 Herramientas case**

#### **Visual Paradigm 8.0 Enterprise Edition:**

Es una herramienta CASE (Computer Aided Software Engineering) que soporta el modelado mediante UML (Unified Model Language) y proporciona asistencia a los analistas, ingenieros de software y desarrolladores durante todos los pasos del ciclo de vida de desarrollo de un software. Facilita el modelado de UML, proporciona herramientas específicas y contribuye a la estandarización de la aplicación. (Schmuller, 2013)

### **1.3.6 Herramientas de pruebas del sistema**

#### **Mocha:**

Mocha es una herramienta de pruebas para NodeJS, que permite ejecutar pruebas de manera síncrona, asíncrona y ordenada tanto en el lado del servidor como del navegador. Este framework permite, mediante la creación de un archivo “test.js” y el comando “ng test” en la consola de proyecto de Angular CLI, efectuar pruebas a la interfaz de la aplicación, que comprobarán el flujo de todo el código HTML. (Reshuán, 2018)

### **1.3.7 Conclusiones parciales**

Luego de finalizado el capítulo se concluye que:

- Durante la elaboración del marco teórico conceptual, se establecieron los principales referentes teóricos-metodológicos que sustentan la gestión de productos y servicios entre clientes y proveedores, permitiendo obtener conceptos e ideas para el apoyo de la propuesta de solución.



- Se realizó un estudio de soluciones existentes en el mundo que realizan el proceso compra y venta de productos y servicios, para sentar las bases en la elaboración de la propuesta de solución.
- Se explicó de forma breve la metodología, las herramientas y tecnologías a utilizar, lográndose un mejor entendimiento de su uso en el desarrollo del proyecto.

## **CAPITULO II Análisis y diseño del sistema**

En el presente capítulo con el objetivo de lograr una visión general del dominio que se desea informatizar, será enunciada y descrita la propuesta de solución. Por otra parte, se obtendrá un conjunto de artefactos correspondientes a la metodología ágil XP. También se describen las principales características que debe presentar la solución propuesta, especificando los requisitos funcionales y no funcionales que debe cumplir el sistema a desarrollar. De igual forma se fundamentan los patrones de diseño empleados y, finalmente se muestra las tarjetas CRC, donde se explican los componentes con que contará la aplicación.

### **2.1 Propuesta de solución**

Con el objetivo de facilitar el proceso de compra-venta de productos y servicios entre consumidores y proveedores en las empresas estatales de Cuba, se realiza el sistema de gestión de productos y servicios entre proveedores y consumidores empleando arquitectura de microservicios. Será capaz de gestionar los procesos asociados entre consumidor y proveedor, como la reservación y ofertas de servicios y productos. Cuenta con dos componentes principales: frontend y backend. En el frontend se desarrolla, empleando el framework AngularJS lo referente a lo visual y cada uno de los métodos que se responsabiliza de manejar la información de los microservicios para mostrarla al usuario y permitirle interactuar con ella.

Por otro lado, el backend cuenta con tres microservicios y una base de datos. Los microservicios son: usuario (se encarga de manejar las creaciones, listas, actualizaciones y eliminaciones de usuarios unido a los pasos de mensajes como notificaciones), de servicios (se responsabiliza de manejar las creaciones, listas, actualizaciones y eliminaciones de los servicios) y de producto (se asegura de manejar las creaciones, listas, actualizaciones y eliminaciones de los productos). Cada uno de ellos brinda la información al frontend mediante las rutas asociadas a sus métodos.

Se reflejan procesos tales como la evaluación, reserva y notificación, creación, eliminación y listado de los productos y de los servicios. Además, la necesidad de autorización por un administrador para permitir el ingreso al sistema de los usuarios en calidad de proveedores o consumidores.

Una vez permitida la entrada al sistema por el administrador se podrá acceder a un menú. En dependencia del rol, se otorga los permisos para, en caso de ser proveedor, crear un producto, asignándole su nombre, descripción, precio unitario, disponibilidad y foto de identificación. Así como crear un servicio, con nombre, descripción, precio no obligatorio, los días de reserva establecidos y una foto de identificación. También puede consultar sus productos y servicios creados para conocer las calificaciones dadas por los usuarios o editarlos. Por otro lado, a los

consumidores se les está permitido listar productos y servicios, reservarlos, consultar los listados de sus reservas y calificarlos, basándose en una escala de 1 a 5 en forma de estrellas.

### 2.1.1 Modelo conceptual

A continuación, se presenta el modelo conceptual (ver Figura 1), siendo esta una forma de representar los conceptos más significativos en el dominio y la relación que existe entre ellos. Se realizó teniendo en cuenta que no se logra determinar el proceso del negocio con fronteras bien establecidas, donde los flujos de información son bien difusos (múltiples orígenes). Dentro de los conceptos más significativos se tienen: usuario online, usuario anónimo, usuario registrado, servicio, producto, consumidor, proveedor, administrador y mensaje.

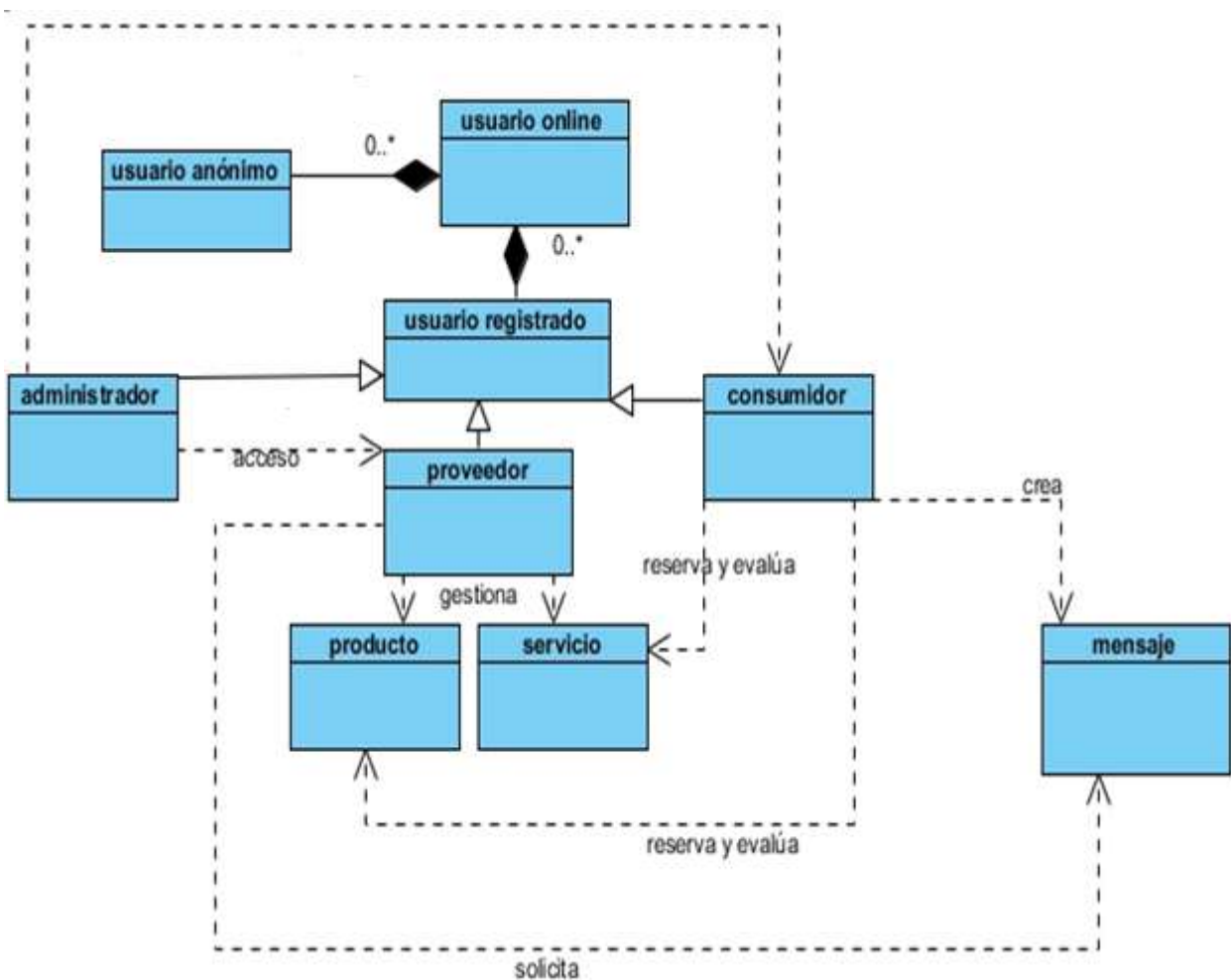


Figura 1. Modelo conceptual

### 2.1.2 Descripción de clases del modelo del dominio

A continuación, se muestra la descripción de cada uno de los conceptos que conforman el modelo conceptual:

**usuario online:**

Puede ser de dos tipos: usuario anónimo y usuario registrado.

**usuario anónimo:**

No posee ningún tipo de atributo y solo es capaz de ver la ventana principal y de requerir su registro en el sistema.

**usuario registrado:**

Puede ser administrador, usuario o proveedor y poseen un nombre, apellidos, municipio, provincia, contacto, email, contraseña, foto de usuario y rol.

**producto:**

Los productos poseen un id autogenerado por MongoDB, un proveedor que se representa con el id del usuario que se encarga de crearlo, este se declara como tipo Schema.ObjectId que apunta al usuario en cuestión. También está representado por un nombre, una descripción, una disponibilidad o stock, un precio y una imagen.

**servicio:**

Los servicios poseen un id autogenerado por MongoDB, un proveedor que se representa con el id del usuario que se encarga de crearlo, se declara como tipo Schema.ObjectId que apunta al usuario. Se registra un nombre, la descripción del servicio a brindar, su fecha de creación, su disponibilidad en forma de un arreglo de Dates, su precio base, aunque puede establecerse en la descripción en caso de estar sujeto a varios parámetros y una imagen.

**mensaje:**

Los mensajes cuentan con un texto, una fecha de cuando se generó, un emisor y un receptor. Estos últimos apuntan al consumidor y al proveedor del producto o servicio en cuestión. El mensaje es creado al realizar una reserva.

