Universidad de las Ciencias Informáticas

Facultad 2

Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas

Microservicios para la obtención de información de los productos que oferta la red de ventas minorista en Cuba basado en la arquitectura de microservicios.

Autor: Michel Damian Collazo Baños

Tutores:

Msc. Jenisley Verde Acosta

Ing. José Alexei Leyva Desdín

La Habana, 2019

"Año 61 de la Revolución

Pensamiento

Declaración de autoría

Se declara que Michel Damian Collazo Baños es el único autor de este trabajo, que tiene por título: Microservicios para la obtención de información de los productos que oferta la red de ventas minorista en Cuba basado en la arquitectura de microservicios y se autoriza a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.
Para que así conste firmo la presente a los ___ días del mes de _____ del año 2019.

_____
Firma del autor.
Michel Damian Collazo

_____
Firma del tutor.
MSc. Jenisley Verde Acosta

_____
Firma del tutor.
Ing. José Alexei Leyva Desdín

Datos de contacto:

Tutor: MSc. Jenisley Verde Acosta

Universidad de las Ciencias Informáticas

Email: jeni@uci.cu

Tutor 2: Ing. José Alexei Leyva Desdín

Empresa de Tecnologías de la Información para la Defensa

Email: jaleyva@xetid.cu

Agradecimientos

A mi familia:

- *A mi madre Lucia por todo el apoyo en todo momento y situación, por confiar en mí y, sobre todo por el amor que siempre me ha brindado pase lo que pase.*
- *A mi hermana por su cariño y paciencia.*
- *A mi abuelita del alma por todo lo que me aporta.*
- *A mi papá Jorge por su apoyo.*
- *A los demás de mi familia que no pueden estar.*

A los profesores que me han ayudado en mi formación en el transcurso de la carrera.

A la profe Jenisley por su incondicional apoyo en todo momento.

A las personas especiales que me ayudaron a poder terminar esta tesis como se debe hacer.

A las personas que me dieron su apoyo incondicional en los momentos más críticos.

A Lissi y Jenni por su ayuda.

De forma general a todo el que se preocupó porque la tesis lograra una feliz terminación.

Dedicatoria

A mi familia en especial a mi madre Lucia.

- *A mi papá.*
- *A mi abuela.*
- *A mi hermana.*

Resumen

En el ámbito comercial la utilización de la tecnología ha traído grandes ventajas en los mercados internacionales, ya que permite las relaciones comerciales entre el mercado y los consumidores. Cuba se encuentra inmersa en un proceso de avance tecnológico e informatización de la sociedad, con el que la población se está viendo beneficiada. Por medio de estudios realizados por la Empresa de Tecnologías de la Información para la Defensa se ha podido detectar que, a pesar del desarrollo alcanzado en el país con el uso de las tecnologías las aplicaciones para la gestión de información de productos de la red de ventas minoristas cuentan con limitantes en su desarrollo por el uso de su arquitectura monolítica .El presente trabajo describe el desarrollo de microservicios para la obtención de información de los productos que oferta la red de ventas minorista en Cuba empleado la arquitectura de microservicios. En el documento se describen los conceptos fundamentales utilizados, además de realizarse un estudio sobre aplicaciones similares en el ámbito nacional e internacional. Durante el desarrollo se utilizaron como lenguajes de programación JavaScript 1.6, además de HTML 5 como lenguaje de etiquetado, CSS 3 como lenguaje de diseño gráfico, Angular 6 como marco de trabajo y como herramientas de modelado el Visual Paradigm para UML 8.0, el gestor de bases de datos PostgreSQL 9.4, el administrador de bases de datos PgAdmin 3, servidor de aplicaciones Node 8.9.3 y el editor de texto Visual Code 1.28.0. La implementación de estos microservicios permiten la generación de información de los productos que se están ofertando en las distintas tiendas minoristas del país. Además de la posibilidad de convertirse en un microservicios genérico para cualquier sistema de gestión de información de productos.

Palabras clave: gestión de información, microservicios, productos, red de ventas minorista.

Summary

In the commercial field, the use of technology has brought great advantages in international markets, since it allows commercial relationships between the market and consumers. Cuba is immersed in a process of technological advancement and computerization of society, with which the population is benefiting. Through studies carried out by the Information Technology Company for Defense, it has been possible to detect that, despite the development achieved in the country with the use of technologies, the applications for the management of product information of the sales network retailers have limitations in their development due to the use of their monolithic architecture. The present work describes the development of microservices to obtain information on the products offered by the retail sales network in the country using the microservices architecture. The document describes the fundamental concepts used, in addition to a study on similar applications in the national and international scope. During development, JavaScript 1.6 programming languages were used, as well as HTML 5 as a labeling language, CSS 3 as a graphic design language, Angular 6 as a working framework and Visual Paradigm for UML 8.0 as the modeling tools. databases PostgreSQL 9.4, the database administrator PgAdmin 3, application server Node 8.9.3 and the text editor Visual Code 1.28.0. The implementation of microservices allows the generation of information on the products that are being offered in the different retail stores in the country for use in different product information management systems from now on.

Keywords: information management, microservices, products, retail sales network.

Tabla de contenido

Índice de tablas

Índice de Diagramas

Índice de Imagen

Introducción

El comercio inicia cuando el hombre comienza a vivir en sociedad, una vez que se establece un núcleo familiar y comienzan a crearse comunidades. Los orígenes del comercio se remontan a finales del Neolítico, cuando se descubrió la agricultura, pasando por diferentes etapas de desarrollo hasta llegar a la era tecnológica donde aparece la tendencia a la compra-venta de productos y servicios a través de medios electrónicos e informáticos (Viollaz, 2014).

Con la aparición de las nuevas tecnologías de la información y las comunicaciones (TICs) comienzan a automatizarse los principales procesos llevados a cabo por el hombre para un mejor uso de los recursos tanto económicos como relacionados a su vida diaria y su evolución. (Gubern, 2010).

La tecnología aporta grandes beneficios a la humanidad, su papel principal es crear mejores herramientas útiles para simplificar el ahorro de tiempo y esfuerzo de trabajo. Las TIC ejercen importantes repercusiones tanto sobre las organizaciones como sobre las relaciones que se establecen entre ellas. No sólo están revolucionando la manera de organizar y coordinar la distribución, reduciendo los costes de los flujos de marketing y generando nuevas producciones de servicios, sino que además llevan a redefinir los límites naturales de los mercados, modificar las reglas básicas de la competencia, reformular el alcance de las actividades comerciales y proporcionar un nuevo conjunto de armas competitivas (ACM, 2014).

Por otro lado, estas tecnologías también han propiciado la generación de cambios diversos en los mercados. En concreto, la proliferación de diversos sistemas y tecnologías de TIC de uso común por parte del consumidor en conjunción con los cambios en las estructuras sociales, demográficas y económicas de la población, han provocado una serie de alteraciones en las pautas y hábitos de compra tradicionales de los consumidores. Estas transformaciones indudablemente han ejercido una notable influencia sobre la evolución y aparición de nuevos formatos comerciales, que se han visto forzados a ofrecer un surtido de productos y servicios más adaptado a las necesidades de los consumidores finales (Salazar, 2016).

Desde esta perspectiva la globalización y los avances tecnológicos, generan beneficios en su posicionamiento, agilización de procesos dentro y fuera de la empresa, ingreso en el mundo virtual, búsqueda de información de una forma más ágil y rápida, ofrecer sus productos y/o servicios de una forma diferente por medio de la red, y visualizar la Internet como el futuro de la tecnología y de la comunicación. (ACM, 2014) La integración basada en redes de información mejora su funcionamiento y crea valor para sus clientes y sus socios, este concepto no se aplica solamente a empresas virtuales cuyas actividades están en su totalidad basadas en la Web, sino también está enfocado a negocios tradicionales.

En poco tiempo los avances de la Internet han producido un rápido aumento de tiendas y las compras en línea convirtiendo esto en ventaja para los propietarios de tiendas que les brinda la posibilidad de ofertar sus productos desde cualquier parte del mundo y al usuario le permitirá acceder a la información de este desde un ordenador conectado a la red de redes sin importar su ubicación geográfica.

Cuba no se ha quedado atrás en el uso de las TICs empleándolas en diferentes esferas como la salud, educación, cultura, industrias, etc. Encontrándose en la actualidad inmersa en un proceso de avance tecnológico e informatización de la sociedad, con el que la población se está viendo beneficiada, teniendo acceso a servicios como correo electrónico, transmisión de archivos, consulta a páginas Web, comercio electrónico, publicaciones periódicas en texto completo, boletines electrónicos, sistemas de conferencias, foros electrónicos (UCI, 2018).

Miguel Díaz-Canel Bermúdez, Presidente de los Consejos de Estado y Ministros, expresó este 17 de diciembre de 2018 en la asamblea nacional del poder popular, que la informatización de la sociedad cubana ya es una realidad, es un proceso que avanza paso a paso.

La Empresa de Tecnologías de la Información para la Defensa (XETID), empresa militar que trabaja en el desarrollo de soluciones informáticas para automatizar la defensa del país. Se rige por la política de informatización de la sociedad cubana principalmente en el sector de las fuerzas armadas revolucionarias. Sus áreas temáticas abarcan procesos de la gestión del capital humano, la planificación, gestión de suministros, la seguridad en las entidades y las comunicaciones entre otros temas que se pueden consultar en el sitio oficial de la entidad (XETID, 2018).

Por medio de estudios realizados por esta empresa, se detecta que en ella existen limitaciones en el desarrollo de aplicaciones informática de forma ágil para el momento actual de informatización de la sociedad cubana. El uso de aplicaciones monolíticas para la promoción, ofertas y ventas traería desventajas para el desarrollo productivo, ya que la construcción de sistemas con esta arquitectura al necesitar un cambio en alguna funcionalidad trae consigo un relanzamiento de la aplicación en su conjunto y aunque la arquitectura es muy rígida no puede ser fácilmente actualizada, pues un error comprometería el código en su conjunto. Además, por su estructura de código único no da la posibilidad de trabajar en ambientes simultáneamente.

Partiendo de esta situación se determina como **Problema a resolver**: Cómo brindar información a la población, de los productos que oferta la red de ventas minorista del país.

Como **objeto de estudio** de esta investigación se tiene: La gestión de la información de productos.

Para dar solución a la problemática planteada, la presente investigación tiene como **Objetivo general**: Desarrollar microservicios para la obtención de información de los productos que oferta la red de ventas minorista en Cuba empleando la arquitectura de microservicios.

Cuyo **campo de acción** está enmarcado en: La arquitectura de microservicios.

Las **preguntas científicas** que guían y orientan el desarrollo del proceso investigativo son las siguientes:

- ¿Cómo desarrollar microservicios para la obtención de información de los productos que oferta la red de ventas minorista en Cuba?
- ¿Cuáles son los referentes teóricos a tener en cuenta para abordar la solución del problema planteado relacionado con la gestión de la información de productos?
- ¿Qué propuesta de solución se define para implantar?
- ¿Cómo se valida el correcto funcionamiento de la solución?

Para dar solución al problema planteado y dar respuesta a las preguntas científicas formuladas se proponen las siguientes **tareas de investigación**:

- Análisis de los principales conceptos asociados a la gestión de la información de productos para obtener los fundamentos teóricos necesarios para el desarrollo de la solución.

- Estudio de los sistemas homólogos para identificar tendencias actuales relacionadas con las funcionalidades y mecanismos utilizados.

- Investigación y selección de la metodología de software, herramientas informáticas y tecnologías para su posterior utilización en el desarrollo de los microservicios.

- Definición de los requisitos funcionales y no funcionales para la implementación de los microservicios.

- Análisis de los métodos y técnicas para la realización de pruebas de software que permitan comprobar el correcto funcionamiento de los microservicios.

Métodos de la investigación:

Los métodos científicos de investigación son la forma de abordar la realidad, de estudiar la naturaleza, la sociedad y el pensamiento, con el propósito de descubrir su esencia y sus relaciones (Sampieri, 2010). Se clasifican en teóricos y empíricos. En esta investigación se utilizaron:

Métodos teóricos:

Analítico-Sintético: Para analizar y extraer de la documentación especializada los aspectos referentes a la gestión de la información de productos y a los microservicios.

Modelación: Se utiliza en la modelación de los diagramas correspondientes a la etapa de análisis, diseño e implementación de la solución a desarrollar en la presente investigación.

Métodos empíricos:

Observación: Se utilizó en dos momentos en la investigación. El primer momento para determinar el problema por el cual se realiza esta investigación y nuevamente se emplea para comprender la forma en que se realiza el proceso que se investiga en la actualidad.

El trabajo de diploma presenta como **novedad** el uso de la arquitectura de microservicios para la gestión de información de productos y como **aporte práctico** la creación de microservicios que permiten la generación de información de los productos que se están ofertando en las distintas tiendas minoristas del país. Además de la posibilidad de convertirse en un microservicios genérico para cualquier sistema de gestión de información de productos.

Para lograr una mejor organización y lectura el trabajo de diploma se estructuró de la siguiente manera: Resumen, Introducción, 3 Capítulos, Conclusiones, Recomendaciones, Referencias Bibliográficas y Anexos, donde se abarca todo lo relacionado con la investigación realizada.

- Capítulo 1 Fundamentación teórica: Se definen los conceptos más relevantes relacionados con la gestión de la información de productos. Se abordan los conceptos y aspectos más importantes para la investigación. Además, se realiza una descripción de las tecnologías a emplear como lenguaje de programación y se analizan las principales herramientas para el diseño, modelado e implementación de la solución.
- Capítulo 2 Diseño de la propuesta de solución: Se especifican los requisitos funcionales que debe cumplir la propuesta, además de la descripción de estos. También se muestran los modelos y diagramas diseñados, la explicación de la arquitectura del sistema y los patrones de diseño utilizados.
- Capítulo 3 Implementación y validación de la propuesta de solución: En este capítulo se abarca todo lo relacionado con la implementación de la solución y las pruebas realizadas a la propuesta de solución, en pos de verificar la calidad y efectividad de acuerdo con las necesidades del cliente.