## **2.2 Roles y responsabilidad.**

**Administrador:**

El administrador es el encargado de autorizar los registros de nuevos usuarios en el sistema, sean consumidores o proveedores; o de eliminarlos.

**Proveedor:**

Los proveedores, una vez autorizado su acceso al sistema por el administrador, pueden crear productos y servicios, además de consultar la información de los existentes y editarlos.

**Consumidor:**

Los consumidores, después de haber sido autorizado su registro por parte del administrador, tienen permitido consultar información de los productos y los servicios, así como reservarlos.

## **2.3 Fase de planeación.**

La actividad de planeación (también llamada juego de planeación) comienza con la actividad para recabar requerimientos. Permite que los miembros técnicos del equipo XP entiendan el contexto del negocio para el software y adquieran la sensibilidad de la salida y características principales y funcionalidad que se requieren. (Pressman, 2010)

En dicha fase se definen los requisitos funcionales y no funcionales que posee el software. Así como un conjunto de Historias de Usuario que ofrecen la descripción de los requisitos funcionales, su tiempo estimado de desarrollo, la prioridad para el negocio y su riesgo de desarrollo.

### **2.3.1 Disciplina de requisitos.**

La gestión de requisitos apoya el logro del éxito durante la construcción del software. Propiciando un entendimiento común entre el cliente y el grupo de desarrolladores de aquellos requisitos que deben ser tomados en cuenta en la idea del producto final. Esta etapa es una de las más complejas y de ser errónea afecta en gran medida la construcción del sistema. A partir de la correcta definición de los requisitos del software se realizan las Historias de Usuario y los requisitos o características no funcionales que debe cumplir el sistema. Lográndose un entendimiento entre el equipo de desarrollo y el cliente, y especificando las necesidades reales de forma tal que satisfaga sus expectativas. Los primeros representan las condiciones que debe cumplir el sistema y los segundos se refieren a las restricciones con las que debe operar el mismo.

A continuación, se presentan un total de 19 requisitos funcionales en 10 Historias de Usuario, la iteración en la que se desarrollan y su prioridad para el negocio, esta última determinada por la complejidad a la hora de codificarla. Siendo Alta las funcionalidades de mayor grado de importancia para el cliente, Media las de un nivel medio de importancia y Baja las que no son importantes para él.

#### **Iteración 1. (Prioridad Alta)**

**RF1.** Crear producto (nombre: string, descripción: string, precio: number, disponibles: number, imagen: string)

**RF2.** Modificar producto (nombre: string, descripción: string, precio: number, disponibles: number, imagen: string)

**RF3.** Listar producto (id: string)

**RF4.** Eliminar producto (id: string)

**RF5.** Crear servicio (nombre: string, descripción: string, precio: number, reservaciones: Date[ ] , imagen: string)

**RF6.** Modificar servicio (nombre: string, descripción: string, precio: number, reservaciones: Date[ ] , imagen: string)

**RF7.** Listar servicio (id: string)

**RF8.** Eliminar servicio (id: string)

### **Iteración 2. (Prioridad Media)**

**RF9.** Autorizar registro (id: string, registered: boolean)

**RF10.** Reservar producto (id: string)

**RF11.** Reservar servicio (id: string)

**RF12.** Notificar reserva (texto: string, fecha: Date, emisor: Schema.ObjectId ref:'User', receptor: Schema.ObjectId ref: 'User')

**RF13.** Evaluar producto (id: string)

**RF14.** Evaluar servicio (id: string)

### **Iteración 3. (Prioridad Baja)**

**RF15.** Registrar usuario (nombre: string, apellidos: string, correo: string, provincia: string, municipio: string, usuario: string, rol: string, contacto: string)

**RF16.** Modificar usuario (nombre: string, apellidos: string, correo: string, provincia: string, municipio: string, usuario: string, rol: string, contacto: string, imagen: string)

**RF17.** Listar usuario (id: string)

**RF18.** Eliminar usuario (id: string)

**RF19.** Autenticar usuario (correo: string, contraseña: string)

### **2.3.2 Historias de usuario.**

Las Historias de Usuario (HU) son la técnica utilizada en la metodología XP para especificar los requisitos del software. Describen los requisitos del software en forma de tareas que el sistema debe hacer utilizando un lenguaje natural. Guían la construcción de otros artefactos incluidos en la metodología (Tareas de ingeniería y Pruebas de aceptación), utilizadas para estimar tiempos de desarrollo. (Pressman, 2010)

La prioridad en el negocio está dada por el siguiente esquema:

- Alta: Cuando son consideradas por los clientes esenciales para el funcionamiento del negocio.
- Media: Cuando el cliente cree que son necesarias, pero estas no intervienen en gran medida en el desarrollo del negocio.
- Bajo: Cuando el cliente cree que no son necesarias, pero estas no intervienen en el desarrollo del negocio.

El riesgo de desarrollo se determinó mediante los siguientes indicadores:

- Alto: Cuando en la implementación de las HU pueden surgir errores que lleven a la inoperatividad del código.
- Medio: Cuando en la implementación de las HU pueden existir errores que retrasen la entrega del producto.
- Bajo: Cuando pueden aparecer errores que serán tratados con relativa facilidad sin que conlleven perjuicios para el desarrollo del proyecto.

El tiempo estimado se define en base a la complejidad de desarrollo de cada uno de los requisitos funcionales que intervienen en la realización de la Historia de Usuario. El tiempo real se establece al terminar la codificación de cada requisito funcional que compone la Historia de Usuario.

**Tabla 1 HU\_1 Gestionar producto**

<b>Numero:</b> HU_1	<b>Nombre:</b> Gestionar producto
<b>Programador responsable:</b> Roberto Camejo	<b>Iteración asignada:</b> 1
<b>Prioridad en negocio:</b> Alta	<b>Tiempo estimado:</b> 6
<b>Riesgo de desarrollo:</b> Alto	<b>Tiempo real:</b> 5
<b>Descripción:</b> Permite al proveedor crear sus productos y modificar cualquiera de sus campos o eliminarlo. En caso del consumidor puede listar todos los productos, listar los productos que ha adquirido, generados al hacer las reservas y evaluarlos.	
<b>Observaciones:</b> El proveedor y el consumidor pueden acceder a estas funcionalidades una vez su cuenta haya sido verificada por el administrador y estén autenticados.	

**Tabla 2 HU\_2 Gestionar servicio**

<b>Numero:</b> HU_2	<b>Nombre:</b> Gestionar servicios
<b>Programador responsable:</b> Roberto Camejo	<b>Iteración asignada:</b> 1
<b>Prioridad en negocio:</b> Alta	<b>Tiempo estimado:</b> 4
<b>Riesgo de desarrollo:</b> Alto	<b>Tiempo real:</b> 4
<b>Descripción:</b> Permite al proveedor crear sus servicios y modificar cualquiera de sus campos o eliminarlo. En caso del consumidor puede listar todos los servicios, listar los	

servicios que ha adquirido, generados al hacer las reservas y evaluarlos.
<b>Observaciones:</b> El proveedor y el consumidor pueden acceder a estas funcionalidades una vez su cuenta haya sido verificada por el administrador y estén autenticados.

**Tabla 3 HU\_3 Autorizar registro**

<b>Numero:</b> HU_3	<b>Nombre:</b> Autorizar registro
<b>Programador responsable:</b> Roberto Camejo	<b>Iteración asignada:</b> 2
<b>Prioridad en negocio:</b> Media	<b>Tiempo estimado:</b> 3
<b>Riesgo de desarrollo:</b> Medio	<b>Tiempo real:</b> 3
<b>Descripción:</b> Permite al administrador autorizar el registro de las cuentas de usuario de los proveedores y los consumidores.	
<b>Observaciones:</b> El administrador puede acceder al sistema siempre y cuando se haya autenticado anteriormente.	

**Tabla 4 HU\_4 Reservar producto**

<b>Numero:</b> HU_4	<b>Nombre:</b> Requerir reserva de producto
<b>Programador responsable:</b> Roberto Camejo	<b>Iteración asignada:</b> 2
<b>Prioridad en negocio:</b> Media	<b>Tiempo estimado:</b> 2
<b>Riesgo de desarrollo:</b> Bajo	<b>Tiempo real:</b> 2
<b>Descripción:</b> Permite al consumidor solicitar la cantidad de productos deseados para su posterior reservación.	
<b>Observaciones:</b> El consumidor puede acceder al sistema siempre y cuando se haya autenticado anteriormente.	

**Tabla 5 HU\_5 Reservar servicio**

<b>Numero:</b> HU_5	<b>Nombre:</b> Requerir reserva de servicio
<b>Programador responsable:</b> Roberto Camejo	<b>Iteración asignada:</b> 2
<b>Prioridad en negocio:</b> Media	<b>Tiempo estimado:</b> 2
<b>Riesgo de desarrollo:</b> Bajo	<b>Tiempo real:</b> 2
<b>Descripción:</b> Permite al consumidor solicitar los servicios, reflejados en un calendario, para su posterior reserva.	
<b>Observaciones:</b> El consumidor puede acceder al sistema siempre y cuando se haya autenticado anteriormente.	



**Tabla 6 HU\_6 Evaluar producto**

<b>Numero:</b> HU_6	<b>Nombre:</b> Evaluar producto
<b>Programador responsable:</b> Roberto Camejo	<b>Iteración asignada:</b> 2
<b>Prioridad en negocio:</b> Media	<b>Tiempo estimado:</b> 1
<b>Riesgo de desarrollo:</b> Bajo	<b>Tiempo real:</b> 1
<b>Descripción:</b> Permite al consumidor evaluar los productos adquiridos con una calificación de entre 0 y 5 puntos, dicha evaluación será resultado del promedio entre el valor de las votaciones y la cantidad de votos del producto.	
<b>Observaciones:</b> El consumidor puede acceder al sistema siempre y cuando se haya autenticado anteriormente.	

**Tabla 7 HU\_7 Evaluar servicio**

<b>Numero:</b> HU_7	<b>Nombre:</b> Evaluar servicio
<b>Programador responsable:</b> Roberto Camejo	<b>Iteración asignada:</b> 2
<b>Prioridad en negocio:</b> Media	<b>Tiempo estimado:</b> 1
<b>Riesgo de desarrollo:</b> Bajo	<b>Tiempo real:</b> 1
<b>Descripción:</b> Permite al consumidor evaluar los servicios adquiridos con una calificación de entre 0 y 5 puntos, dicha evaluación será resultado del promedio entre el valor de las votaciones y la cantidad de votos del servicio.	
<b>Observaciones:</b> El consumidor puede acceder al sistema siempre y cuando se haya autenticado anteriormente.	