Capítulo I Fundamentación teórica

1. Introducción

En el presente capítulo se detallan los elementos teóricos que apoyan la investigación y el desarrollo del tema, abordándose los conceptos y aspectos más importantes para la investigación. Se realiza un estudio detallado de sistemas existentes a nivel nacional e internacional que brinden información de los productos que venden. También se exponen las principales características de la metodología, herramientas, lenguajes y tecnologías informáticas que se emplearán en el desarrollo de la solución.

1.1. Definiciones asociadas a la investigación

Gestión

Del latín gestĭo, el concepto hace referencia a la acción y a la consecuencia de administrar o gestionar algo. Al respecto, hay que decir que gestionar es llevar a cabo diligencias que hacen posible la realización de una operación comercial o de un anhelo cualquiera. La noción de gestión, por lo tanto, se extiende hacia el conjunto de trámites que se llevan a cabo para resolver un asunto o concretar un proyecto. La gestión es también la dirección o administración de una compañía o de un negocio. Su objetivo primordial el conseguir aumentar los resultados óptimos de una industria o compañía (Los procesos de gestión, 2014).

Información de productos

La información de producto es la base de la cadena de suministro y de las aplicaciones de venta minorista en todos los sectores. Hace referencia a los procesos y las tecnologías que se centran en gestionar la información de productos de forma centralizada (por ejemplo, en varias cadenas de suministros). Es esencial que se utilicen las definiciones de producto, la documentación, los atributos y los identificadores compartidos. En los varios módulos de una solución empresarial, la información y la configuración específica de un producto son necesarias para administrar procesos de negocio relacionados con productos, familias de productos o categorías de productos específicas (Guerra, 2014).

Gestión de información de producto

La gestión de información de productos funciona con una definición de producto, una clasificación e identificadores compartidos entre todas las entidades jurídicas, y también de configuraciones específicas de un producto que se ajusten a los procesos empresariales. papel interdisciplinario, tendiendo un puente entre las brechas dentro de la empresa entre equipos de diferentes conocimientos, entre los que destacan los equipos orientados a la ingeniería y los equipos orientados comercialmente. Esta información de los productos también puede tener uno o más informes directos que administren tareas operacionales y/o un gestor de cambios que puedan supervisar nuevas iniciativas. La fabricación es separada de la función de la investigación, el encargado del producto tiene la responsabilidad de tender un puente sobre los boquetes si cualesquiera existen (Ruesta, 2010).

Red de ventas minoristas

Red de ventas

Son todas aquellas personas que, perteneciendo a la empresa o vinculadas a la misma, se ocupan de todas aquellas tareas relacionadas directamente con la venta de los productos fabricados o distribuidos por la empresa (Hernandez, 2009). Normalmente la red de ventas se compone de tres equipos principales:

- ✓ Equipo interior de ventas
- ✓ Equipo exterior de ventas
- ✓ El equipo de exportación

Red de ventas minoristas

La red de venta minorista se define como la empresa comercial o persona en régimen de autónomo que vende productos al consumidor final. Son el último eslabón del canal de distribución, el que está en contacto con el mercado. Las ventas minoristas pueden alterar, frenando o potenciando, las acciones de mercadotecnia y micro mercadotecnia de los fabricantes y mayoristas. Son capaces de influir en las ventas y resultados finales de los artículos que comercializan (Salem, 2010).

Arquitectura de microservicios.

En la actualidad, a nivel empresarial y tanto en el sector privado como público se desarrollan software para suplir las necesidades de automatización de procesos internos, este desarrollo ha seguido las tendencias impuestas por la plataforma, lenguaje de programación o por la experiencia del área de desarrollo (Lopez, 2012).

El término microservicios no es relativamente nuevo, este estilo arquitectural fue acuñado por Martin Fowler en un taller de arquitectos de software como una descripción del nuevo campo que los participantes estaban explorando. No existe una definición en concreto para microservicio, sin embargo, una aproximación para esta la da (Newman, 2015) como: “Pequeños servicios autónomos que trabajan juntos”.

La Arquitectura de microservicios, conocida por las siglas MSA (del inglés Microservices Architecture) es una aproximación para el desarrollo software que consiste en construir una aplicación como un conjunto de pequeños servicios, los cuales se ejecutan en su propio proceso y se comunican con mecanismos ligeros. Cada servicio se encarga de implementar una funcionalidad completa del negocio. Cada servicio es desplegado de forma independiente y puede estar programado en distintos lenguajes y usar diferentes tecnologías de almacenamiento de datos (Microservices a definition of this new architectural term, 2014).

La utilización de arquitecturas basadas en microservicios, supone un nuevo punto de vista en el desarrollo de aplicaciones web completamente diferente al que se ha estado desarrollando en los últimos años, durante los cuales las aplicaciones web se han desplegado generalmente siguiendo arquitecturas monolíticas. Esta propone, sin embargo, una arquitectura en la que cada funcionalidad quede dividida en un nuevo servicio web lo más independiente posible, lo cual va a mejorar en gran medida los puntos débiles de las aplicaciones monolíticas (Maya, 2017).

Los microservicios se comunican a través de la interfaz de programación de aplicaciones (API). Hay una distinción importante entre Apis para microservicios que se invocan dentro de los límites del sistema (APIs internas) y llamadas API que se emiten desde fuera (APIs externas). Todas estas API deben gestionarse, pero dependiendo del contexto empresarial y arquitectónico, pueden existir diferencias entre la gestión de la API para las API internas y externas. (Bortenschlager, 2016)

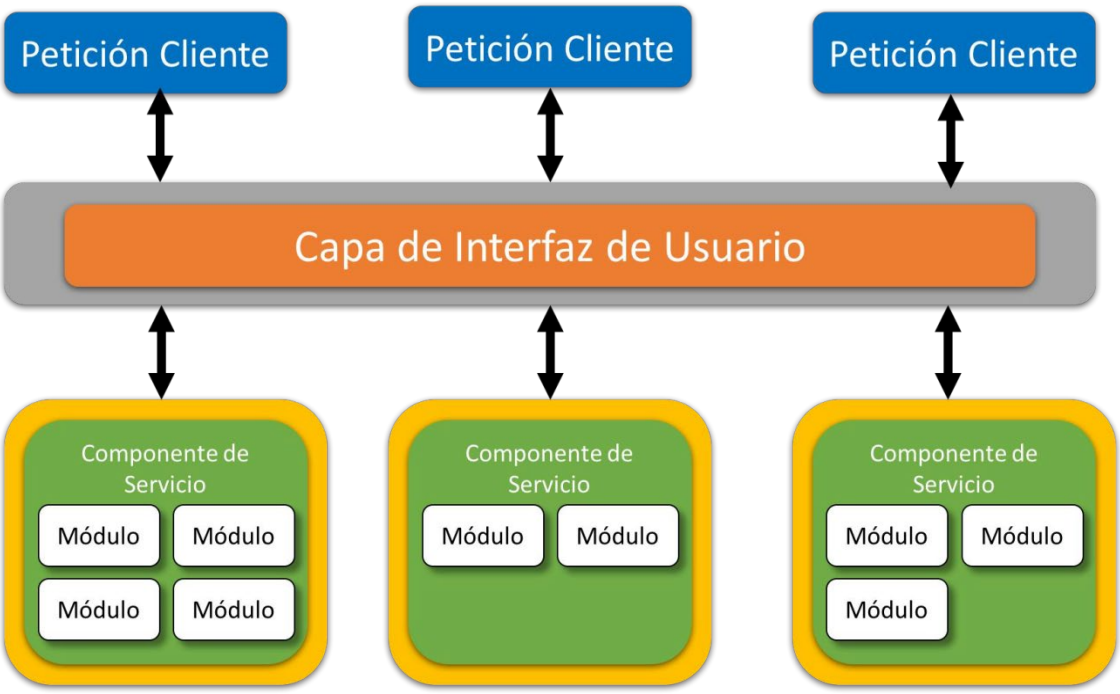


Imagen 1: Esquema general de la arquitectura de microservicios tomado de internet

Entre las ventajas de utilizar microservicios se tiene 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. La agilidad, reducción de costes y la escalabilidad granular, traen algunos retos de los sistemas distribuidos y las prácticas de gestión de los equipos de desarrollo que deben ser considerados. (Villamizar, 2015)

Entre los principales beneficios del uso de esta tecnología se pueden observar en la imagen a continuación:

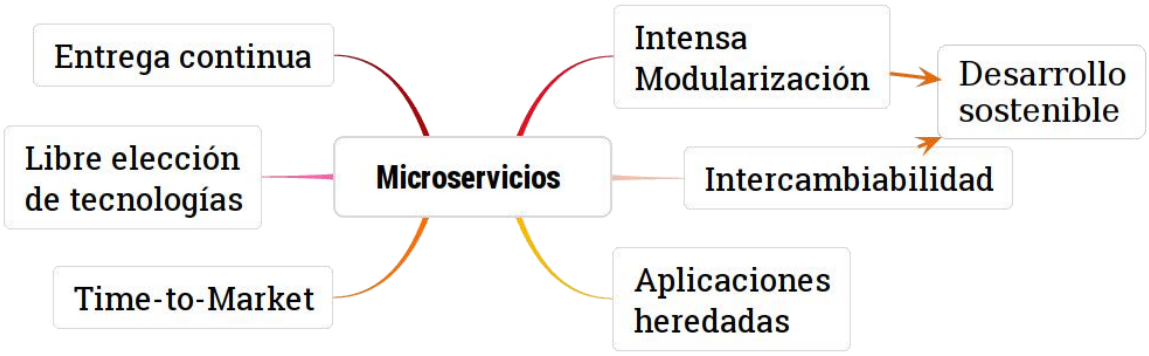


Imagen 2: Beneficio del uso de microservicios, tomada de internet

1.2. Estado del arte

Se realiza una búsqueda y estudio de sistemas informáticos similares a la solución propuesto en la investigación, tanto en el ámbito internacional como nacional con el fin de utilizarlo como guía para comprender los procesos de gestión de la información manejada en cada uno de ellos.

Sistemas internacionales

Comprando

Es la primera red social de productos y precios de Argentina. Entre sus funcionalidades está la de permitir ver y compartir información de productos y precios en cualquier punto de venta y compararlos en tiempo real, con información no solo de precios claros, sino de la compartida por toda la comunidad en cualquier tipo de comercio y punto del país. Además, entre otras características esta que permite al usuario tomar decisiones de compra con toda la información de precios y productos, también la búsqueda, escaneo o compartimiento de la información de todos los productos de los comercios, además de ubicar al cliente al establecimiento más cercano. Se desarrolló sobre el sistema operativo Android. Esta desarrollado bajo software privativo (Comprando, 2019)

Tiendeo

Entre sus funcionalidades principales esta que la aplicación que toma la ubicación del cliente y busca los supermercados más cercanos por la zona, permite pulsar sobre estos y ver el folleto de ofertas válido en ese momento. Además de brindar la posibilidad ver donde

se encuentra más barato aquello que busca el cliente. Una de las características distintivas es que también ofrece información sobre los horarios, teléfono y la dirección exacta. Además, permite la opción de añadir tiendas a favoritas y recibir alertas cuando los folletos se actualicen. Es desarrollada sobre el sistema operativo Android. Su software es privativo. (Gutierrez, 2019)

EBay

Cuenta con aplicaciones móviles completas que permiten que los compradores pueden navegar, buscar y pujar mediante un par de toques, mientras que los vendedores pueden añadir un producto solamente en segundos. Proporciona avisos de ofertas del día, nuevos listados de productos, pero siempre se pueden terminar las transacciones mediante pagos seguros además de rastrear los envíos personales. Su aplicación principal comprende varios servicios autónomos, y cada uno ejecuta la lógica propia de cada área funcional que se ofrece a los clientes, ya que esta sobre la arquitectura de microservicios. Es de software privativo y de fácil acceso a través de los navegadores web. (Ebay, 2019)

Amazon Mobile

Permite a los usuarios navegar por las listas, comparar precios, leer comentarios, compartir productos con amigos, hacer compras. Esta aplicación permite utilizar el escáner de código de barras para revisar el precio y la disponibilidad de cualquier artículo mientras se está comprando, haciendo que todas las configuraciones se sincronicen para ordenar fácilmente. Hizo su migración hace 3 años para la arquitectura de microservicios siendo unas de las grandes compañías que la implementan en producción, puede recibir grandes cantidades de solicitudes a diario gracias a sus Apis de servicios. Es de software privativo y de fácil acceso a través de cualquier navegador web (Amazon, 2019)

Sistemas nacionales

Donde hay

En la actualidad el país cuenta con la aplicación para móvil **Donde hay,** la cual es una de las iniciativas nacidas dentro del proceso de informatización desarrollado actualmente en la isla. Concebido para sistema Android, el software permite al usuario localizar determinado producto en las tiendas de la Red Panamericana del Grupo Empresarial CIMEX.

Una vez seleccionado el elemento de búsqueda, Dónde hay ofrece a los usuarios diversos datos, entre ellos, el nombre y ubicación del establecimiento que posee en venta el producto deseado, así como la cantidad existente del elemento seleccionado. Permite averiguar en tiempo real cuál es el mejor precio disponible luego de buscar en las bases de datos disponibles en línea indicando los precios e información sobre productos similares que se venden en Internet o en las tiendas. Desarrollado bajo el principio de software libre y de fácil acceso a su aplicación (Cimex, 2019).