**Tabla 8 HU\_8 Notificar reserva**

<b>Numero:</b> HU_8	<b>Nombre:</b> Notificar reserva
<b>Programador responsable:</b> Roberto Camejo	<b>Iteración asignada:</b> 2
<b>Prioridad en negocio:</b> Media	<b>Tiempo estimado:</b> 4
<b>Riesgo de desarrollo:</b> Bajo	<b>Tiempo real:</b> 3
<b>Descripción:</b> Cuando el consumidor reserva un producto o un servicio se envía una notificación al proveedor con los detalles de la misma.	
<b>Observaciones:</b> El consumidor y el proveedor pueden acceder al sistema siempre y cuando se haya autenticado anteriormente.	

**Tabla 9 HU\_9 Gestionar Usuario**

<b>Numero:</b> HU_9	<b>Nombre:</b> Gestionar usuario
---------------------	----------------------------------

<b>Programador responsable:</b> Roberto Camejo	<b>Iteración asignada:</b> 3
<b>Prioridad en negocio:</b> Baja	<b>Tiempo estimado:</b> 8
<b>Riesgo de desarrollo:</b> Alto	<b>Tiempo real:</b> 7
<b>Descripción:</b> Permite, en caso de los administradores cambiar el estado de los usuarios para permitirles o no el ingreso al sistema. En caso de cualquier usuario no registrado le da la posibilidad de registrarse introduciendo sus datos de nombre, apellidos, email, provincia, contacto, municipio y usuario que lo va a identificar dentro del sistema. Una vez registrados y con permiso de acceder al sistema pueden editar sus respectivos perfiles.	
<b>Observaciones:</b> El administrador puede acceder al sistema siempre y cuando tenga una cuenta válida y los proveedores y consumidores si su cuenta ha sido aprobada por el administrador.	

**Tabla 10 HU\_10 Autenticar usuario**

<b>Numero:</b> HU_10	<b>Nombre:</b> Autenticar usuario
<b>Programador responsable:</b> Roberto Camejo	<b>Iteración asignada:</b> 3
<b>Prioridad en negocio:</b> Baja	<b>Tiempo estimado:</b> 3
<b>Riesgo de desarrollo:</b> Alto	<b>Tiempo real:</b> 3
<b>Descripción:</b> Permite al usuario ingresar al sistema al introducir su dirección de correo electrónico y su contraseña.	
<b>Observaciones:</b> El administrador puede acceder al sistema, siempre y cuando posea una cuenta válida y los proveedores y consumidores si su cuenta ha sido aprobada por el administrador.	

### 2.3.3 Requisitos no funcionales.

“Los requerimientos no funcionales, como su nombre sugieren, son aquellos requerimientos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino, a las propiedades emergentes de éste como la fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento” (Somerville, 2005).

En muchos casos los requisitos no funcionales son fundamentales en el éxito del producto, por tanto, deben ser apropiadamente definidos y analizados. A continuación, se describen los requisitos no funcionales que establecen las características del sistema, los cuales son listados a continuación:

**Usabilidad:**

**RNF1:** La aplicación deberá contar con una interfaz sencilla, descriptiva y fácil de usar por cualquier usuario con conocimientos básicos de computación.

**RNF2:** Debe estar adaptada a las pautas de diseño de la universidad y a la estrategia marcaria de la XETID.

**RNF3:** Las operaciones a informatizar serán lo más parecidas posible a los procesos que se realizan actualmente en las entidades, para así lograr un menor tiempo en cuanto a la comprensión y adaptación del sistema.

**Software:**

**RNF4:** Se recomienda hacer uso del navegador Mozilla Firefox (versión 3.5 o superior), Internet Explorer o Google Chrome.

**Hardware:**

**RNF5:** Como requerimientos mínimos para acceder al sistema debe tener una CPU Dual Core a 1.0 GHz o superior, 512 MB de RAM o superior, 128 GB de disco duro o superior.

**RNF6:** El sistema se desarrollará con la tecnología que ofrece NodeJs en su versión 6.0.9.

**RNF7:** El sistema utilizará una base datos implementada en MongoDB versión 4.0 o superior.

**Seguridad****Confidencialidad:**

**RNF8:** La información manejada por el sistema está protegida de acceso no autorizado y divulgación, pestableciéndose un nivel de acceso a la aplicación mediante usuario y contraseña, además, de los permisos asignados por usuarios.

**Integridad:**

**RNF9:** La información manejada por el sistema será objeto de cuidadosa protección contra la corrupción y estados inconsistentes.

**Disponibilidad:**

**RNF10:** La aplicación deberá estar disponible en todo momento para aquellas personas que necesiten acceder y manejar la información.

## **2.4 FASE DE DISEÑO**

El diseño en la metodología XP sigue rigurosamente el principio MS (mantenlo sencillo). Un diseño sencillo siempre se prefiere sobre una representación más compleja. Además, el diseño guía la implementación de una historia conforme se escribe: nada más y nada menos. Se desalienta el diseño de funcionalidad adicional porque el desarrollador supone que se requerirá después. (Pressman, 2010)

### **2.4.1 Arquitectura de software.**

#### **Arquitectura de Microservicios (Backend):**

Los microservicios, o arquitectura de microservicios, es un enfoque para el desarrollo de software en el que una aplicación grande se construye como un conjunto de componentes o servicios modulares. Cada módulo admite una tarea específica o un objetivo de negocio y utiliza una interfaz simple y bien definida, como una interfaz de programación de aplicaciones (API), para comunicarse con otros conjuntos de servicios. (Wolff, 2016)

Una de las ventajas de utilizar microservicios es la capacidad de publicar una aplicación grande como un conjunto de pequeñas aplicaciones (microservicios) que se pueden desarrollar, desplegar, escalar, manejar y visualizar de forma independiente. Los microservicios permiten a las empresas gestionar las aplicaciones de código base grande usando una metodología más práctica donde las mejoras incrementales son ejecutadas por pequeños equipos en bases de código y despliegues independientes. (Villamizar, 2017)

#### **Module-Template-Component-Injector (Arquitectura de desarrollo de AngularJS):**

La arquitectura empleada es por funcionalidad, definida por los desarrolladores del framework Angular JS. Coloca en carpetas por lógica del negocio o funcionalidad un componente de cada tipo, permitiendo en un futuro reutilizar esa funcionalidad porque debe estar aislada y ser modular. (Molina, 2018) Estos componentes son:

1. **Controllers:** Deben de tener toda interacción con el usuario y delegar tareas entre los diferentes servicios.
2. **Directives:** Aquí se coloca toda la lógica de componentes reutilizables para la aplicación y el manejo del dom.
3. **Serviles:** El archivo tiene la lógica de modelos de datos, el formato de datos y las peticiones al servicio.
4. **Carpeta Views:** Aquí se coloca todas las vistas que necesite la aplicación.

## 2.4.2 Tarjeta CRC.

Las tarjetas CRC (clase-responsabilidad-colaborador) identifican y organizan las clases orientadas a objetos que son relevantes para el incremento actual de software. (Roger S. Pressman, Ph.D. 2010)

Se utilizan, individualmente, para representar objetos. La clase del objeto puede ser escrita en la parte superior de la tarjeta, las responsabilidades enumeradas en la parte izquierda y las clases que colaboran son listadas a la derecha de cada responsabilidad. Se dice: "puede ser escrito" porque una vez que una sesión de CRC se encuentra en pleno desarrollo, por lo general, los participantes sólo necesitan unas cuantas tarjetas con el nombre de la clase y prácticamente ninguna tarjeta escrita en su totalidad. (Zambrano, 2017)

**Tabla 11 Tarjeta CRC Usuario**

Tarjeta CRC Usuario	
<b>Clase: userSchema</b>	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> <li>• Se encarga de registrar un usuario mediante el método crearUsuario().</li> <li>• Se encarga de listar los usuarios mediante el método obtenerUsuarios().</li> <li>• Se encarga de modificar los usuarios mediante el método actualizarUsuario().</li> <li>• Se encarga de eliminar un usuario mediante el método eliminarUsuario().</li> <li>• Se encarga de autenticar a los usuarios mediante el método api.post().</li> </ul>	<ul style="list-style-type: none"> <li>• Administrador</li> <li>• Consumidor</li> <li>• Proveedor</li> </ul>

**Tabla 12 Tarjeta CRC Producto**

Tarjeta CRC Producto	
<b>Clase: productoSchema</b>	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> <li>• Se encarga de registrar un producto mediante el método crearProducto ().</li> <li>• Se encarga de listar los productos mediante el método obtenerProducto</li> </ul>	<ul style="list-style-type: none"> <li>• Consumidor</li> <li>• Proveedor</li> </ul>

<p>().</p> <ul style="list-style-type: none"> <li>• Se encarga de modificar los productos mediante el método actualizarProducto ().</li> <li>• Se encarga de eliminar un producto mediante el método eliminarProducto ().</li> <li>• Se encarga de evaluar los productos mediante la función evalProd().</li> <li>• Se encarga de la solicitud de los productos mediante el método reqPedido()</li> </ul>	
---	--

**Tabla 13 Tarjeta CRC Servicio**

<b>Tarjeta CRC Servicio</b>	
<b>Clase: serviceSchema</b>	
<b>Responsabilidad</b>	<b>Colaboración</b>
<ul style="list-style-type: none"> <li>• Se encarga de registrar un producto mediante el método crearServicio().</li> <li>• Se encarga de listar los productos mediante el método obtenerServicio().</li> <li>• Se encarga de modificar los productos mediante el método actualizarServicio().</li> <li>• Se encarga de eliminar un producto mediante el método eliminarServicio().</li> <li>• Se encarga de evaluar los productos mediante la función evalServ().</li> <li>• Se encarga de la solicitud de los productos mediante el método reqAlquil()</li> </ul>	<ul style="list-style-type: none"> <li>• Consumidor</li> <li>• Proveedor</li> </ul>

## 2.5 Representación de los datos

```
{  
  name: {type:String, required: true},  
  apellidos: {type:String, required: true},  
  email: {type:String, unique:true, required: true},  
  provincia: {type:String, required: true},  
  contacto:{type:String, require:true},  
  municipio: {type:String, required: true},  
  usuario: {type:String, unique: true, required: true},  
  password: {type:String, required: true},  
  image: {type:String, required: false},  
  rol: {type:String, required: false},  
  registered: {type:Boolean, required: false, default: false},  
  productos:[{type: Schema.ObjectId, ref: 'Producto', required: false}],  
  servicios:[{type: Schema.ObjectId, ref: 'Servicio', required: false}]  
}
```



Representación de  
la clase Mensaje  
en formato JSON

```
{  
  fecha:{type: Date, required: false, default: Date.now()},  
  emisor:{type: Schema.ObjectId, ref: 'User',required: false},  
  receptor:{type: Schema.ObjectId, ref: 'User',required: false},  
  texto:{type: String, required: false},  
}
```



Representación de  
la clase Usuario en  
formato JSON

```
{  
  nombre: {type: String,required: true},
```

```

created_at: {type: Date,default: Date.now,required :false},
bussy_days: [{type: Date,required: false}],
descripcion: {type: String,required: false},
price: {type: Number,required: false},
evalu: {type: Number,required: false},
votos: {type: Number,required: false},
proveedor: {type: Schema.ObjectId,ref: 'User',required: true},
image: {type: String,required: false}
}

```



Representación de la clase Servicio en formato JSON

```

{
nombre: {type: String,required: true},
stock: {type: Number,required: true},
descripcion: {type: String,required: false},
price: { type: Number,required: false},
proveedor: {type: Schema.ObjectId,ref: 'User',required:true},
evalu: { type: Number,required: false},
votos: { type: Number,required: false},
image: {type: String,required: false,}
}

```



Representación de la clase Producto en formato JSON

## 2.6 Patrones GRASP

Los patrones de diseño constituyen la base para la búsqueda de soluciones a problemas comunes en el proceso de desarrollo de software. Se caracterizan por su reusabilidad, flexibilidad y aplicabilidad a diferentes problemas de diseño en diversas circunstancias. El patrón es un esquema de solución que se aplica a un tipo de situación, donde la aplicación del mismo no es mecánica, sino que requiere de adaptación.

Aunque se considera que más que patrones propiamente dichos, son una serie de "buenas prácticas" de aplicación recomendable en el diseño de software. Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos y sirven para asignar correctamente las responsabilidades en el diseño de la programación orientada a objetos (POO).



Los patrones GRASP son un método de enseñanza que ayuda a entender el Diseño de POO y aplica el análisis y diseño de objetos de un modo sistemático, racional y explicable. (Alfonso, 2017)

**Experto:** Asigna la responsabilidad a la clase que tiene la información necesaria para cumplir la responsabilidad. Toda la información de los datos la manejan las clases modelos, pues son estas las que acceden a la base de datos en respuesta de una petición de las clases controladoras para satisfacer un evento iniciado por el usuario. Por ejemplo (Ver Figura 2), a la hora de evaluar un servicio se obtienen los datos a evaluar de la clase Servicio, del servicio en cuestión.

```
this.servicios = [];  
id.forEach(aaa => {  
  this.encuestaservice.getServiciox(aaa).subscribe(qwe => {  
    this.servicios.push(qwe);  
    this.servicios[this.servicios.lastIndexOf(qwe)]  
      .evalu = this.servicios[this.servicios.lastIndexOf(qwe)]  
      .evalu / this.servicios[this.servicios.lastIndexOf(qwe)]  
      .votos;  
  });  
});
```

Figura 2. Ejemplo de fragmento de código donde se aplica el patrón xExperto

**Controlador:** Asignar la responsabilidad de manejar los eventos del sistema a una clase. Un evento de entrada al sistema es algún evento desde algún actor externo. La clase controlador es responsable de decidir qué hacer con el evento. El controlador actúa como una fachada entre la interfaz y la aplicación. Se pone de manifiesto al pulsar el botón de cierre de sesión, que desencadena el método logout() e indica que la sesión debe cerrarse. (Ver Figura 3)

```
logout() {  
  localStorage.clear();  
  this.identity = null;  
  this._router.navigate(['/',]);  
}
```

Figura 3. Ejemplo de fragmento de código donde se aplica el patrón Controlador

**Bajo acoplamiento:** Es la idea de tener las clases lo menos ligadas entre sí como sea posible. De tal forma que, en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de clases, lo que potencia la reutilización, y disminuye la dependencia entre las clases. El patrón es empleado desde la misma declaración de cada una de las clases, pues están separadas en microservicios que las hacen independientes unas de otras.

**Alta cohesión:** Se aplica en la mayoría de las clases del diseño, en cada una solo se implementan las funcionalidades que le corresponden. La aplicación contiene controladores por cada clase que exportan módulos con los métodos específicos de cada una. (Ver Figura 4)

```
module.exports = {
  obtenerUsuarios,
  crearUsuario,
  actualizarUsuario,
  eliminarUsuario
};
```

Figura 4. Ejemplo de fragmento de código donde se aplica el patrón Alta cohesión

## 2.7 Conclusiones parciales.

Una vez terminado el capítulo se puede concluir que:

- El modelo conceptual de la propuesta de solución permitió representar los conceptos significativos del negocio y la relación que existe entre estos.
- La fase de Planificación propuesta por la metodología XP permitió describir un total de 10 Historias de Usuario (HU) del sistema, obtener una estimación de cada una de ellas y las iteraciones a realizar.
- La descripción de las HU identificadas permitió un mejor entendimiento para su futura implementación.
- La identificación de las características no funcionales permitió conocer las condiciones bajo las cuales debe operar el sistema, identificándose un total de 10 requisitos no funcionales.
- La fase de Diseño favoreció obtener una representación de la arquitectura del sistema, así como los patrones de diseño a utilizar y las tarjetas CRC, esto último, como artefacto propuesto en la fase por la metodología que guía el desarrollo de la presente investigación.

## CAPITULO 3 Implementación y pruebas del software

En este capítulo se desarrollan las dos últimas fases que plantea la metodología XP: Desarrollo y Pruebas. En el caso de la primera se da inicio con las tareas de ingeniería, donde cada HU se convierte en tareas a cumplir por los programadores y por último se definen los estándares de codificación a tener en cuenta durante el desarrollo. En la última fase se realizan pruebas de aceptación y unitarias para verificar el correcto funcionamiento del sistema que se obtuvo como resultado de la presente investigación.

### 3.1 Tarea de ingeniería

Se sugiere que antes de comenzar cualquier trabajo técnico es una buena idea aplicar un conjunto de tareas de ingeniería a los requerimientos. Éstas llevarán a la comprensión de cuál será el efecto que tendrá el software en el negocio, qué es lo que quiere el cliente y cómo interactuarán los usuarios finales con el software. (Roger S. Pressman, Ph.D. 2010)

Antes de comenzar la realización de las Tareas de Ingeniería se debe tener en cuenta que un punto de estimación equivale a un mes de programación, un mes de programación corresponde a 22 días, 880 horas en desarrollo, 40 horas en la semana, 8 horas durante 5 días de la semana del calendario normal. Los puntos de estimación se calcularon teniendo en cuenta el tiempo de realización de cada tarea y la cantidad de transacciones que se desarrollan en la misma.

**Tabla 14 Tarea 1 Insertar Producto**

Tarea	
<b>Numero de tarea:</b> 1	<b>Numero de historia de usuario:</b> HU_1
<b>Nombre de tarea:</b> Insertar producto	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.4
<b>Fecha de inicio:</b> 23/dic/2018	<b>Fecha de fin:</b> 25/dic/2018
<b>Programador responsable:</b> Roberto Camejo Domínguez	
<b>Descripción:</b> El proveedor, una vez autenticado, procede a llenar el formulario del producto, el cual finaliza al presionar el botón "Enviar"	

**Tabla 15 Tarea 2 Modificar Producto**

Tarea	
<b>Numero de tarea:</b> 2	<b>Numero de historia de usuario:</b> HU_1
<b>Nombre de tarea:</b> Modificar producto	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.6

<b>Fecha de inicio:</b> 25/dic/2018	<b>Fecha de fin:</b> 27/dic/2018
<b>Programador responsable:</b> Roberto Camejo Domínguez	
<b>Descripción:</b> El proveedor accede al listado de sus productos y selecciona mediante el botón "Editar" el producto que desea modificar, pulsando el botón "Enviar" para terminar con la modificación.	

**Tabla 16 Tarea 3 Listar Productos**

Tarea	
<b>Numero de tarea:</b> 3	<b>Numero de historia de usuario:</b> HU_1
<b>Nombre de tarea:</b> Listar productos	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.2
<b>Fecha de inicio:</b> 9/enero/2019	<b>Fecha de fin:</b> 11/enero/2019
<b>Programador responsable:</b> Roberto Camejo Domínguez	
<b>Descripción:</b> El consumidor, una vez autenticado en el sistema, accede a la pestaña de productos, donde obtiene la lista de los productos disponibles.	

**Tabla 17 Tarea 4 Eliminar Producto**

Tarea	
<b>Numero de tarea:</b> 4	<b>Numero de historia de usuario:</b> HU_1
<b>Nombre de tarea:</b> Eliminar producto	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.2
<b>Fecha de inicio:</b> 11/enero/2019	<b>Fecha de fin:</b> 12/enero/2019
<b>Programador responsable:</b> Roberto Camejo Domínguez	
<b>Descripción:</b> El proveedor accede a la pestaña de productos, una vez autenticado en el sistema, y selecciona los productos a eliminar. Estos son eliminados al presionar el botón "Eliminar"	

**Tabla 18 Tarea 5 Insertar Servicio**

Tarea	
<b>Numero de tarea:</b> 5	<b>Numero de historia de usuario:</b> HU_2
<b>Nombre de tarea:</b> Insertar servicio	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.4
<b>Fecha de inicio:</b> 12/enero/2019	<b>Fecha de fin:</b> 13/enero/2019
<b>Programador responsable:</b> Roberto Camejo Domínguez	

**Descripción:** El proveedor, una vez autenticado, procede a llenar el formulario del servicio, el cual finaliza al presionar el botón “Enviar”

**Tabla 19 Tarea 6 Modificar Servicio**

Tarea	
<b>Numero de tarea:</b> 6	<b>Numero de historia de usuario:</b> HU_2
<b>Nombre de tarea:</b> Modificar servicio	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.6
<b>Fecha de inicio:</b> 13/enero/2019	<b>Fecha de fin:</b> 14/enero/2019
<b>Programador responsable:</b> Roberto Camejo Domínguez	
<b>Descripción:</b> El proveedor accede al listado de sus servicios y selecciona mediante el botón “Editar” el servicio que desea modificar, pulsando el botón “Enviar” para terminar con la modificación.	