Tienda en línea de 5ta. Y 42

Otra aplicación novedosa es la apertura de la tienda en línea de 5ta y 42, en su versión digital, experiencia que debe extenderse, según directivos de la Cadena de Tiendas Caribe, al menos una tienda por provincia en el primer semestre del año próximo (Prensa Latina, 2018). La dirección para acceder a esta plataforma ya se encuentra visible en la red de redes, los requisitos para la compra en línea son: tener acceso a la red nacional, disponer de una tarjeta magnética en CUP, así como poseer una tarjeta del servicio de banca electrónica (Telebanca).Desarrollado bajo el principio de software libre y fácil acceso desde cualquier navegador web (XETID, 2019)

Superfácil

También es una oferta atractiva Superfácil**,** prestación creada por Citmatel. Pensada como un centro comercial virtual, dispone de varias tiendas para la compra de contenidos digitales cubanos, soluciones para negocios propios y pagos de servicios a Citmatel. Las opciones de pago son similares a la tienda de 5ta y 42. Permite averiguar en tiempo real cuál es el mejor precio disponible luego de buscar en las bases de datos disponibles en línea indicando los precios e información sobre productos similares que se venden en Internet o en las tiendas. (Cuba,más digital y conectada, 2018) .

Tabla 1:Resultado del estudio del arte

Aplicaciones	Búsqueda	Navegabilidad	Tipo de software	Arquitectura	Tipo de aplicación	Ubicación de la tienda	Cantidad de productos
EBay	si	si	privativo	microservicios	web	no	no
Amazon Mobile	si	si	privativo	microservicios	web	no	no
Donde Hay	si	si	libre	-	apk	si	no
Tienda 5ta y 42	si	si	libre	-	web	si	no
Superfácil	si	si	libre	-	web	si	no
Comprando	si	si	privativo	-	apk	si	no
Tiendeo	si	si	privativo	-	apk	si	no

Resultado del estudio del estado del arte

Después del análisis realizado a diferentes aplicaciones de promoción y ventas de productos en línea, tanto nacionales, como internacionales, se pudo conocer el funcionamiento de estos sistemas con el objetivo de tenerlos en cuenta en el desarrollo de esta investigación. Se determinó no tomar ninguna de estas aplicaciones ya que la empresa XETID planea la construcción de un ecosistema de microservicios para lo que se enfoca en tomar las siguientes funcionalidades y características relevantes que sirven para el desarrollo de la propuesta de solución:

- ✓ Navegación.
- ✓ Búsqueda.
- ✓ Cantidad del producto existente en las tiendas.
- ✓ Ubicación del establecimiento que posee el producto.

1.3. Metodología de desarrollo de software

Metodología de desarrollo de software es un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo en sistemas de información. Es el conjunto integrado de técnicas y métodos que permite abordar de forma homogénea y abierta cada una de las actividades del ciclo de vida de un proyecto de desarrollo (Comunicación, 2009).

El desarrollo de software no es una tarea fácil. Prueba de ello es que existen numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo. Por una parte, se tienen aquellas propuestas más tradicionales que se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, y las herramientas y notaciones que se usarán. Estas propuestas han demostrado ser efectivas y necesarias en un gran número de proyectos, pero también han presentado problemas en muchos otros (Páez, 2014), ejemplo de ellas son Rational Unified Process (RUP) y Microsoft Solution Framework (MSF).

Por otra parte, se encuentran las metodologías ágiles las cuales dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas. Este enfoque está mostrando su efectividad en proyectos con requisitos muy cambiantes y cuando se exige reducir drásticamente los tiempos de desarrollo, pero manteniendo una alta calidad (Páez, 2014) entre ellas se pueden encontrar a Extreme Programing (XP), SCRUM, Agile Unified Process (AUP) y Proceso Unificado Ágil para el Desarrollo de la Actividad Productiva en la Universidad de las Ciencias Informáticas (AUP-UCI).

Para el desarrollo de esta investigación se determinó emplear la metodología ágil, AUP-UCI. Dicha metodología fue estandarizada en la Universidad de las Ciencias Informáticas, para guiar el proceso de desarrollo de software en sus distintos centros productivos.

1.4. Metodología de desarrollo a utilizar

AUP es una versión simplificada del Proceso Unificado de Software (RUP). Describe de una forma fácil de entender una aproximación al desarrollo de software empresarial mediante técnicas ágiles manteniéndose fiel a las técnicas y conceptos definidos por RUP. Las técnicas ágiles empleadas por AUP incluyen desarrollo dirigido por pruebas, modelado ágil, gestión de cambios ágil y refactorización de base de datos para mejorar la productividad. La metodología AUP-UCI responde a una variación que se realiza a la metodología ágil AUP.

Esta metodología de tipo ágil divide el ciclo de vida de la producción del software en tres fases (Sánchez, 2015):

Inicio: Fase donde se realiza un estudio inicial de la organización cliente que permite obtener una información fundamental acerca del alcance del proyecto, realizar estimaciones de tiempo, esfuerzo y costo y decidir si se ejecuta o no el proyecto.

Ejecución: Fase donde se ejecutan las actividades requeridas para desarrollar el software, incluyendo el ajuste de los planes del proyecto considerando los requisitos y la arquitectura. Durante el desarrollo se modela el negocio, obtienen los requisitos, se elaboran la arquitectura y el diseño, se implementa y se libera el producto.

Cierre: Fase donde se analizan tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto.

AUP-UCI propone para el ciclo de vida de los proyectos siete disciplinas, donde los flujos de trabajo son los siguientes: Modelado de negocio, Requisitos, Análisis y diseño, Implementación, Pruebas internas, Pruebas de liberación y Pruebas de aceptación. Además, está compuesta por cuatro escenarios (Sánchez, 2015):

Escenario No 1: Proyectos que modelen el negocio con casos de uso del negocio solo pueden modelar el sistema con casos de uso del sistema.

Escenario No 2: Proyectos que modelen el negocio con modelo conceptual solo pueden modelar el sistema con casos de uso del sistema.

Escenario No 3: Proyectos que modelen el negocio con descripción de procesos de negocio solo pueden modelar el sistema con descripción de requisitos por procesos.

Escenario No 4: Proyectos que no modelen negocio solo pueden modelar el sistema con historias de usuario.

Se selecciona para esta investigación el escenario número dos porque este aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan que no es necesario incluir las responsabilidades de las personas que ejecutan las actividades, de esta forma modelan exclusivamente los conceptos fundamentales del negocio. Se recomienda este escenario para proyectos donde el objetivo primario es la gestión y presentación de información.

1.5. Lenguajes y herramientas utilizadas

Para la realización del módulo se deben tener en cuenta que las tecnologías y herramientas a utilizar sean de fácil manejo y total eficiencia. Se deben utilizar

tecnologías en constante evolución para que este cumpla con los requerimientos establecidos para su uso y actualización.

Lenguajes utilizados

JavaScript versión 1.6

JavaScript es lo que se conoce como lenguaje script, es decir: se trata de código de programación que se inserta dentro de un documento. JavaScript fue desarrollado por la empresa Netscape con la idea de potenciar la creación de páginas Web dinámicas para su navegador (Sánchez, 2013).

Lenguaje de marcado de hipertexto (HTML) versión 5

El lenguaje HTML es un estándar reconocido en todo el mundo y cuyas normas define un organismo sin ánimo de lucro llamado consorcio mundial de la red, más conocido como W3C. Como se trata de un estándar reconocido por todas las empresas relacionadas con el mundo de Internet, una misma página HTML se visualiza de forma muy similar en cualquier navegador de cualquier sistema operativo. El propio W3C define el lenguaje HTML como "un lenguaje reconocido universalmente y que permite publicar información de forma global" (Perez, 2014)

Lenguaje de hoja de estilo (CSS) versión 3

Se utiliza el estilo CSS el cual es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas. Una vez creados los contenidos, se utiliza el lenguaje CSS para definir el aspecto de cada elemento: color, tamaño y tipo de letra del texto, separación horizontal y vertical entre elementos, posición de cada elemento dentro de la página (Perez, 2014).

Lenguaje de modelado y herramienta

Lenguaje de modelado unificado (UML)

Es un lenguaje estándar para escribir diseños de software. Puede usarse para visualizar, especificar, construir y documentar los artefactos de un sistema de software intensivo (Pressman, 2010). En la investigación se empleó el UML para la construcción del modelo conceptual y se trabajó utilizando una herramienta CASE.

La ingeniería de software asistida por computadora (CASE, por sus siglas en inglés) consiste en diversas aplicaciones que pueden ayudar en el ciclo de vida de desarrollo de software. Visual Paradigm es una herramienta de diseño y administración poderosa, multiplataforma y fácil de usar para los sistemas de tecnología de información. Ofrece a los desarrolladores de software una plataforma de desarrollo innovadora para crear aplicaciones de calidad de manera más rápida. Facilita una excelente interoperabilidad la mayoría de los entornos de desarrollo integrado (Sommerville, 2012).

Sistema gestor de bases de datos

PostgreSQL versión 9.4

PostgreSQL, es un sistema de gestión de base de datos relacional y libre, que agiliza la interacción de cliente, servidor y base de datos, donde postgreSQL es el que realiza la mayoría del trabajo referente a bases de datos cuando se le hacen peticiones. La estabilidad, potencia, robustez, facilidad de administración e implementación de estándares han sido las características que más se han tenido en cuenta durante su desarrollo. PostgreSQL funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema. Se considera extensible ya que su código fuente está disponible para todos sin costo, es multiplataforma y fue diseñado para ambientes de alto volumen (PostgesSQL, 2019).

Administrador de bases de datos

PgAdmin 3 es una aplicación gráfica para la administración del gestor de base de datos PostgreSQL. Es compatible con plataformas como: Linux, Solaris, Mac OS y Windows. Presenta una interfaz que permite la gestión de la información de una forma más rápida. Crear desde simples consultas SQL hasta consultas de una alta complejidad. Permite trabajar eficientemente con las tablas de la base de datos. Soporta grandes volúmenes de información y presenta una amplia documentación. Es una herramienta de código abierto respaldada por una amplia comunidad de desarrolladores que encaminan sus esfuerzos al perfeccionamiento de la misma. (PgAdmin, 2019)

Entornos de desarrollo

Visual Studio Code versión 1.28.0: Visual Studio Code es un editor de código fuente ligera pero potente que se ejecuta en escritorio y está disponible para Windows, MacOS y Linux. Viene con soporte incorporado para JavaScript, Typescript y Node.js y tiene un rico ecosistema de extensiones para otros idiomas (como C ++, C #, Java, Python, PHP,

Go) y tiempos de ejecución (como .NET y Unity). Posee depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código y fragmentos (Code, 2019).

Node.js versión 8.9.3: Concebido como un entorno de ejecución de JavaScript orientado a eventos asíncronos, Node está diseñado para construir aplicaciones en red escalables. Es de código abierto para la capa de servidor en una arquitectura basada en eventos y basado en el motor V8 de Google (NodeJs, 2019).

Marco de trabajo

Angular versión 6

Angular 6 es un marco de JavaScript para crear aplicaciones y aplicaciones web en JavaScript, HTML y Typescript, que es un súper conjunto de JavaScript. Angular proporciona funciones integradas para animación, servicio http y materiales que, a su vez, incluyen funciones como autocompletar, navegación, barra de herramientas, menús, etc. El código está escrito en Typescript, que se compila en JavaScript y se muestra en el navegador (Basalo, 2013).

Conclusiones del capítulo

Luego de desarrollar el capítulo se ha podido arribar a las siguientes conclusiones parciales:

- El estudio de los principales conceptos permitió una mejor comprensión de la presente investigación.
- El estudio de sistemas homólogos permitió identificar las características principales de los sistemas de compra y ventas de productos en línea tanto internacionales como nacionales.
- El estudio de las tecnologías, herramientas, lenguajes y metodologías proporcionó los conocimientos necesarios para su utilización en el desarrollo de la solución propuesta.

Capítulo II Diseño de la propuesta de solución

2. Introducción

El presente capítulo tiene como objetivo describir la propuesta de solución, precisar las características con que contará el módulo e implementarlas. Se evidencian las disciplinas Modelado de negocio, Requisitos y Análisis, diseño e Implementación que propone la metodología AUP-UCI para el ciclo de vida del proyecto. Para ello se realiza una representación del modelo conceptual, se definen los requisitos funcionales y no funcionales del módulo. Se presenta el diagrama de casos de uso; así como su descripción para una mayor comprensión.

2.1. Descripción de la propuesta de solución

Para dar solución a la situación problemática planteada en esta investigación se propone como solución: la creación de microservicios que permitan la obtención de información de los productos que oferta la red de ventas minorista en Cuba basado en la arquitectura de microservicios con la conformación de pequeños módulos que trabajan de forma autónoma permitiendo la división de las funcionalidades como servicios web independientes. Esta cuenta con dos tipos de usuarios: el usuario proveedor que después de estar registrado y autenticado es el encargado de toda la gestión (insertar, modificar, eliminar y listar) referentes a los productos a ofertar en la red de ventas y ver la información que se almacene de los productos de su entidad y el usuario administrador que autenticado controla todo el flujo de información que se genere en la solución con respecto a los productos y los proveedores registrados y ver la información general de todos los productos y registros. Los microservicios que se desarrollan cuentan con su base de datos propia e independiente y los servicios a su vez son inyectados al componente mediante la API que controla el funcionamiento de cada uno de estos, para así tener mejor distribución de los recursos. Entre los microservicios construidos está, el que gestiona los productos, llamado **Producto**, este a su vez contiene toda la información de estos para su posterior gestión por parte de proveedores y administrador y sus datos están almacenados en una base de datos llamada productos. Además, se encuentra el microservicio **Registro**, el que gestiona la información del registro de todos los proveedores, almacenado en la base de datos de registros.

2.2. Modelo Conceptual

Un modelo conceptual (MC) proporciona una visión estructural que puede ser completada por los modelos de casos de uso. En él se describen las distintas entidades, atributos, papeles y relaciones, además de las restricciones que rigen el dominio del problema (Jacobson, 1998). El MC define un vocabulario y es útil como herramienta de comunicación, ayuda al entendimiento del negocio entre los equipos, añadiendo precisión y enfoque en el tema a tratar.

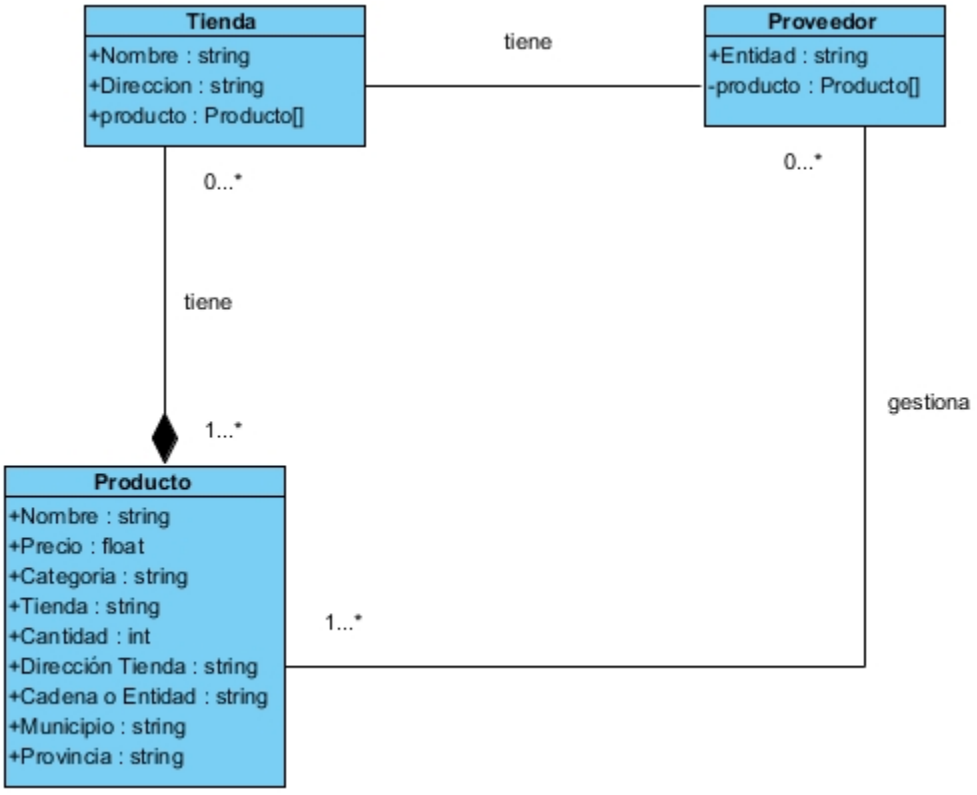


Diagrama 1: Modelo Conceptual elaboración propia

2.3. Diccionario de datos

Un diccionario de datos es una colección de nombres, definiciones y atributos sobre los elementos de datos que se utilizan o capturan en una base de datos, sistema de información o parte de un proyecto de investigación. Describe los significados y los propósitos de los elementos de datos dentro del contexto de un proyecto, y proporciona orientación sobre la interpretación, los significados aceptados y la representación (Pressman, 2010). En la tabla 1 se evidencia la descripción del concepto Producto, el resto se puede consultar en el Anexo 1 Diccionario de datos:

Tabla 2: Diccionario de datos (Producto)

Descripción	Esta entidad representa las propiedades de un producto en el módulo.					
Atributos						
Nombre	**Descripción**	**Tipo**	**¿Puede ser nulo?**	**¿Es único?**	**Restricciones**	
					Clases válidas	**Clases no válidas**
Identificador	Valor único que identifica a cada producto.	Número entero	No	Si	Toma valores que se obtengan de la combinación de los dígitos del [0-9].	Puede tomar como valor máximo 2*10^9.
Nombre del producto	Nombre propio del producto	Cadena de caracteres.	No	No	Puede tomar cualquier combinación de letras [a-z], [A-Z], espacio. Admite hasta 50 caracteres.	No admite caracteres especiales.
Precio	Precio del producto	Número Real.	No	No	Toma valores que se obtengan de la combinación de los dígitos del [0-9].	
Categoría	Categoría donde se encuentra ubicada el producto en oferta.	Cadena de caracteres.	No	Si	Puede tomar cualquier combinación de letras [a-z], [A-Z], espacio.	
Tienda o Entidad	Entidad o tienda proveedora.	Cadena de caracteres.	No	No	Puede tomar cualquier combinación de letras [a-z], [A-Z], espacio y números.	No debe comenzar con un número.

Cantidad	Cantidad en existencia del producto en oferta.		No	No	Toma valores que se obtengan de la combinación de los dígitos del [0-9].	
Dirección	Dirección donde se encuentra ubicada la entidad proveedora del producto.	Cadena de caracteres.	No	Si	Puede tomar cualquier combinación de letras [a-z], [A-Z], espacio y números.	
Provincia	Provincia donde se encuentra ubicada la entidad proveedora del producto.	Cadena de caracteres.	No	Si	Puede tomar cualquier combinación de letras [a-z], [A-Z], espacio.	
Municipio	Municipio donde se encuentra ubicada la entidad proveedora del producto.	Cadena de caracteres.	No	Si	Puede tomar cualquier combinación de letras [a-z], [A-Z], espacio.	

2.4. Requisitos funcionales

Los requisitos funcionales definen los componentes de un sistema de software, describen un conjunto de entradas, comportamientos, salidas y funcionalidades específicas que un sistema debe cumplir. Los requisitos funcionales son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que éste debe reaccionar a entradas y las salidas que debe devolver según situaciones particulares (Sommerville, 2016). En este trabajo se identificaron los siguientes requisitos funcionales:

Tabla 3: Tabla de requisitos funcionales

No	Requisito	Descripción
RF1	Autenticar usuario	El sistema deberá permitir a un usuario proveedor o administrador autenticarse. Los datos de entrada (**usuario-contraseña**)
RF2	Crear registro	El sistema deberá permitir a un usuario proveedor insertarse como entidad. Los datos de entrada (**entidad, cadena, usuario, contraseña**)
RF3	Editar registro	El sistema deberá permitir a un usuario proveedor modificar su registro. Los datos de entrada (**entidad, cadena, usuario, contraseña**)
RF4	Listar registro	El sistema deberá permitir a un usuario administrador ver el listado de las entidades registradas.
RF5	Crear producto	El sistema deberá permitir a un usuario proveedor insertar un nuevo producto que se vaya a ofertar. Los datos de entrada (**nombre del producto, precio, categoría, entidad, cantidad de producto, cadena, dirección, provincia, municipio**)
RF6	Modificar producto	El sistema deberá permitir a un usuario proveedor modificar un producto. Los datos de entrada(**nombre del producto, precio, categoría, entidad, cantidad de producto, cadena, dirección, provincia, municipio**)
RF7	Eliminar producto	El sistema deberá permitir a un usuario proveedor eliminar un producto.
RF8	Buscar productos	El proveedor o administrador podrá realizar búsquedas del producto que necesita. El dato de entrada (**nombre del producto**)
RF9	Mostrar detalles de los productos	El sistema deberá permitir a un usuario proveedor o administrador ver los detalles de los productos.
RF 10	Listar producto	El sistema deberá permitir a un usuario administrador o proveedor ver el listado los productos.

2.5. Requisitos no funcionales

Los requisitos no funcionales son atributos de calidad que debe poseer el sistema, especifica criterios que pueden usarse para juzgar la operación del mismo. Son aquellos requisitos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de este (Sommerville, 2016). Define propiedades como rendimiento, disponibilidad, seguridad, accesibilidad, usabilidad, estabilidad, interfaz, entre otros. Para este trabajo se acordaron los siguientes requisitos no funcionales:

Usabilidad

Utilizar el idioma español para los mensajes y textos de la interfaz.

El módulo debe proporcionar mensajes de error que sean informativos y orientados al usuario final.

Fiabilidad

El módulo contará con campos obligatorios, debidamente identificados para garantizar la integridad de la información que se introduce.

La información contenida dentro de la base de datos debe ser consistente en correspondencia con los datos reales de los productos y debe permanecer disponible en todo momento.

Portabilidad

Sistema operativo: Servidor de bases de datos relacional PostgreSQL 9.4 con memoria RAM: 2GB, Disco Duro: 500GB y microprocesador: i3, i5, i7.

Servidor de aplicaciones Node.js versión 8.9.3 con memoria RAM: 1GB, Disco Duro: 500GB y microprocesador: i3, i5, i7.

El sistema será accesible desde estaciones de trabajo de escritorio, laptop. Estos deberán contar con un navegador web como Mozilla Firefox versión 63.0b3 o Google Chrome versión 70.0.3538.110).

Seguridad

Se usan mecanismos de encriptación de los datos que por cuestiones de seguridad no deben estar en texto plano, por lo que las contraseñas en la base de datos se almacenarán de forma cifrada en el formato de md5.

2.6. Definición de los casos de uso

Los casos de usos son una técnica para la especificación de requisitos funcionales. Modela la funcionalidad del sistema tal como la perciben los agentes externos, denominados actores, que interactúan con el sistema desde un punto de vista particular (Jacobson, 1998).

CU 1. Gestionar Producto.

CU 2. Mostrar detalle del producto.

CU 3. Buscar Producto.

CU 4. Gestionar Registro.

CU 5. Autenticar Usuario.

2.7. Definición de los actores

El sistema cuenta con dos tipos de usuario:

Usuario proveedor: Este usuario es el encargado de gestionar (insertar, modificar y eliminar) la información de los productos que se ofertan de los mercados mediante el módulo, además tiene la facilidad de realizar una búsqueda de estos.

Usuario administrador: Este usuario controla todo el flujo de información que se genere y gestione en el módulo, además de tener acceso a todos los productos y registros

2.8. Diagrama de caso de uso del sistema

Un diagrama de casos de uso es una forma de diagrama de comportamiento UML mejorado (Rodríguez, 2013). El Lenguaje de Modelado Unificado (UML), define una notación gráfica para representar casos de uso llamada modelo de casos de uso. UML no define estándares para que el formato escrito describa los casos de uso, y así mucha gente no entiende que esta notación gráfica define la naturaleza de un caso de uso; sin embargo una notación gráfica puede solo dar una vista general simple de un caso de uso o un conjunto de casos de uso.

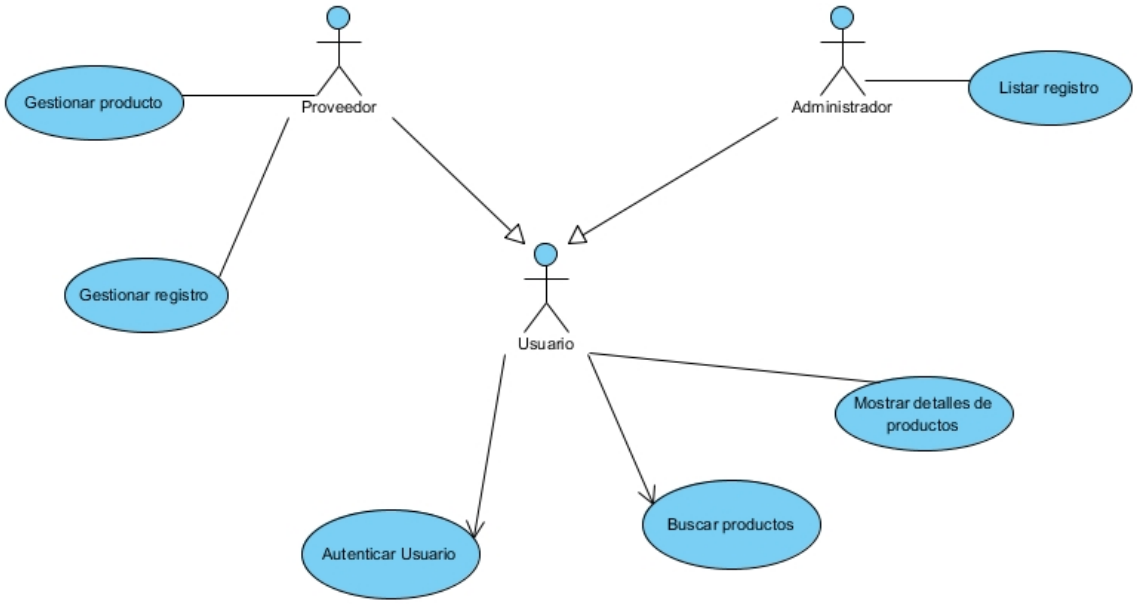


Diagrama 2: Diagrama de casos de uso de sistema. Elaboración propia.