**Tabla 20 Tarea 7 Listar Servicios**

Tarea	
<b>Numero de tarea:</b> 7	<b>Numero de historia de usuario:</b> HU_2
<b>Nombre de tarea:</b> Listar servicios	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.2
<b>Fecha de inicio:</b> 14/enero/2019	<b>Fecha de fin:</b> 15/enero/2019
<b>Programador responsable:</b> Roberto Camejo Domínguez	
<b>Descripción:</b> El consumidor, una vez autenticado en el sistema, accede a la pestaña de servicios, donde obtiene la lista de los servicios disponibles al pulsar el botón “Buscar”.	

**Tabla 21 Tarea 8 Eliminar Servicio**

Tarea	
<b>Numero de tarea:</b> 8	<b>Numero de historia de usuario:</b> HU_2
<b>Nombre de tarea:</b> Eliminar servicio	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.2
<b>Fecha de inicio:</b> 15/enero/2019	<b>Fecha de fin:</b> 15/enero/2019
<b>Programador responsable:</b> Roberto Camejo Domínguez	
<b>Descripción:</b> El proveedor accede a la pestaña de servicios, una vez autenticado en el sistema, y selecciona los servicios a eliminar. Estos son eliminados al presionar el botón “Eliminar”	

**Tabla 22 Tarea 9 Autorizar registro**

Tarea	
<b>Numero de tarea:</b> 9	<b>Numero de historia de usuario:</b> HU_3
<b>Nombre de tarea:</b> Autorizar registro de usuario	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.5
<b>Fecha de inicio:</b> 16/enero/2019	<b>Fecha de fin:</b> 18/enero/2019
<b>Programador responsable:</b> Roberto Camejo Domínguez	
<b>Descripción:</b> El administrador, después de autenticarse en el sistema, accede al panel donde están listados los usuarios pendientes de aprobación, luego de seleccionarlos presiona el botón de “Enviar”.	

**Tabla 23 Tarea 10 Requerir Reserva de Producto**

Tarea	
<b>Numero de tarea:</b> 10	<b>Numero de historia de usuario:</b> HU_4
<b>Nombre de tarea:</b> Requerir reserva de producto	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.4
<b>Fecha de inicio:</b> 20/enero/2019	<b>Fecha de fin:</b> 21/enero/2019
<b>Programador responsable:</b> Roberto Camejo Domínguez	
<b>Descripción:</b> El consumidor accede al producto que desea adquirir, luego de autenticarse en el sistema, y selecciona la cantidad del producto. Esta acción concluye al presionar el botón “Enviar” el cual envía una notificación al proveedor de dicho producto para informarle de la adquisición.	

**Tabla 24 Tarea 11 Requerir Alquiler de Servicio**

Tarea	
<b>Numero de tarea:</b> 11	<b>Numero de historia de usuario:</b> HU_5
<b>Nombre de tarea:</b> Requerir alquiler del servicio	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.3
<b>Fecha de inicio:</b> 22/enero/2019	<b>Fecha de fin:</b> 23/enero/2019
<b>Programador responsable:</b> Roberto Camejo Domínguez	
<b>Descripción:</b> El consumidor accede al servicio que desea adquirir, luego de autenticarse en el sistema, y selecciona los días de alquiler del servicio. Esta acción concluye al presionar el botón “Enviar” el cual envía una notificación al proveedor de dicho servicio para informarle del alquiler.	

**Tabla 25 Tarea 12 Evaluar Producto**

Tarea	
<b>Numero de tarea:</b> 12	<b>Numero de historia de usuario:</b> HU_6
<b>Nombre de tarea:</b> Evaluar producto	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.3
<b>Fecha de inicio:</b> 4/febrero/2019	<b>Fecha de fin:</b> 4/febrero/2019
<b>Programador responsable:</b> Roberto Camejo Domínguez	
<b>Descripción:</b> El consumidor accede al listado de productos adquiridos, una vez autenticado en el sistema, y emite una calificación del producto seleccionado.	

**Tabla 26 Tarea 13 Evaluar Servicio**

Tarea	
<b>Numero de tarea:</b> 12	<b>Numero de historia de usuario:</b> HU_7
<b>Nombre de tarea:</b> Evaluar servicio	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.2
<b>Fecha de inicio:</b> 6/febrero/2019	<b>Fecha de fin:</b> 6/febrero/2019
<b>Programador responsable:</b> Roberto Camejo Domínguez	
<b>Descripción:</b> El consumidor accede al listado de servicios adquiridos, una vez autenticado en el sistema, y emite una calificación del servicio seleccionado.	

**Tabla 27 Tarea 14 Notificar reserva**

Tarea	
<b>Numero de tarea:</b> 13	<b>Numero de historia de usuario:</b> HU_8
<b>Nombre de tarea:</b> Notificar reserva	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.2
<b>Fecha de inicio:</b> 5/enero/2019	<b>Fecha de fin:</b> 7/enero/2019
<b>Programador responsable:</b> Roberto Camejo Domínguez	
<b>Descripción:</b> El consumidor reserva una cantidad de productos en el menú de productos o reserva un servicio por un tiempo determinado y el sistema envía una notificación al proveedor del producto o servicio reservado.	

**Tabla 28 Tarea 15 Registrar usuario**

Tarea	
<b>Numero de tarea:</b> 15	<b>Numero de historia de usuario:</b> HU_9
<b>Nombre de tarea:</b> Registrar usuario	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.8
<b>Fecha de inicio:</b> 7/enero/2019	<b>Fecha de fin:</b> 8/enero/2019
<b>Programador responsable:</b> Roberto Camejo Domínguez	
<b>Descripción:</b> El usuario llena los campos del formulario de registro para crear su cuenta en el sistema (correo: string y contraseña: string) y presiona el botón de registrar para su confirmación.	

**Tabla 29 Tarea 16 Modificar Usuario**

Tarea	
<b>Numero de tarea:</b> 29	<b>Numero de historia de usuario:</b> HU_9
<b>Nombre de tarea:</b> Modificar usuario	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.7
<b>Fecha de inicio:</b> 8/enero/2019	<b>Fecha de fin:</b> 9/enero/2019
<b>Programador responsable:</b> Roberto Camejo Domínguez	
<b>Descripción:</b> El usuario accede al panel de usuario donde puede modificar datos de su información personal y presiona el botón de "hecho" para su confirmación.	

**Tabla 30 Tarea 17 Listar usuario**

Tarea	
<b>Numero de tarea:</b> 17	<b>Numero de historia de usuario:</b> HU_9
<b>Nombre de tarea:</b> Listar usuario	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.3
<b>Fecha de inicio:</b> 27/dic/2018	<b>Fecha de fin:</b> 29/dic/2018
<b>Programador responsable:</b> Roberto Camejo Domínguez	
<b>Descripción:</b> El administrador accede a su panel al autenticarse donde solicita el listado de los usuarios para comprobar los pendientes de registro.	

**Tabla 31 Tarea 18 Eliminar usuario**

Tarea	
<b>Numero de tarea:</b> 18	<b>Numero de historia de usuario:</b> HU_9



<b>Nombre de tarea:</b> Eliminar usuario	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.3
<b>Fecha de inicio:</b> 29/dic/2018	<b>Fecha de fin:</b> 29/dic/2018
<b>Programador responsable:</b> Roberto Camejo Domínguez	
<b>Descripción:</b> El administrador, luego de listar los usuarios, selecciona los que no se pueda comprobar la autenticidad de su cuenta y los elimina mediante el botón "Eliminar".	

**Tabla 32 Tarea 19 Autenticar Usuario**

<b>Tarea</b>	
<b>Numero de tarea:</b> 19	<b>Numero de historia de usuario:</b> HU_10
<b>Nombre de tarea:</b> Autenticar usuario	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 0.7
<b>Fecha de inicio:</b> 8/febrero/2019	<b>Fecha de fin:</b> 9/febrero/2019
<b>Programador responsable:</b> Roberto Camejo Domínguez	
<b>Descripción:</b> El usuario inserta su "usuario" y "contraseña" para ingresar al sistema.	

### 3.2 Estrategia de pruebas

Uno de los pilares de la metodología XP es el proceso de pruebas, pues anima a probar constantemente, o tanto como sea posible. Permitiendo aumentar la calidad de los sistemas, reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y su detección. (Joskowicz, 2008)

La metodología XP divide las pruebas en dos grupos: pruebas unitarias, desarrolladas por los programadores, encargadas de verificar el código de forma automática. Y las pruebas de aceptación, destinadas a evaluar si al final de una iteración se obtuvo la funcionalidad requerida, además de comprobar que dicha funcionalidad sea la esperada por el cliente. (Pressman, 2010)

Las características clave de las pruebas en XP son:

- Desarrollo previamente probado.
- Desarrollo de pruebas incremental a partir de los escenarios.
- Participación del usuario en el desarrollo de las pruebas y en la validación.
- El uso de bancos de pruebas automatizados.

### 3.2.1 Pruebas unitarias

La creación de pruebas unitarias antes de que comience la codificación es un elemento clave del enfoque de la metodología XP. Las pruebas unitarias que se crean deben implementarse con el uso de una estructura que permita automatizarlas (de modo que puedan ejecutarse repetidas veces y con facilidad).

#### Jasmine y Karma:

Jasmine es un behavior-driven development framework para probar código JavaScript. No depende de ningún otro framework de JavaScript y tiene una sintaxis clara y obvia para que se pueda escribir fácilmente las pruebas. Este tiene un test-runner denominado Karma que permite automatizar las tareas de testing del framework. Estas herramientas son empleadas para las pruebas debido a que son las creadas por los desarrolladores de Angular para analizar las aplicaciones que emplean este framework como tecnología de desarrollo.