2.9. Descripción de caso de uso del sistema

En la descripción de los casos de uso del sistema se evidencia el caso de uso, los actores involucrados y un breve resumen de este. La tabla 3 muestra el caso de uso del sistema Gestionar producto ver ejemplo en la tabla 3 El resto de los casos de uso se pueden consultar en el Anexo 2 Casos de uso del sistema

Tabla 4: CU: Gestionar producto

	Crear, modificar, listar o eliminar un producto.	
Actores	Proveedor	
Resumen	El caso de uso se inicia cuando el proveedor necesita crear, modificar, listar o eliminar un producto en el sistema.	
Complejidad	**Alta**	
Prioridad	**Critica**	
Precondiciones	El usuario proveedor debe estar autenticado en el sistema. Para las secciones Modificar y Eliminar producto debe existir al menos un producto creado en el sistema.	
Pos condiciones	Se almacenan los productos creados o modificados, permite listar los mismos y en cuanto al producto eliminado desaparece de manera permanente del sistema.	
Flujo de eventos		
Flujo básico Gestionar producto		
	Actor	**Sistema**
1	Crea, modifica, lista o eliminar un producto.	
2		Permite realizar varias acciones con el producto: - Crear un producto. Ver Sección 1: Crear Producto. - Modificar un producto. Ver Sección 2: Modificar Producto. - Listar un producto. Ver Sección 3: Listar Producto. - Eliminar un producto. Ver Sección 4: Eliminar Producto.
3		Termina el caso de uso.
Sección 1: “Crear Producto”		
Flujo básico Crear Producto		
	Actor	**Sistema**
1	Selecciona la opción Crear Producto.	
1.		El sistema muestra un formulario con los siguientes datos: Nombre

		Precio Categoría Nombre de la tienda Cantidad del producto Dirección de la tienda Cadena de tienda Municipio Provincia
2.	Inserta los valores	
3.	Presiona el botón Guardar	
4.		El sistema valida la información insertada
5.		El sistema crea el producto
6.		El sistema muestra el mensaje siguiente: “El producto se creó correctamente”
7.		Termina el Caso de Uso.
Flujos alternos		
4ª Presionar el botón Cancelar		
	Actor	**Sistema**
1	Presionar botón Cancelar	
2		El sistema cancela la acción y muestra la interfaz anterior.
3		Termina el Caso de Uso
5ª Existen campos obligatorios vacíos		
	Actor	**Sistema**
1		El sistema muestra encima del campo obligatorio, el mensaje “El campo no puede estar vacío”
2		Continuar en el paso 3 del flujo básico.
5b Existen campos incorrectos		
	Actor	**Sistema**
1		El sistema muestra el mensaje “El nombre es incorrecto”
2		Continuar en el paso 3 del flujo básico.
Sección 2: “Modificar Producto”		
Flujo básico Modificar Producto		
	Actor	**Sistema**
1	Presione el ícono Modificar del producto escogido	
2		El sistema muestra un formulario con los siguientes datos: Nombre

		Precio Categoría Nombre de la tienda Cantidad del producto Dirección de la tienda Cadena de tienda Municipio Provincia
3	Inserta los valores	
4	Presiona el botón Aceptar	
5		El sistema valida la información insertada
6		El sistema modifica el producto
7		Termina el Caso de Uso.
Flujos alternos		
4ª Presionar el botón Cancelar		
	Actor	Sistema
1	Presionar botón Cancelar	
2		El sistema cancela la acción y muestra la interfaz anterior.
3		Termina el Caso de Uso
5ª Existen campos obligatorios vacíos		
	Actor	Sistema
1		El sistema muestra encima del campo obligatorio, el mensaje “El campo no puede estar vacío”
2		Continuar en el paso 3 del flujo básico.
5b Existen campos incorrectos		
	Actor	**Sistema**
1		El sistema muestra el mensaje “El nombre es incorrecto”
2		Continuar en el paso 3 del flujo básico.
Sección 3: “Listar Producto”		
Flujo básico Listar Producto		
	Actor	Sistema
1	Selecciona la opción Listar Producto.	
2		Muestra la interfaz donde se listan los productos.
3		Terminar Caso de Uso.
Sección 4: “Eliminar Producto”		
Flujo básico Eliminar Producto		
	Actor	Sistema

1	Presiona el ícono “Eliminar” del producto escogido.	
2		El sistema muestra la interfaz de confirmación de eliminación.
3	Selecciona el botón “Eliminar”.	
4		El sistema elimina el producto escogido.
5		El sistema actualiza el listado de productos.
6		Termina el Caso de Uso.
Flujos alternos		
3ª Presionar el botón Cancelar		
	Actor	**Sistema**
1	Presionar botón Cancelar	
2		El sistema cancela la acción y muestra la interfaz anterior.
3		Termina el Caso de Uso

Relaciones	**CU incluidos**	N/A
	CU extendidos	N/A
Requisitos no funcionales	N/A	
Asuntos pendientes	N/A	

2.10. Técnicas de trazabilidad de requisitos

La matriz de trazabilidad de requisitos es un cuadro que vincula los requisitos del producto desde su origen hasta los entregables que los satisfacen. La implementación de una matriz de trazabilidad de requisitos ayuda a asegurar que cada requisito agrega valor al negocio, al vincularlo con los objetivos del negocio y del proyecto. Proporciona un medio para realizar el seguimiento de los requisitos a lo largo del ciclo de vida del proyecto, lo cual contribuye a asegurar que al final del proyecto se entreguen efectivamente los requisitos aprobados en la documentación de requisitos. Por último, proporciona una estructura para gestionar los cambios relacionados con el alcance del producto. (PMI, 2013)

En esta investigación se realizaron 4 tipos de matrices de trazabilidad: Matriz de trazabilidad Requisito Funcional - Concepto, Matriz de trazabilidad Requisito Funcional - Requisito Funcional, Matriz de trazabilidad Requisito Funcional – Caso de Uso y Matriz de trazabilidad Caso de Uso – Caso de Prueba.

A continuación, se enumeran los casos de uso del sistema para facilitar el entendimiento de la matriz de trazabilidad Requisito Funcional -Caso de Uso que se muestra en la Imagen 2. El resto de las matrices podrán ser consultadas en el Anexo 3 Matrices de trazabilidad.

CU 1. Gestionar Producto.

CU 2. Mostrar detalle del producto.

CU 3. Buscar Producto.

CU 4. Gestionar Registro.

CU 5. Autenticar Usuario.

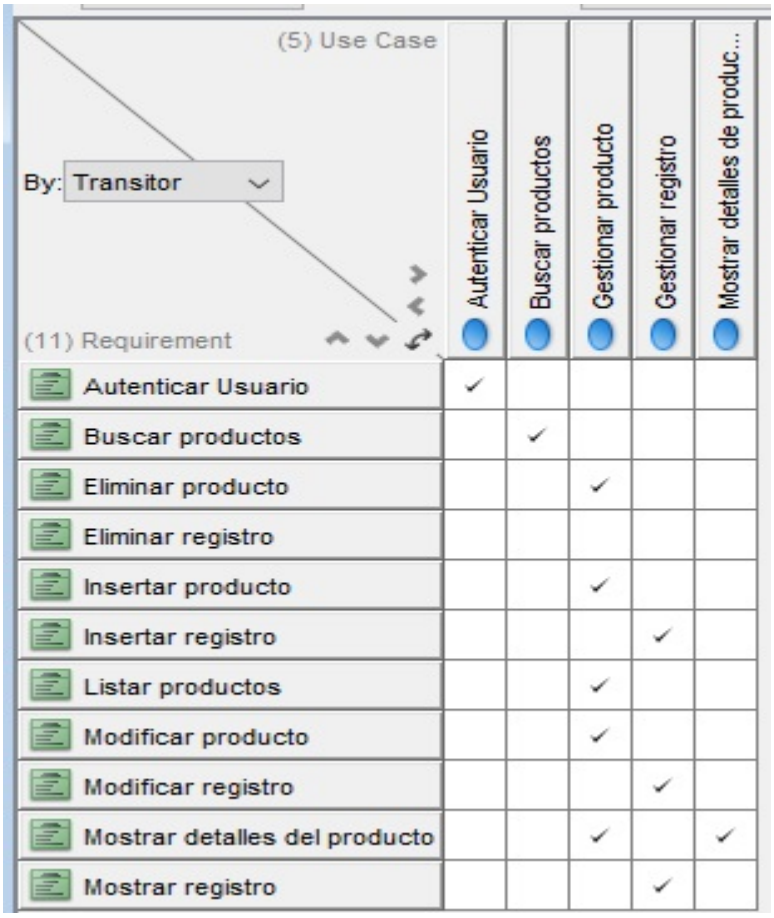


Imagen 3: Matriz de trazabilidad. Elaboración propia.

2.11. Modelo de datos

Un modelo de datos es una serie de conceptos que puede utilizarse para describir un conjunto de datos y las operaciones para manipularlos (Pressman, 2010). A

continuación, se presenta el modelo de datos de la investigación donde se observa que está constituido por 4 entidades y sus relaciones entre ellas: producto, registro, provincia, municipio.

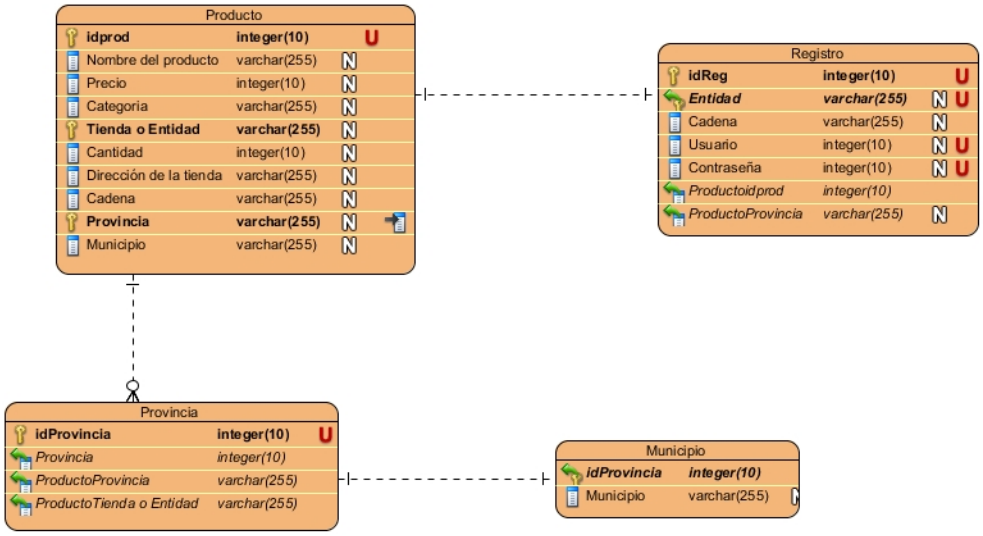


Imagen 4:Modelo de datos

2.12. Descripción del patrón arquitectónico

Un patrón arquitectónico, como un estilo de arquitectura, impone la transformación del diseño de una arquitectura. Sin embargo, un patrón difiere de un estilo en varias formas fundamentales: 1) el alcance del patrón es menos amplio y se centra en un aspecto de la arquitectura más que en el total de ésta, 2) un patrón impone una regla a la arquitectura, describe la manera en la que el software manejará ciertos aspectos de su funcionalidad en el nivel de la infraestructura (por ejemplo, la concurrencia) [Bos00], 3) los patrones arquitectónicos tienden a abocarse a aspectos específicos del comportamiento en el contexto de la arquitectura (por ejemplo, cómo manejarán la sincronización o las interrupciones las aplicaciones en tiempo real). (Pressman, 2010)

La arquitectura utilizada para la implementación es la propuesta por el framework Angular 6. La idea es utilizar la estructura MVC para presentar una organización en el código, donde el manejo de los datos (Modelo) estará separado de la lógica (Controlador) de la aplicación, y a su vez la información presentada al usuario (Vistas) se encontrará totalmente independiente. Es un proceso bastante sencillo donde el usuario interactúa con las vistas de la aplicación, éstas se comunican con los controladores notificando las acciones del usuario, los controladores realizan peticiones

a los modelos y estos gestionan la solicitud según la información brindada. Esta estructura provee una organización esencial a la hora de desarrollar aplicaciones de gran escala (Caule, 2014).

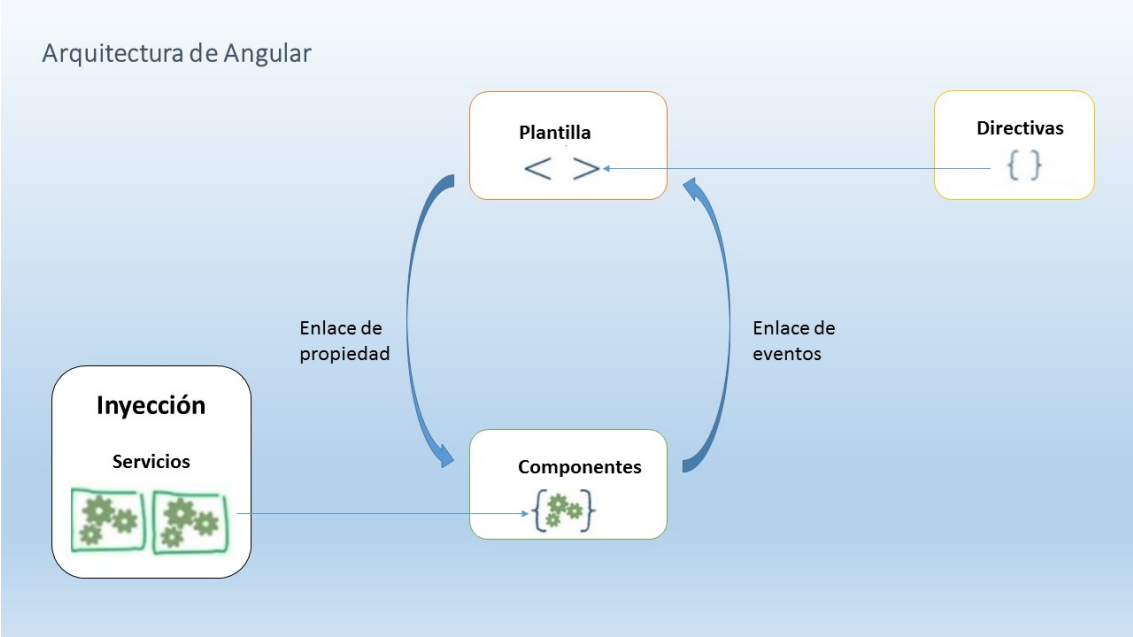


Imagen 5: Arquitectura de Angular 6, tomada de internet

2.13. Patrones de diseño

GRASP

Describen los principios fundamentales del diseño de objetos y la asignación de responsabilidades, expresados como patrones (Larman, 2003).

Experto:

La responsabilidad de realizar una labor específica es otorgada a la clase que contiene la información necesaria para realizar dicha labor (Larman, 2003).En la propuesta de solución un objeto de tipo Producto es el responsable de validar sus metadatos y exponer dicha evaluación a los demás objetos, pues la clase Producto es el experto en esa información ver el código:

```
export interface Productos {
    idprod: number;
    unit_name: string;
    unit_price: number;
    unit_cat: string;
    unit_shop: string;
```

```
    unit_cant: number;
    unit_cad: string;
    unit_dir: string;
    IdProvincia: number;
    unit_mun: string;
 }
```

Bajo Acoplamiento:

El diseño se debe realizar de forma tal que la dependencia entre clases sea baja. De esta forma se tiene clases menos dependientes, logrando una reducción del impacto de los cambios y garantizando un mayor grado de reutilización (Larman, 2003). Este patrón se pone en práctica al asociar a cada controlador un solo servicio, disminuyendo de esta forma el acoplamiento, ver código a continuación:

```
todoOk: Boolean=false;

addProd(unit_name, unit_price,unit_cat,unit_shop,unit_cant,
unit_cad,unit_dir,idProvincia,unit_mun) {

const obj = { unit_name: unit_name, unit_price:
unit_price,unit_cat:unit_cat,unit_shop: unit_shop, unit_cant: unit_cant,
unit_cad: unit_cad,unit_dir: unit_dir, idProvincia: idProvincia, unit_mun:
unit_mun

        };

      console.log (obj);

      this.http.post (this.url + '/api/prod/', obj).subscribe (res => {

        If (! this.todoOk) {

      window.alert('Su producto se  insertó correctamente.'+ unit_name);

      console.log('HECHO')

         }})}
```

Alta Cohesión:

Los elementos deben tener responsabilidades altamente relacionadas y efectuar la menor cantidad de operaciones posibles. Al existir una alta cohesión se hace más fácil el mantenimiento, entendimiento y la reutilización. (Larman, 2003). La utilización de este patrón se evidencia en los controladores, estos mantienen su código lo más simple posible manejando solo los eventos del sistema y delegando las demás tareas, como la implementación de la lógica de negocio a los servicios, ver código a continuación:

```
router.get('/prod/:id', function(req, res, next) {
    const results = [];
    const id=req.params.id;
    // Get a Postgres client from the connection pool
    pg.connect(connectionString, function(err, client, done) {
        // Handle connection errors
        if (err) {
            done();
            console.log(err);
            return res.status(500).json({
                success: false,
                data: err
            });
        }
        // SQL Query > Select Data
        const query = client.query('SELECT * FROM productos join "Provincia"
using("idProvincia") WHERE "idprod"=($1)  ', [id]);
        // Stream results back one row at a time
        query.on('row', function(row) {
            results.push(row);
        });
        // After all data is returned, close connection and return results
        query.on('end', function() {
            done();
            return res.json(results);
        });
    });
});
```

Conclusiones del capítulo

Una vez realizada la propuesta de solución de la investigación se llegó a las siguientes conclusiones:

- La definición del modelo conceptual con su respectivo diccionario de datos, facilitó el entendimiento del negocio a informatizar.
- La definición de los requisitos funcionales permitió identificar las funcionalidades a desarrollar.
- La descripción de casos de uso del sistema proporcionó una visión inicial de cómo pudiera quedar la solución en términos de interfaz.

- La arquitectura de software empleada y los patrones de diseño sentaron las bases para la construcción de la solución.

Capítulo III Implementación y validación de la propuesta de solución

3. Introducción

El presente capítulo tiene como objetivo la implementación y prueba de la propuesta, donde primeramente se muestra el diagrama de componentes, los estándares de codificación y un ejemplo de implantación de un microservicio, para después analizar la disciplina Pruebas Internas y Pruebas de Aceptación que propone la metodología AUP-UCI para el ciclo de vida del proyecto. Además de diseñar los casos de pruebas para facilitar la disciplina Pruebas internas.

3.1 Diagrama de componente

El diagrama de componentes describe la descomposición física del sistema en componentes, a efectos de construcción y funcionamiento. La descomposición del diagrama de componentes se realiza en términos de componentes y de relaciones entre ellos. Los componentes identifican objetos físicos que hay en tiempos de ejecución, de compilación o de desarrollo y tienen identidad propia con una interfaz bien definida (Cillero, 2019).

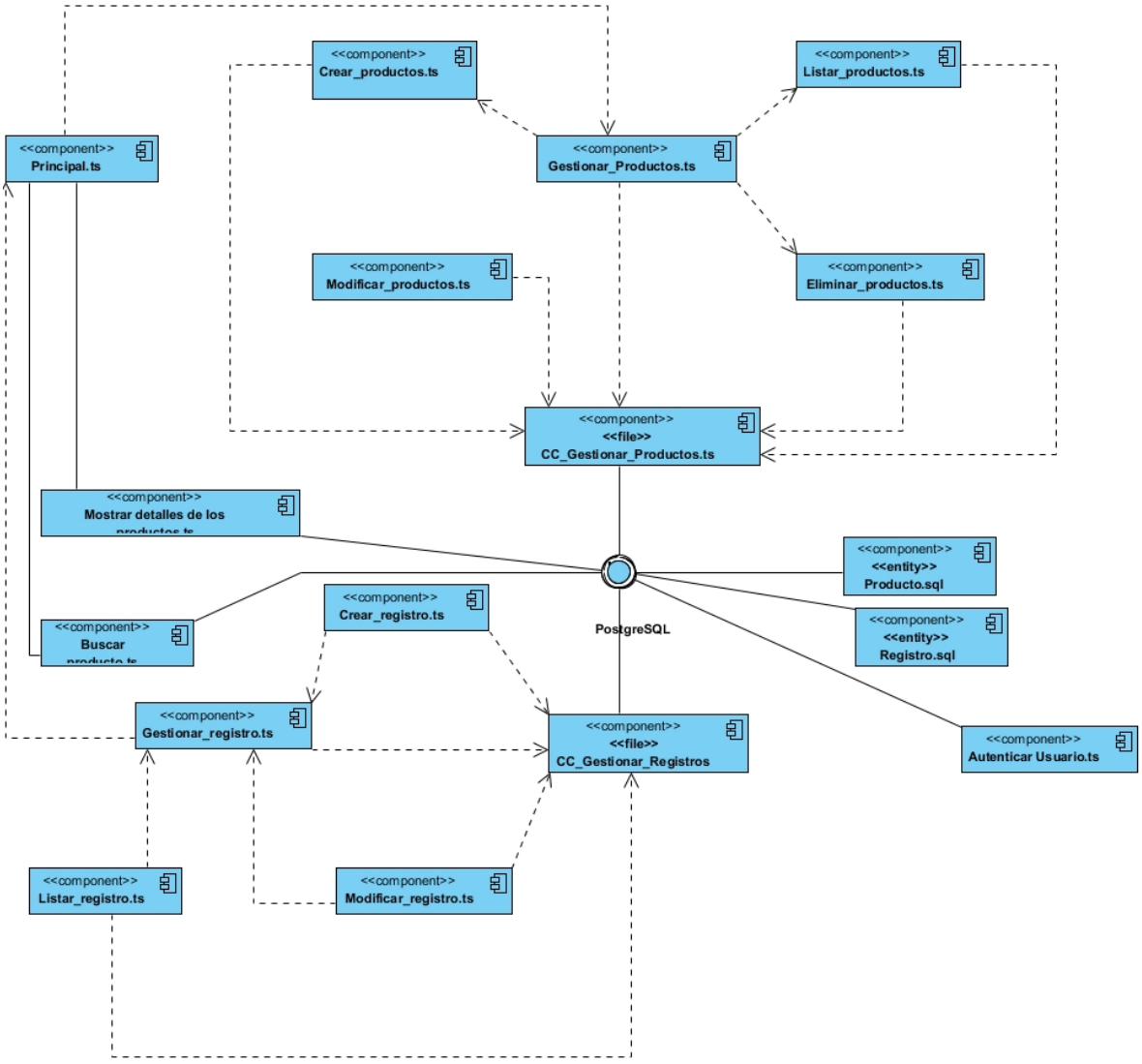


Diagrama 3: Diagrama de componente. Elaboración propia.

3.2 Estándares de codificación

- Los nombres que se usen deben ser significativos.
- Los nombres deben estar en minúsculas, excepto la primera letra de cada palabra a partir de la segunda.
- Los corchetes de un bloque if o for deben ir en la misma línea de la cláusula.
- Utilizar comillas si el valor contiene espacios.
- Utilizar siempre el **alt** atributo con imágenes.
- Las apis definida deben ser bien definidas y la dirección separada por **'/'**.

3.3 Implementación de microservicios

A continuación, se muestra la implementación del microservicio de obtención de todos los productos en la base de datos. Este fue implementado en el Visual Studio Code con Node, Express, Angular y PostgreSQL. Se procederá por pasos lógicos:

1. Se comienza por instalar el generador Express.
2. Crear un nuevo proyecto e instalar las dependencias.
3. Con su servidor PostgreSQL en funcionamiento en el puerto 5432, hacer una conexión de base de datos es fácil con la biblioteca pg.
4. Configurar un script de creación de tabla simple.
5. Agregar cada dirección.
6. Hacer las configuraciones de cada una de las funcionalidades.

Después de realizado estos pasos queda creado un microservicio como el que se muestra a continuación:

```
router.get('/prod', function(req, res, next) {
    const results = [];
    // Get a Postgres client from the connection pool
    pg.connect(connectionString, function(err, client, done) {
        // Handle connection errors
        if (err) {
            done();
            console.log(err);
            return res.status(500).json({
                success: false,
                data: err
            });
        }
        // SQL Query > Select Data
        const query = client.query('SELECT idprod, unit_name, unit_price, unit_cat, unit_shop, unit_cant, unit_dir,unit_cad, unit_mun, provincia
        // Stream results back one row at a time
        query.on('row', function(row) {
            results.push(row);
        });
        // After all data is returned, close connection and return results
        query.on('end', function() {
            done();
            return res.json(results);
        });
    });
});
```

Imagen 6: Microservicio Listar Producto

3.4 Pruebas de software

Las pruebas constituyen una actividad en la cual un sistema o componente es ejecutado bajo condiciones específicas, se observan o almacenan los resultados y se realiza una evaluación de algún aspecto del sistema o componente (Pressman, 2010).

Las pruebas del software son un elemento crítico para la garantía de calidad del software y representa una revisión final de las especificaciones, del diseño y de la codificación (Pressman, 2010).

El proceso de pruebas puede tener los siguientes objetivos (Board, 2010):

- Identificar defectos
- Aumentar la confianza en el nivel de calidad
- Facilitar información para la toma de decisiones
- Evitar la aparición de defectos

La metodología de software empleada en esta investigación define tres disciplinas de pruebas: Pruebas internas, Pruebas de liberación y Pruebas de aceptación. Para dicha investigación se desarrollan las disciplinas: Pruebas internas y Pruebas de aceptación.

3.5 Pruebas Internas

En esta disciplina se verifica el resultado de la implementación probando cada construcción, incluyendo tanto las construcciones internas como intermedias, así como las versiones finales a ser liberadas. Se deben desarrollar artefactos de prueba como: diseños de casos de prueba, listas de chequeo y de ser posible componente de prueba ejecutable para automatizar las pruebas (Pressman, 2010).

En este contexto se realizan pruebas al módulo mediante el empleo del método de prueba caja blanca haciendo uso de la técnica asociada a este método, prueba del camino básico y del método de prueba caja negra utilizando la técnica partición equivalente, asociada al mismo.

3.5.1 Método de Prueba Caja Blanca

Las pruebas de caja blanca, denominadas a veces pruebas de caja de cristal es un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para obtener los casos de prueba (Pressman, 2010). Con la aplicación de este método se garantiza que:

- Se verifique por lo menos una vez todos los caminos independientes de cada módulo.
- Se ejerciten todas las decisiones lógicas en sus vertientes verdaderas y falsas.

- Se ejecuten todos los bucles en sus límites y con sus límites operacionales.
- Se entrenen las estructuras internas de datos para asegurar su validez.

Técnica de prueba para el método de caja blanca: Prueba del Camino Básico.

La prueba del camino básico es una técnica de prueba de la Caja Blanca propuesta por Tom McCabe. Esta técnica permite obtener una medida de la complejidad lógica de un diseño y usar esta medida como guía para la definición de un conjunto básico (Pressman, 2010).

La idea es derivar casos de prueba a partir de un conjunto dado de caminos independientes por los cuales puede circular el flujo de control. Para obtener dicho conjunto de caminos independientes se construye el Grafo de Flujo asociado y se calcula su complejidad ciclomática. Los pasos que se siguen para aplicar esta técnica son:

1. A partir del diseño o del código fuente, obtiene el grafo de flujo asociado.

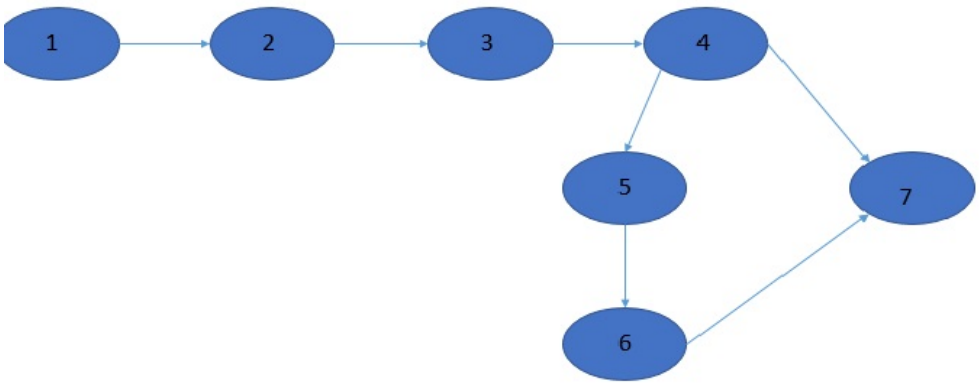


Imagen 7: Grafo asociado al código de Caja Blanca. Elaboración propia.