A continuación se muestra un ejemplo de las pruebas efectuadas al sistema con el uso de las herramientas anteriormente mencionadas. (Ver Figuras 5 y 6)

Prueba número 1 con Jasmine y Karma:



Figura 5. Prueba unitaria aplicada al sistema con las herramientas Jasmine y Karma

En la interfaz se muestra el resultado de la primera prueba arrojando un total de tres errores en la interfaz de la aplicación, producidos al dejar varias etiquetas sin uso dentro de los módulos HTML.

Prueba número 2 con Jasmine y Karma:



Figura 6. Prueba unitaria aplicada al sistema con las herramientas Jasmine y Karma

En la interfaz se muestra los resultados satisfactorios arrojados por la aplicación de las herramientas de prueba, obteniéndose un total de 0 errores en la funcionalidad de la interfaz de la aplicación.

### **Mocha:**

A mocha se le puede definir como un framework de pruebas rico en características que puede ser implementado en NodeJS o por medio de un navegador. Mocha ejecuta las pruebas en serie permitiendo reportes flexibles y exactos, haciendo que el código pasado pruebas sea óptimo al momento de estar en producción. En esta prueba se crea un archivo test.js el cual contiene los métodos de la API y su respuesta. La herramienta facilita la información del estado de los retornos de las funciones y del tiempo promedio de respuesta del servicio. (Ver Figuras 7, 8 y 9)

Resultados arrojados por la herramienta Mocha para el backend:

Prueba realizada al microservicio Usuario:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1:cmd
1 passing (40ms)
E:\sico\apis\usuarios\mocha

obtenerUsuarios
✓ debe retornar listado de usuarios
✓ debe retornar usuario actualizado
✓ debe retornar usuario creado
✓ debe retornar usuario eliminado

4 passing (40ms)
E:\sico\apis\usuarios
```

Figura 7. Prueba aplicada al sistema con la herramienta Mocha

Los resultados de la aplicación de la prueba arrojaron un tiempo promedio de respuesta del servicio Usuario de 40 ms y todos los retornos de los métodos en un estado correcto.

Prueba realizada al microservicio Producto:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1:cmd
E:\sico\apis\servicios>cd..
E:\sico\apis>cd productos
E:\sico\apis\productos\mocha

service
✓ debe retornar listado de productos
✓ debe retornar producto actualizado
✓ debe retornar producto creado
✓ debe retornar producto eliminado

4 passing (53ms)
E:\sico\apis\productos
```

Figura 8. Prueba aplicada al sistema con la herramienta Mocha

Los resultados de la prueba arrojaron un tiempo promedio de respuesta del servicio: Productos de 53 ms y todos los retornos de los métodos en un estado correcto.

Prueba realizada al microservicio Servicio:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1:cmd
E:\sico\apis\productos>cd..
E:\sico\apis>cd servicios
E:\sico\apis\servicios\mocha

service
✓ debe retornar listado de servicios
✓ debe retornar servicio actualizado
✓ debe retornar servicio creado
✓ debe retornar servicio eliminado