2. Se calcula la complejidad ciclomática del grafo obtenido.

   V (G)= # de regiones =2

   V (G)=A-N+2=7-7+2=2

   V (G)= P+1 =1+1

3. Se determina un conjunto básico de caminos independientes.

   Primer Camino: 1-2-3-4-5-6-7

Segundo Camino: 1-2-3-4-7

4. Se preparan los casos de prueba que obliguen a la ejecución de cada camino del conjunto básico.

Tabla 5: Diseño de caso de prueba para el camino 1

Descripción	Permite guardar los datos del producto que se inserta en la base de datos del módulo.
Condición de ejecución	La variable **todoOk** tiene que ser **true,** que denota que todo está bien.
Entrada	Datos del producto
Resultado	Devuelve un mensaje que expresa que su producto se insertó correctamente.

3.5.2 Método de Prueba Caja Negra

Las pruebas de caja negra también denominadas pruebas de comportamiento, se centran en los requisitos funcionales del software. O sea, permite obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa (Pressman, 2010).

Las pruebas de caja negra se llevan a cabo sobre la interfaz del software, obviando el comportamiento interno y la estructura del programa. Los casos de prueba de la caja negra pretenden demostrar que:

- Las funciones del software son operativas.
- La entrada se acepta de forma correcta.
- Se produce una salida correcta.
- La integridad de la información externa se mantiene.

Técnica de prueba para el método de caja negra: Partición Equivalente.

Esta técnica intenta dividir el dominio de entrada de un programa en un número finito de clases de equivalencia y definir el menor número de casos de prueba que descubran clases de errores (Sommerville, 2016).

El diseño de casos de prueba según esta técnica consta de dos pasos:

1. Identificar las clases de equivalencia.

2. Identificar los casos de prueba.

- Una prueba realizada con un valor representativo de cada clase es equivalente a una prueba realzada con cualquier otro valor de dicha clase.
- Si el caso de prueba correspondiente a una clase de equivalencia detecta un error, el resto de los casos de prueba de dicha clase de equivalencia deben detectar el mismo error.
- Un caso de prueba es un conjunto de entradas de pruebas, condiciones de ejecución, resultados esperados desarrollados para cumplir un objetivo en particular o una función esperada. La entidad más simple que siempre es ejecutada como una unidad, desde el comienzo hasta el final (Jacobson, 1998).

A continuación, se muestra el Caso de Prueba Buscar Producto y en el Anexo 4 Casos de prueba, se describen el resto de los casos de pruebas por escenarios.

Tabla 5: Caso de Prueba: Buscar Producto

Escenario	Descripción	Nombre del producto	Respuesta del módulo	Flujo Central
EC 1.1 Buscar Producto	El escenario de prueba permite al cliente buscar un producto	V Papa V papa V PAPA V ppa	El módulo verifica los datos y devuelve la lista de productos que coinciden.	1- Presionar el botón “Listado de productos” 2- Insertar dato en el formulario “Buscar producto” 3- Presionar “Buscar”
EC 1.2 Datos incorrectos	El escenario de prueba permite al cliente validar	I Zapato2%1	El módulo muestra un mensaje: “El valor entrado	<Lo mismo>

	campos incorrectos.	I Zapato. I 2zapato	contiene caracteres no validos”.	
EC 1.3 Campos obligatorios vacíos	El escenario de prueba permite al cliente validar campos vacíos.	I	El módulo muestra un mensaje: “El campo está vacío”.	<Lo mismo>
EC 1.4 No existe el producto	El escenario de prueba permite al usuario validar campos vacíos	V papa	El módulo muestra un mensaje: “El producto no está base de datos”.	<Lo mismo>

3.5.3 Pruebas Servicios Web Rest Crud

Estas pruebas son realizadas a los servicios web REST generados a partir de las entidades de la aplicación. Estos servicios son generados automáticamente por lo que se mostrarán las pruebas realizadas a un servicio web REST donde se solicite crear, editar, eliminar y leer datos de una entidad (Create, Read, Update, Delete) (Muñoz, 2013).

Ayuda a satisfacer los requisitos de la integración que son críticos para construir sistemas en los que los datos se puedan combinar fácilmente, y para extender o construir un conjunto de servicios de RESTful básicos y crear algo mucho mayor. Al realizar prueba a los servicios del módulo se obtiene los siguientes resultados mediante imágenes, en la imagen 6 se puede observar la petición y en la imagen 7 se observa la respuesta. Las demás pruebas se pueden observar en el Anexo 5 Pruebas de servicio:

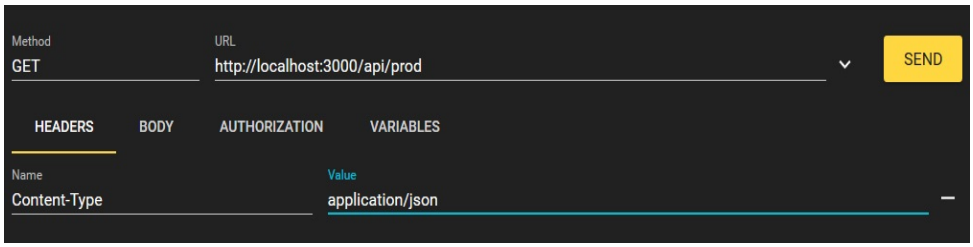


Imagen 8: Petición Get al servidor

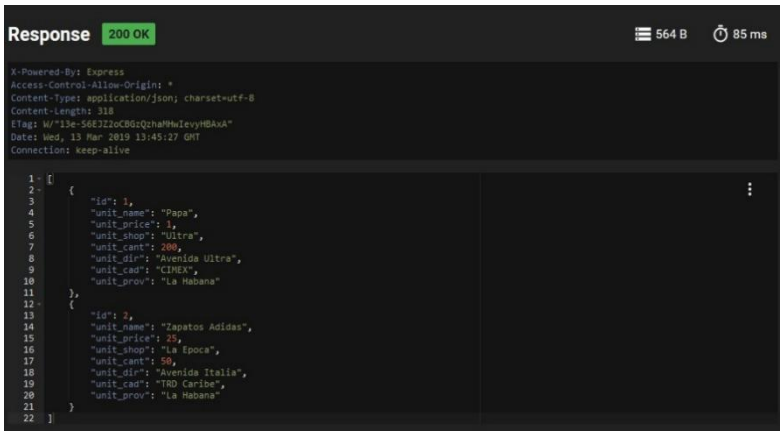


Imagen 9: Respuesta del servidor

Resultados de las pruebas internas

Con la terminación de las pruebas internas realizadas, se comprobó que el módulo cumple con los requisitos funcionales definidos para la investigación. En la tabla que se muestra a continuación, se observan la relación de no conformidades detectadas, resueltas y pendientes por iteración.

Tabla 6: No conformidades

Iteración	NC detectadas	Asociadas a	NC resueltas	NC pendientes
I	6	Funcionalidad Interfaz Implementación	4	2
II	2+2	Funcionalidad Interfaz	2	2
III	1+2	Funcionalidad	3	0

Como se aprecia en la tabla anterior se realizaron 3 iteraciones de pruebas ejecutadas por el desarrollador. A lo largo de la primera iteración se detectaron 6 no conformidades: 3 asociadas funcionalidad e implementación (opciones que no funcionan) y 3 de interfaz. De estas no conformidades fueron resueltas 4 en dicha iteración, quedando 2 pendientes para la segunda iteración. En la segunda iteración fueron detectadas 2 no conformidades, las cuales, junto a las 2 pendientes de la iteración anterior, sumaron un total de 4 no conformidades. Dichas no conformidades estaban asociadas a errores de interfaz y de funcionalidad, siendo satisfactoriamente solucionadas 2, quedando pendiente 2 no conformidades para una tercera iteración. En una tercera iteración se detectó 1 no

conformidades, las cuales junto a las 2 pendientes de la iteración anterior sumaron 3 no conformidades a resolver, de estas se resolvieron las 3 y culmina la solución sin errores.

3.6 Pruebas de aceptación:

Es la prueba final antes del despliegue del sistema. Su objetivo es verificar que el software está listo y que puede ser usado por usuarios finales para ejecutar aquellas funciones y tareas para las cuales el software fue construido (Sanchez, 2015).

Como resultado de dicha prueba, se obtuvo un acta de aceptación por el cliente (XETID) donde quedó plasmada su conformidad con el módulo, ver Anexo 6 Acta de aceptación del módulo.

Conclusiones del capítulo

Luego de desarrollar el capítulo se ha podido arribar a las siguientes conclusiones parciales:

- La definición del diagrama de componente y de los estándares de codificación contribuyeron en la construcción de la propuesta de solución.
- Se definieron los métodos y técnicas a utilizar para el desarrollo de los casos de pruebas que se utilizan para probar la solución.
- Se pudo validar el correcto funcionamiento de la solución a través de las pruebas internas y las pruebas de aceptación realizadas.

Conclusiones Generales

- Durante el desarrollo de la investigación se evidenció la necesidad de desarrollar microservicios para la obtención de información de los productos que oferta la red de ventas minorista en Cuba basado en la arquitectura de microservicios.
- El estudio de los referentes teóricos permitió identificar las características principales de la solución, propiciando un mejor entendimiento del diseño de la propuesta de solución.
- La implementación de los microservicios contribuye a la obtención de información de los productos que oferta la red de ventas minorista en el país.
- Las pruebas aplicadas a la propuesta de solución propiciaron una correcta validación de las funcionalidades y el diseño de la propuesta de solución.

Referencias Bibliográficas

2019. 5tay42. [En línea] online. http://5tay42.xetid.cu.

ACM. 2014. Computing Carrers and Degrees. [En línea] 2014.

Amazon. 2019. Amazon Mobile. *Amazon Mobile.* [En línea] 2019. www.amazon.com.

2019. Amazon.com. [En línea] online. www.amazon.com.

Basalo, Antonio. 2013. *Introducción a Angular.* 2013.

Board, ISTQB International Software Testing Qualifications. 2010. 2010.

Bortenschlager, Manfred. 2016. *Logrando la Agilidad de la empresa con Microservices y Gestión de API.* 2016.

Caule, Cecilio Álvarez. 2014. *Arquitecturas web con angular.js.* 2014.

Cillero, Manuel. 2019. Diagrama de Componentes. *manuel.cillero.es.* [En línea] 2019. [Citado el: 14 de mayo de 2019.] https://manuel.cillero.es/doc/metrica-3/tecnicas/diagrama-de-componentes/.

Cimex, Grupo Empresaril. 2019. Donde Hay. [En línea] 2019. www.DondeHay.cimex.cu.

Code, Visual Studio. 2019. Visual Studio Code. [En línea] 2019. https://code.visualstudio.com.

Comprando. 2019. Comprando. *Comprando.* [En línea] 2019. www.comprando.com.ar.

Comunicación, Instituto Nacional de Tecnologías de la. 2009. Ingeniería del software: metodologías y ciclos de vida. Madrid : s.n., 2009.

Cuba,más digital y conectada. **Guevara, Yurisander. 2018.** La Habana : Juventud Rebelde, 2018.

Diagrama de Despliegue. **Huallpara, Hugo Marca.**

Ebay. 2019. Ebay. *Ebay.* [En línea] 2019. www.ebay.com.

2019. Ebay.com. [En línea] online. www.ebay.com.

Gubern, Roman. 2010. *El simio informatizado.* Madrid : Fundesco, 2010.

Guerra, Martin. 2014. *La dirección estratégica de la empresa. Teoría y aplicaciones.* Madrid : s.n., 2014.

Gutierrez, Manuel J. 2019. Tiendeo. *Tiendeo.* [En línea] 2019. www.tiendeo.com.

Hernandez, Julian. 2009. *Curso de administrativo comercial: La gestion administrativa del departamento comercial.* 2009.

2016. Historia del comercio. [En línea] online. http://economiaes.com/comercio/historia-del-comercio.html.

1991. *IEEE.* 1991.

2018. Importancia del comercio. [En línea] online. [Citado el: 14 de mayo de 2019.] https://www.importancia.org/comercio.php.

Institute, Project Management. 2013. *GUÍA DE LOS FUNDAMENTOS PARA LA DIRECCIÓN DE PROYECTOS .* s.l. : 5ta Ed, 2013.

Jacobson, Ivar. 1998. *Applying UML in The Unified Process" Presentación. Rational Software.* 1998.

Larman, Henri. 2003. *Patrones de diseño Grasp.* 2003.

Lopez, Daniel. 2012. *Arquitectura de Software basada en Microservicios para desarrollo de aplicaciones web.* 2012.

Los procesos de gestión. **Huergo, Jorge. 2014.** 2014.

Maya, Edgar. 2017. *Arquitectura de Software basada en Microservicios para desarrollo de aplicaiones web.* 2017.

Microservices a definition of this new architectural term. **Fowler, Martin. 2014.** 2014.

Muñoz, Julian Camilo Ortega. 2013. *Pruebas funcionales y aceptacion.* 2013.

NodeJs. 2019. NodeJs. [En línea] 2019. https://nodejs.org.