4 passing (79ms)
E:\sico\apis\servicios
```

Figura 9. Prueba aplicada al sistema con la herramienta Mocha

Los resultados de la aplicación de la prueba arrojaron un tiempo promedio de respuesta del servicio: Servicios de 79 ms y todos los retornos de los métodos en un estado correcto.

### 3.2.2 Pruebas de aceptación

Las pruebas de aceptación que la metodología XP propone, también llamadas pruebas del cliente, son especificadas por el cliente y se centran en las características y funcionalidad generales del sistema que son visibles y revisables por parte del cliente. Las pruebas de aceptación se derivan de las historias de los usuarios que se han implementado como parte de la liberación del software. (Pressman, 2010)

**Tabla 33 Prueba Registrar Usuario**

<b>Caso de prueba de aceptación</b>	
<b>Código:</b> HU9_P1	<b>Historia de usuario:</b> HU_9
<b>Nombre:</b> Registrar usuario	
<b>Descripción:</b> Prueba de la funcionalidad de registrar usuario	
<b>Condiciones de ejecución:</b> El usuario debe estar en la página de registro de usuario	
<b>Pasos de ejecución:</b> <ol style="list-style-type: none"> <li>1. Se ingresan los datos del usuario. (Nombre, Apellidos, Provincia, Municipio, Contacto, Correo, Usuario, Contraseña)</li> <li>2. Se selecciona la categoría (Consumidor o Proveedor).</li> <li>3. Se presiona el botón "Registrar".</li> <li>4. Aparece una notificación que dice: "Usuario Registrado"</li> </ol>	
<b>Resultados esperados:</b> Satisfactorio	
<b>Evaluación:</b> Bien	

**Tabla 34 Prueba Modificar Usuario**

<b>Caso de prueba de aceptación</b>	
<b>Código:</b> HU9_P2	<b>Historia de usuario:</b> HU_9
<b>Nombre:</b> Modificar usuario	
<b>Descripción:</b> Prueba de la funcionalidad de registrar usuario	
<b>Condiciones de ejecución:</b> El usuario debe estar autenticado en el sistema	
<b>Pasos de ejecución:</b> <ol style="list-style-type: none"> <li>1. Se presiona el botón "Editar perfil".</li> <li>2. Se editan los campos de su información personal que estime (Nombre, Apellidos, Municipio, Provincia, Imagen).</li> <li>3. Se presiona el botón "Enviar".</li> </ol>	

4. Aparece una notificación que dice: “Usuario actualizado”.
<b>Resultados esperados: Satisfactorio</b>
<b>Evaluación: Bien</b>

**Tabla 35 Prueba Listar Usuario**

<b>Caso de prueba de aceptación</b>	
<b>Código:</b> HU9_P3	<b>Historia de usuario:</b> HU_9
<b>Nombre:</b> Listar usuario	
<b>Descripción:</b> Prueba de la funcionalidad de listar usuario	
<b>Condiciones de ejecución:</b> El administrador debe estar autenticado	
<b>Pasos de ejecución:</b>	
<ol style="list-style-type: none"> <li>1. Se accede al panel de administración.</li> <li>2. Se presiona la pestaña “Usuarios a registrar”.</li> <li>3. Se obtiene el listado de los usuarios.</li> </ol>	
<b>Resultados esperados: Satisfactorio</b>	
<b>Evaluación: Bien</b>	

**Tabla 36 Prueba Eliminar Usuario**

<b>Caso de prueba de aceptación</b>	
<b>Código:</b> HU9_P4	<b>Historia de usuario:</b> HU_9
<b>Nombre:</b> Eliminar usuario	
<b>Descripción:</b> Prueba de la funcionalidad de eliminar usuario	
<b>Condiciones de ejecución:</b> El administrador debe estar autenticado.	
<b>Pasos de ejecución:</b>	
<ol style="list-style-type: none"> <li>1. Se accede al panel de administración.</li> <li>2. Se presiona la pestaña “Autorizar Registros”.</li> <li>3. Se obtiene el listado de los usuarios no registrados.</li> <li>4. Se marca o marcan los que se van a eliminar.</li> <li>5. Se presiona el botón “Eliminar”.</li> <li>6. Aparece una ventana de confirmación con el texto: “Usuarios eliminados”</li> </ol>	
<b>Resultados esperados: Satisfactorio</b>	
<b>Evaluación: Bien</b>	

**Tabla 37 Prueba Autenticar Usuario**

<b>Caso de prueba de aceptación</b>	
<b>Código:</b> HU10_P1	<b>Historia de usuario:</b> HU_10
<b>Nombre:</b> Autenticar usuario	
<b>Descripción:</b> Prueba de la funcionalidad de autenticar usuario	
<b>Condiciones de ejecución:</b> El usuario debe estar en la pestaña de autenticación	
<b>Pasos de ejecución:</b> <ol style="list-style-type: none"> <li>1. Introducir correo y contraseña.</li> <li>2. Presionar el botón “Entrar”.</li> <li>3. Aparece una notificación de: “Bienvenido”</li> </ol>	
<b>Resultados esperados:</b> Satisfactorio	
<b>Evaluación:</b> Bien	

**Tabla 38 Prueba Registrar Producto**

<b>Caso de prueba de aceptación</b>	
<b>Código:</b> HU1_P1	<b>Historia de usuario:</b> HU_1
<b>Nombre:</b> Registrar producto	
<b>Descripción:</b> Prueba de la funcionalidad de registrar producto	
<b>Condiciones de ejecución:</b> El usuario debe estar autenticado con el rol de proveedor	
<b>Pasos de ejecución:</b> <ol style="list-style-type: none"> <li>1. El usuario accede a la pestaña de productos.</li> <li>2. Selecciona la opción “Añadir producto”.</li> <li>3. Rellena el formulario con los datos del producto (Nombre, Descripción, Disponibilidad, Precio, Imagen).</li> <li>4. Presiona el botón “Publicar”.</li> <li>5. Aparece una notificación de: “Producto creado”.</li> </ol>	
<b>Resultados esperados:</b> Satisfactorio	
<b>Evaluación:</b> Bien	

**Tabla 39 Prueba Modificar Producto**

<b>Caso de prueba de aceptación</b>	
<b>Código:</b> HU1_P2	<b>Historia de usuario:</b> HU_1
<b>Nombre:</b> Modificar producto	
<b>Descripción:</b> Prueba de la funcionalidad de mostrar producto	

<b>Condiciones de ejecución:</b> El usuario debe estar autenticado con el rol de proveedor
<b>Pasos de ejecución:</b> <ol style="list-style-type: none"> <li>1. El usuario accede a la pestaña de productos.</li> <li>2. Se seleccionan un producto en el botón “Detalles”.</li> <li>3. Se selecciona la opción “Editar”.</li> <li>4. Se modifican los campos del producto que se deseen.(Nombre, Descripción, Disponibilidad, Precio, Imagen)</li> <li>5. Se presiona el botón “Enviar”.</li> <li>6. Aparece una notificación de: “Producto Actualizado”.</li> </ol>
<b>Resultados esperados:</b> Satisfactorio
<b>Evaluación:</b> Bien

**Tabla 40 Prueba Listar Producto**

<b>Caso de prueba de aceptación</b>	
<b>Código:</b> HU1_P3	<b>Historia de usuario:</b> HU_1
<b>Nombre:</b> Listar producto	
<b>Descripción:</b> Prueba de la funcionalidad de listar productos.	
<b>Condiciones de ejecución:</b> El usuario debe estar autenticado con el rol de proveedor o consumidor	
<b>Pasos de ejecución:</b> <ol style="list-style-type: none"> <li>1. Se accede a la pestaña de productos.</li> <li>2. Se presiona el botón “Buscar”.</li> <li>3. Aparecen todos los productos.</li> </ol>	
<b>Resultados esperados:</b> Satisfactorio	
<b>Evaluación:</b> Bien	

**Tabla 41 Prueba Eliminar Producto**

<b>Caso de prueba de aceptación</b>	
<b>Código:</b> HU1_P4	<b>Historia de usuario:</b> HU_1
<b>Nombre:</b> Eliminar producto	
<b>Descripción:</b> Prueba de la funcionalidad de eliminar producto	
<b>Condiciones de ejecución:</b> El usuario debe estar autenticado con el rol de proveedor	
<b>Pasos de ejecución:</b> <ol style="list-style-type: none"> <li>1. El usuario accede a la pestaña de productos.</li> <li>2. Se seleccionan un producto en el botón “Detalles”.</li> </ol>	



<ol style="list-style-type: none"> <li>3. Se selecciona la opción “Eliminar”.</li> <li>4. Aparece una notificación de “Producto Eliminado”.</li> </ol>
<b>Resultados esperados: Satisfactorio</b>
<b>Evaluación: Bien</b>

**Tabla 42 Prueba Registrar Servicio**

<b>Caso de prueba de aceptación</b>	
<b>Código:</b> HU2_P1	<b>Historia de usuario:</b> HU_2
<b>Nombre:</b> Registrar servicio	
<b>Descripción:</b> Prueba de la funcionalidad de registrar servicio	
<b>Condiciones de ejecución:</b> El usuario debe estar autenticado con el rol de proveedor	
<b>Pasos de ejecución:</b>	
<ol style="list-style-type: none"> <li>1. El usuario accede a la pestaña de servicios.</li> <li>2. Selecciona la opción “Añadir servicio”.</li> <li>3. Rellena el formulario con los datos del servicio. (Nombre, Descripción, Precio, Días reservados, Imagen)</li> <li>4. Presiona el botón “Publicar”.</li> <li>5. Aparece una notificación de: “Servicio creado”.</li> </ol>	
<b>Resultados esperados: Satisfactorio</b>	
<b>Evaluación: Bien</b>	

**Tabla 43 Prueba Modificar Usuario**

<b>Caso de prueba de aceptación</b>	
<b>Código:</b> HU2_P2	<b>Historia de usuario:</b> HU_2
<b>Nombre:</b> Modificar servicio	
<b>Descripción:</b> Prueba de la funcionalidad de mostrar servicio	
<b>Condiciones de ejecución:</b> El usuario debe estar autenticado con el rol de proveedor	
<b>Pasos de ejecución:</b>	
<ol style="list-style-type: none"> <li>1. El usuario accede a la pestaña de servicios.</li> <li>2. Se seleccionan un servicio en el botón “Detalles”.</li> <li>3. Se selecciona la opción “Editar”.</li> <li>4. Se modifican los campos del servicio que se deseen. (Nombre, Descripción, Días reservados, Imagen)</li> <li>5. Se presiona el botón “Enviar”.</li> </ol>	

6. Aparece una notificación de: “Servicio Actualizado”.
<b>Resultados esperados: Satisfactorio</b>
<b>Evaluación: Bien</b>

**Tabla 44 Prueba Listar Servicio**

<b>Caso de prueba de aceptación</b>	
<b>Código:</b> HU2_P3	<b>Historia de usuario:</b> HU_2
<b>Nombre:</b> Listar servicio	
<b>Descripción:</b> Prueba de la funcionalidad de listar servicios.	
<b>Condiciones de ejecución:</b> El usuario debe estar autenticado como proveedor o consumidor	
<b>Pasos de ejecución:</b>	
<ol style="list-style-type: none"> <li>1. Se accede a la pestaña de servicio.</li> <li>2. Se presiona el botón “Buscar”.</li> <li>3. Aparecen todos los servicios.</li> </ol>	
<b>Resultados esperados: Satisfactorio</b>	
<b>Evaluación: Bien</b>	

**Tabla 45 Prueba Eliminar Servicio**

<b>Caso de prueba de aceptación</b>	
<b>Código:</b> HU2_P4	<b>Historia de usuario:</b> HU_2
<b>Nombre:</b> Eliminar servicio	
<b>Descripción:</b> Prueba de la funcionalidad de eliminar servicio	
<b>Condiciones de ejecución:</b> El usuario debe estar autenticado con el rol de proveedor	
<b>Pasos de ejecución:</b>	
<ol style="list-style-type: none"> <li>1. El usuario accede a la pestaña de servicios.</li> <li>2. Se seleccionan un servicio en el botón “Detalles”.</li> <li>3. Se selecciona la opción “Eliminar”.</li> <li>4. Aparece una notificación de “Servicio Eliminado”.</li> </ol>	
<b>Resultados esperados: Satisfactorio</b>	
<b>Evaluación: Bien</b>	

**Tabla 46 Prueba Autorizar Registro**

<b>Caso de prueba de aceptación</b>	
<b>Código:</b> HU3_P1	<b>Historia de usuario:</b> HU_3

<b>Nombre:</b> Autorizar Registro
<b>Descripción:</b> Prueba de la funcionalidad de autorizar registro
<b>Condiciones de ejecución:</b> El administrador debe estar autenticado
<b>Pasos de ejecución:</b> <ol style="list-style-type: none"> <li>1. Se accede al panel de administración.</li> <li>2. Se presiona la pestaña “Usuarios a registrar”.</li> <li>3. Se obtiene el listado de los usuarios.</li> <li>4. Se presiona el botón “Autorizar”.</li> <li>5. El usuario desaparece de la lista.</li> </ol>
<b>Resultados esperados:</b> Satisfactorio
<b>Evaluación:</b> Bien

**Tabla 47 Prueba Requerir Compra de Producto**

<b>Caso de prueba de aceptación</b>	
<b>Código:</b> HU4_P1	<b>Historia de usuario:</b> HU_4
<b>Nombre:</b> Requerir compra de producto.	
<b>Descripción:</b> Prueba de la funcionalidad de requerir compra de producto.	
<b>Condiciones de ejecución:</b> El usuario debe estar autenticado con el rol de consumidor	
<b>Pasos de ejecución:</b> <ol style="list-style-type: none"> <li>1. Se accede a la pestaña de productos.</li> <li>2. Se presiona el botón “Buscar”.</li> <li>3. Aparecen todos los productos.</li> <li>4. Se ingresa al producto mediante el botón detalles.</li> <li>5. Se selecciona la cantidad a adquirir menor o igual a la disponible.</li> <li>6. Se presiona el botón “Adquirir”</li> <li>7. Se muestra una notificación de “Producto adquirido”</li> </ol>	
<b>Resultados esperados:</b> Satisfactorio	
<b>Evaluación:</b> Bien	

**Tabla 48 Prueba Requerir Alquiler de Servicio**

<b>Caso de prueba de aceptación</b>	
<b>Código:</b> HU5_P1	<b>Historia de usuario:</b> HU_5
<b>Nombre:</b> Requerir compra de servicio.	
<b>Descripción:</b> Prueba de la funcionalidad de requerir compra de servicio.	

<b>Condiciones de ejecución:</b> El usuario debe estar autenticado con el rol de consumidor
<b>Pasos de ejecución:</b> <ol style="list-style-type: none"> <li>1. Se accede a la pestaña de servicios.</li> <li>2. Se presiona el botón “Buscar”.</li> <li>3. Aparecen todos los servicios.</li> <li>4. Se ingresa al servicio mediante el botón detalles.</li> <li>5. Se seleccionan los días que se solicita el alquiler del servicio.</li> <li>6. Se presiona el botón “Alquilar”.</li> <li>7. Se muestra una notificación de “Servicio Alquilado”.</li> </ol>
<b>Resultados esperados:</b> Satisfactorio
<b>Evaluación:</b> Bien

**Tabla 49 Prueba Evaluar Producto**

<b>Caso de prueba de aceptación</b>	
<b>Código:</b> HU6_P1	<b>Historia de usuario:</b> HU_6
<b>Nombre:</b> Evaluar producto	
<b>Descripción:</b> Prueba de la funcionalidad de evaluar producto	
<b>Condiciones de ejecución:</b> El usuario debe estar autenticado con el rol de consumidor	
<b>Pasos de ejecución:</b> <ol style="list-style-type: none"> <li>1. Se accede a la pestaña de productos.</li> <li>2. Se presiona el botón “Buscar”.</li> <li>3. Aparecen todos los productos.</li> <li>4. Se ingresa al producto mediante el botón detalles.</li> <li>5. Se ingresa la evaluación.</li> </ol>	
<b>Resultados esperados:</b> Satisfactorio	
<b>Evaluación:</b> Bien	

**Tabla 50 Prueba Notificar reserva**

<b>Caso de prueba de aceptación</b>	
<b>Código:</b> HU8_P1	<b>Historia de usuario:</b> HU_8
<b>Nombre:</b> Notificar reserva	
<b>Descripción:</b> Prueba de la funcionalidad de notificar reserva	
<b>Condiciones de ejecución:</b> El usuario debe estar autenticado con el rol de proveedor y algún consumidor haber requerido la reserva de un producto o un servicio.	

<b>Pasos de ejecución:</b> <ol style="list-style-type: none"> <li>1. Se accede a la pestaña de notificaciones.</li> <li>2. Se muestran las notificaciones de las reservas.</li> </ol>
<b>Resultados esperados: Satisfactorio</b>
<b>Evaluación: Bien</b>

A continuación, se muestra un gráfico con las pruebas unitarias realizadas por cada una de las iteraciones, las fallidas y los errores arreglados en cada una. (Ver Figura 10)



Figura 10. Pruebas unitarias por cada iteración

### 3.3 Conclusiones parciales

Tras haber desarrollado una de las fases más importantes de la metodología XP se arribó a las siguientes conclusiones parciales:

- El plan de entrega fue cumplido acorde al tiempo planificado para el desarrollo de las tareas por cada historia de usuario.
- A partir de la realización de las pruebas unitarias se verificó el correcto funcionamiento de los métodos de cada uno de los microservicios y se obtuvo sus tiempos de respuestas promedios.
- Las pruebas de aceptación permitieron que el cliente comprobara el estado de cada una de las funcionalidades desarrolladas en la aplicación.
- El resultado de las pruebas procuró conocer cada una de las no conformidades, que luego, fueron atendidas correctamente, logrando un adecuado funcionamiento de la aplicación ante cualquier situación.

## **Conclusiones generales**

Una vez culminado el presente trabajo de diploma donde se comprende la gestión de productos y servicios entre clientes y proveedores estatales en Cuba, se puede concluir que se le dio cumplimiento al objetivo general de la investigación, resaltando que:

1. La elaboración del marco teórico conceptual de la investigación permitió sentar las bases de la misma. Se realizó un estudio de soluciones en el mundo que contienen el proceso de compra-venta de productos y servicios, aportando ideas a la aplicación que posteriormente fue desarrollada.
2. El diseño empleando la metodología XP permitió identificar los requisitos de software y la descripción de las Historias de Usuario de la propuesta de solución para un mejor detalle de los procesos y funcionalidades de la aplicación desarrollada.
3. Las pruebas permitieron detectar y corregir los errores durante la implementación, posibilitando cumplir con las especificaciones requeridas y la validación de la aplicación implementada.

De esta manera se concluye que, al desarrollar un sistema de gestión de productos y servicios entre clientes y proveedores empresariales, se inicia y facilita el trabajo a cada una de las empresas estatales de Cuba que participan en el proceso de clientes y proveedores, además de cumplirse los principios de soberanía e independencia tecnológica.

## Recomendaciones

A lo largo de la presente investigación fueron identificados elementos que no se contemplaron en el desarrollo de la solución al no ser parte del alcance inicial. Sin embargo, se considera que son relevantes e incrementarían la utilidad del sistema y la calidad de uso del mismo por lo que se recomienda:

A los futuros desarrolladores que trabajen en la herramienta:

- Implementar nuevas funcionalidades que permitan un intercambio más directo entre consumidores y proveedores mediante chats.
- Implementación de pasarelas de pago electrónico como PayPal y la vinculación con la banca móvil para mayor facilidad en las transacciones.
- Implementación de un sistema de mapas vinculado a GoogleMaps o LeafLet, aprovechando el empleo de microservicios en el desarrollo del sistema.

A la XETID:

- Continuar el desarrollo del sistema agregando nuevas funcionalidades que permitan un intercambio más directo entre consumidores y proveedores, así como la vinculación con la banca móvil para mayor facilidad en las transacciones.

## Referencias bibliográficas

- A, Essau. 2016.** ¿Qué somos? [En línea] 2016. <https://amazon.com>.
- Alfonso, Fernando. 2017.** SlideShare. *SlideShare*. [En línea] 22 de Mayo de 2017.
- AngularJS. 2019.** AngularJS: Developer Guide: Introduction. *AngularJS*. [En línea] 2019. <https://docs.angularjs.org/guide/introduction>.
- Antiñanco, Javier Matias. 2017.** *Bases de Datos NoSQL: escalabilidad y alta disponibilidad a través de patrones de diseño*. 2017.
- Bray. 2014.** *The JavaScript Object Notation (JSON) Data Interchange Format*. 2014.
- Camargo-Ortega, JF. 2015.** *Knowing the big data*. 2015.
- Diaz-Canel, Miguel. 2019.** *Realidades y Propuestas del E-commerce en Cuba*. 2019.
- Cuervo, Víctor. 2019.** AiT. *ArquitectoIt*. [En línea] 2019. <http://www.arquitectoit.com/postman/que-es-postman/>.
- ed, Microsoft. 2017.** Visual Studio Official Site. *Visual Studio*. [En línea] 2017.
- Fernández, Martha Mulet. 2019.** Disponible en Internet web cubana dedicada a clasificados y publicidad. Órgano oficial del PCC. *Granma*. 2019.
- Finley, Klint. 2014.** *The Ensembl REST API: Ensembl Data for Any Language*. 2014.
- González, P. 2015.** *Desarrollo de una API para la descripción y gestión de servicios Web REST*. 2015.
- Joskowicz, José. 2008.** *Reglas y Prácticas en eXtreme Programming*. 2008.
- Mackdonald. 2016.** *Introduction to HTML 5. The Visual Language*. 2016.
- Mattson, Jacob. 2010.** 2010.
- Molina, Tatiana. 2018.** Angular: Arquitectura del framework . *Angular Chile*. [En línea] Google, 26 de Septiembre de 2018. [Citado el: 22 de Mayo de 2019.] <https://medium.com/angular-chile/angular-arquitectura-del-framework-e46204f38fef>.
- Montoya, Elba Rosa Pérez. 2019.** Realidades impuestas por el E-commerce en Cuba. *Granma*. 2019.
- Nielsen, D. 2016.** *Investigate availability and maintainability within amicroservice architecture* . 2016.
- Pressman, Roger S. 2010.** 2010.
- RAE-ASALE. 2019.** «Diccionario de la lengua española» - Edición del Tricentenario. [En línea] 2019. <https://dle.rae.es/>.
- Reshuán, S Puente. 2018.** *Seguridad de aplicaciones web basadas en las tecnologías Node. js y MongoDB*. 2018.
- Ruiz, Alejandro López. 2016.** Rankia. *Comercio electrónico: definición, tipos y tendencias*. [En línea] 2016. <https://www.rankia.co/blog/mejores-cdts/3808106-comercio-electronico-definicion-tipos-tendencias>.



**Schetnikovich, Dmitry. 2019.** Robo 3t. *Robo 3t.* [Online] 2019. <https://robomongo.org/>.

**Schmuller, J. 2013.** *Aprendiendo UML en 24 horas.* 2013.

**Somerville. 2005.** 2005.

**Villamizar, M. 2017.** *Evaluating the monolithic and the microservice architecture pattern to deploy web applications in the cloud.* *Computing Colombian Conference (10CCC.* Colombia : s.n., 2017.

**Wolff, E. 2016.** *Microservices:Flexible SoftwareArchitectures.* *CreateSpace Independent Publishing Platfor.* 2016.

**Zambrano, Omar. 2017.** Catálogo SCRIBD. *SCRIBD.* [En línea] 2017.

## Bibliografía

«¿Qué es y para qué sirve una API? Aquí las respuestas necesarias». 2019. 15 de febrero de 2019. <https://web.archive.org/web/20190215165536/https://blog.pandorafms.org/es/para-que-sirve-una-api/>.

«Análisis y evaluación de las arquitecturas REST y SOAP para el desarrollo de servicios web aplicados al ERP AdrisERP y su versión móvil en Android». s. f. Accedido 23 de enero de 2019. <http://repositorio.uss.edu.pe/handle/uss/4067>.

«AngularJS: Developer Guide: Introduction». s. f. Accedido 4 de febrero de 2019. <https://docs.angularjs.org/guide/introduction>.

Maya, Edgar, y Daniel López. 2018. «Arquitectura de Software basada en Microservicios para Desarrollo de Aplicaciones Web».

«Disponible en Internet web cubana dedicada a clasificados y publicidad › Cuba › Granma - Órgano oficial del PCC». s. f. Accedido 16 de febrero de 2019. <http://www.granma.cu/cuba/2015-08-11/disponible-en-internet-web-cubana-dedicada-a-clasificados-y-publicidad>.

Nielsen, D. Investigate availability and maintainability within amicroservice architecture (Tesis Doctoral). Master's thesis, Aarhus University. (2015).

Fowler, M.; Lewis, J. Microservices. Viittattu, (2014)

Borčín T. Service activity monitoring for SilverWare. Masaryk University. (2017)

Newman, S. Building Microservices. O'Reilly Media, Inc. (2015).

Wolff, E. Microservices: Flexible Software Architectures. CreateSpace Independent Publishing Platform. (2016)

Antiñanco, Matías Javier. s. f. «Bases de Datos NoSQL: escalabilidad y alta disponibilidad a través de patrones de diseño». Accedido 10 de febrero de 2019. <http://sedici.unlp.edu.ar/handle/10915/36338>.

González, Puerta, y José Manuel. 2015. «Desarrollo de una API para la descripción y gestión de Servicios Web REST», noviembre. <http://repositori.uji.es/xmlui/handle/10234/156006>.

Roger S. Pressman, Ph.D. 2010. «INGENIERÍA DEL SOFTWARE. UN ENFOQUE PRÁCTICO Séptima edición». The McGraw-Hill Companies, Inc.

«Modelo vista controlador (MVC)». s. f. Accedido 14 de mayo de 2019. <https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>.

Newman, S. Building Microservices. O'Reilly Media, Inc. (2015).

Valverde, F., Pastor, O. Dealing with REST services in model-driven web engineering methods. V Jornadas Científico -Técnicas en Servicios Web y SOA, JSWEB, 243–250. (2009)

Lith, Adam; Mattson, Jakob (2010). «Investigating storage solutions for large data: A comparison of well performing and scalable data storage solutions for real time extraction and batch insertion of data» (PDF).

NoSQL Relational Database Management System: Home Page». Strozzi.it. 2 de octubre de 2007.

Finley, Klint (25 de enero de 2011). «Wait, What's Node.js Good for Again? ». ReadWriteWeb (en inglés). Archivado desde el original el 21 de junio de 2013. Consultado el 16 de marzo de 2019.

## Anexos

Carta de Aceptación emitida por el cliente:

 **CARTA DE ACEPTACIÓN**

En cumplimiento con la fase de desarrollo y en función de la ejecución del proyecto: Sistema de Gestión para la interrelación entre proveedores y consumidores empresariales utilizando la arquitectura de microservicios, se hace entrega de los productos que se relacionan a continuación:

- Sistema de Gestión para la interrelación entre proveedores y consumidores empresariales utilizando la arquitectura de microservicios.

Entrega	Recibe
Nombre y apellidos: Roberto Camejo Domínguez	Nombre y apellidos: Ing. Andrés Sellés González
Cargo: Tesista	Cargo: Jefe de Centro
Firma: 	Firma: 

Fecha: 20 de junio de 2019