Páez, Nicolás. 2014. Construcción de software: una mirada ágil. 2014.

Perez, Javier Eguilez. 2014. *Introduccion a css.* 2014.

Perez, Javier Eguilez. 2014. *Introducción a xhtml.* 2014.

PgAdmin. 2019. PgAdmin. *PgAdmin.* [En línea] 2019. www.pgadmin.com.

PMI. 2013. *GUÍA DE LOS FUNDAMENTOS PARA LA DIRECCION DE PROYECTOS.* s.l. : 5ta Edicion, 2013.

PostgesSQL. 2019. postgresql.org. [En línea] 2019. www.postgresql.org.es.

Pressman, Roger S. 2010. *Ingenieria de software,.Un enfoque práctico.* 2010.

2013. *Pruebas funcionales usando técnicas de caja negra.* 2013. Parte 1.

Rodríguez, Javier JesúsGutiérrez. 2013. *Diagramas UML de casos de uso y de requisitos.* 2013.

Ruesta, Carlota Bustelo. 2010. *Gestión de recursos informativos y documentales desde una perspectiva.* 2010. Vol 2.

Salazar, Pablo Hernán Vera. *Influencia de las TIC en las organizaciones: Cambios y aparición de nuevas formas organizativa.*

Martinez Pedro. 2016. *Influencia de las TIC en las organizaciones: Cambios y aparición de nuevas formas organizativa.* 2016.

Salem, Henrik. 2010. *El nuevo contexto de la distribución comercial.* Madrid : Ediciones Díaz de Santos, S.A., 2010. ISBN 84-7978-124-6..

Sampieri, Dr. Roberto Hernández. 2010. *Metodología de la investigación.* 2010. 5ta Edición.

Sánchez, Jorge. 2013. Introducción al JavaScript. 2013.

Sánchez, Tamara Rodríguez. 2015. *Metodología de desarrollo para la Actividad productiva de la UCI.* La Habana : s.n., 2015.

Santiago, Joaquin Quintas. 2018. *Ayuda de la plataforma Dprisa.* La Habana : s.n., 2018.

Sommerville, Ian. 2012. *Ingenieria de software_Parte III: Diseño.* 2012. 7edición.

Somerville, Ian. 2016. *Software Engineering.* Essex : 10th, 2016.

Sparks, Geoffrey. *Introducción al modelado de sistemas de software usando el Lenguaje Unificado de Modelado (UML) El modelo de componente .*

Sun,2017. [En línea] (http://www.sun.com/suntrademarks/).

2019. Superfacil. [En línea] online. www.superfacil.cu.

Tproduccionmultimedia's. 2019. Tproduccionmultimedia's Blog. [En línea] 2019. [Citado el: 14 de Mayo de 2019.] https://tproduccionmultimedia.wordpress.com/diccionario-de-datos/.

UCI. 2018. Universidad de las ciencias informáticas. [En línea] 2018. [Citado el: 2 de noviembre de 2018.] https://www.uci.cu/.

Viollaz, Mariana. 2014. Inicios del comercio. *Liberalización del comercio e informalidad laboral: ajustes de corto y largo plazo.* 2014.

W3, 2017. [En línea] www.w3.org.

XETID. 2019. Tienda Virtual. [En línea] Xetid, 2019. 5tay42.xetid.cu.

XETID. 2018. XETID. [En línea] 2018. www.xetid.cu.

Anexo

Anexo 1: Diccionario de datos:

Tabla 7: Diccionario de datos del registro

Descripción	Esta entidad representa las propiedades de un registro en el módulo.					
Atributos						
Nombre	**Descripción**	**Tipo**	**¿Puede ser nulo?**	**¿Es único?**	**Restricciones**	
					Clases válidas	**Clases no válidas**
Identificador	Valor único que identifica a cada producto.	Número entero	No	Si	Toma valores que se obtengan de la combinación de los dígitos del [0-9].	Puede tomar como valor máximo 2*10^9.
Entidad	Entidad que gestiona el producto	Cadena de caracteres.	No	Si	Puede tomar cualquier combinación de letras [a-z], [A-Z], espacio. Admite hasta 50 caracteres.	
Cadena	Cadena a la que pertenece la entidad	Cadena de caracteres	No	Si	Puede tomar cualquier combinación de letras [a-z], [A-Z], espacio. Admite hasta 50 caracteres.	
Usuario	Usuario que identifica la entidad dentro del módulo	Cadena de caracteres.	No	Si	Puede tomar cualquier combinación de letras [a-z], [A-Z], espacio. Puede contener algún número.	

Contraseña	Contraseña que identifica al usuario.	Cadena de caracteres.	No	No	Puede tomar cualquier combinación de letras [a-z], [A-Z], espacio y números.	

Anexo 2 Descripción de los casos de uso del sistema:

Tabla 6: Mostrar detalle del producto

Objetivo	Mostrar detalle del producto seleccionado.	
Actores	Proveedor, Cliente.	
Resumen	El caso de uso se inicia cuando un proveedor o un cliente necesitan ver la información detallada de un producto seleccionado.	
Complejidad	**Media**	
Prioridad	**Media**	
Precondiciones	Debe existir al menos un producto creado en el sistema.	
Poscondiciones	Permite ver la información detallada del producto seleccionado.	
Flujo de eventos		
Flujo básico Mostrar detalle del producto		
	Actor	**Sistema**
1	Presiona el ícono “Mostrar detalle” del producto escogido.	
2		El sistema muestra una interfaz con todos los detalles del producto.
3		Termina el caso de uso.
Relaciones	**CU incluidos**	N/A
	CU extendidos	N/A
Requisitos no funcionales	N/A	
Asuntos pendientes	N/A	

Tabla 7: Buscar Producto

Objetivo	Buscar un producto.
Actores	Proveedor, Cliente.
Resumen	El caso de uso se inicia cuando un usuario (proveedor o cliente) presiona el input de texto “Buscar”.
Complejidad	Media
Prioridad	Media
Precondiciones	Debe existir al menos un producto creado en el sistema.
Poscondiciones	Se buscan los productos que coincidan con la búsqueda.
Flujo de eventos	
Flujo básico Buscar producto	

	Actor	Sistema
1	Introduce un término de búsqueda.	
2		El sistema muestra todos los productos que coincidan con el término de búsqueda.
3		Termina el caso de uso.
Flujos alternos		
2.a Evento Entrada incorrecta de los datos		
	Actor	**Sistema**
1		El sistema muestra el mensaje “No se encontraron resultados.”
2		Termina el Caso de Uso.

Relaciones	**CU incluidos**	N/A
	CU extendidos	N/A
Requisitos no funcionales	N/A	
Asuntos pendientes	N/A	

Tabla 8: Gestionar registro

Objetivo	Crear, modificar, listar o eliminar un registro.
Actores	Proveedor
Resumen	El caso de uso se inicia cuando el proveedor necesita crear, modificar, listar o eliminar un registro en el sistema.
Complejidad	Alta
Prioridad	Critica
Precondiciones	El usuario proveedor debe estar autenticado en el sistema. Para las secciones Modificar y Eliminar registro debe estar registrado en el sistema.
Poscondiciones	Se almacenan el registro creados o modificados, permite ver los detalles de su registro y si se elimina desaparece de manera permanente del sistema.
Flujo de eventos	
Flujo básico Gestionar registro	

	Actor	Sistema
1	Crea, modifica, lista o eliminar un registro.	
2		Permite realizar varias acciones con el registro: - Crear un registro. Ver Sección 1: Crear registro. - Modificar un registro.

		Ver Sección 2: Modificar registro. - Listar un registro. Ver Sección 3: Listar registro.
3		Termina el caso de uso.
Sección 1: "Crear registro"		
Flujo básico Crear registro		
	Actor	**Sistema**
1	Selecciona la opción Registrarse.	
8.		El sistema muestra un formulario con los siguientes datos: Entidad Cadena Usuario Contraseña
9.	Inserta los valores	
10.	Presiona el botón Guardar	
11.		El sistema valida la información insertada
12.		El sistema crea el registro
13.		El sistema muestra el mensaje siguiente: "El registro se creó correctamente"
14.		Termina el Caso de Uso.
Flujos alternos		
4ª Presionar el botón Cancelar		
	Actor	**Sistema**
1	Presionar botón Cancelar	
2		El sistema cancela la acción y muestra la interfaz anterior.
3		Termina el Caso de Uso
5ª Existen campos obligatorios vacíos		
	Actor	**Sistema**
1		El sistema muestra encima del campo obligatorio, el mensaje "El campo no puede estar vacío"
2		Continuar en el paso 3 del flujo básico.
5b Existen campos incorrectos		
	Actor	**Sistema**
1		El sistema muestra el mensaje "El nombre es incorrecto"
2		Continuar en el paso 3 del flujo básico.

Sección 2: “Modificar registro”		
Flujo básico Modificar registro		
	Actor	**Sistema**
1	Presione el ícono “Modificar” el registro escogido.	
2		El sistema muestra un formulario con los siguientes datos: Entidad Cadena Usuario Contraseña
3	Inserta los valores	
4	Presiona el botón Aceptar	
5		El sistema valida la información insertada
6		El sistema modifica el registro
7		Termina el Caso de Uso.
Flujos alternos		
4ª Presionar el botón Cancelar		
	Actor	**Sistema**
1	Presionar botón Cancelar	
2		El sistema cancela la acción y muestra la interfaz anterior.
3		Termina el Caso de Uso
5ª Existen campos obligatorios vacíos		
	Actor	**Sistema**
1		El sistema muestra encima del campo obligatorio, el mensaje “El campo no puede estar vacío”
2		Continuar en el paso 3 del flujo básico.
5b Existen campos incorrectos		
	Actor	**Sistema**
1		El sistema muestra el mensaje “El nombre es incorrecto”
2		Continuar en el paso 3 del flujo básico.
Sección 3: “Listar registro”		
Flujo básico Listar registro		
	Actor	**Sistema**
1	Selecciona la opción Listar registro.	
2		Muestra la interfaz donde se listan los registros.
3		Terminar Caso de Uso.
Relaciones	**CU incluidos**	N/A

	CU extendidos	N/A
Requisitos no funcionales	N/A	
Asuntos pendientes	N/A	

Tabla9: Autenticar Usuario

Objetivo	Permitir que el usuario se autentique para tener acceso al sistema.	
Actores	Usuario	
Resumen	Es donde se verifica si se tiene acceso al sistema.	
Complejidad	Alta	
Prioridad	Crítico	
Precondiciones	El sistema debe verifica si el usuario tiene o no acceso al sistema	
Poscondiciones	Si existe este usuario en la base dato el sistema dejara que este acceda.	
Flujo de eventos		
Flujo básico <Autenticar usuario>		
	Actor	**Sistema**
1.	Se revisa si hay acceso al sistema	
2.	Acciones a ejecutar por pasos	Introducir Usuario Introducir Contraseña
1.		Termina el caso de uso
Flujos alternos		
Nº1 Evento <Los datos son incorrectos>		
	Actor	**Sistema**
1.	Verifica la correctitud de los datos	Lanza un mensaje que dice que verifique sus datos.

Anexo 3 Matrices de trazabilidad:

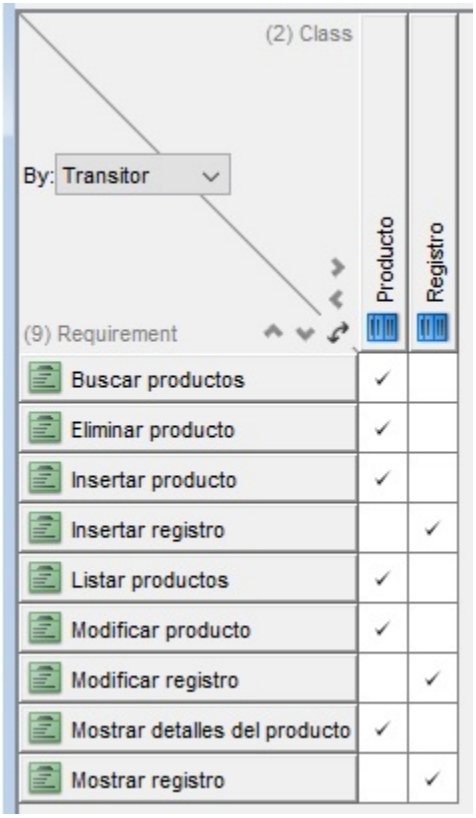


Imagen 10: Matriz de trazabilidad: Requisito-Modelo Conceptual

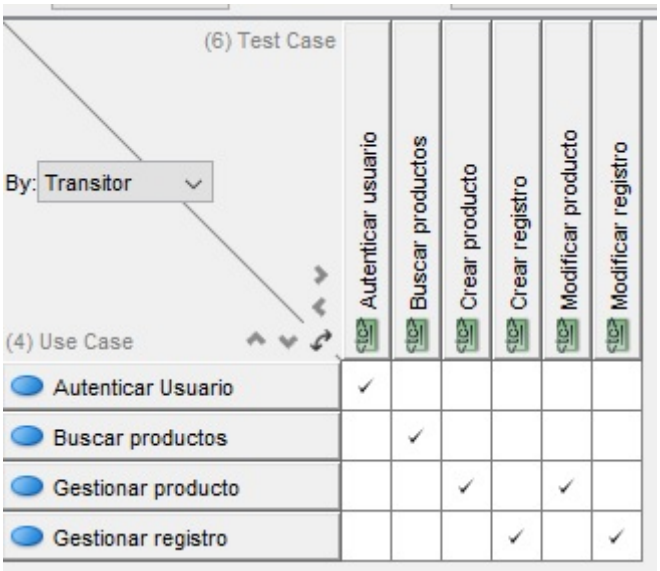
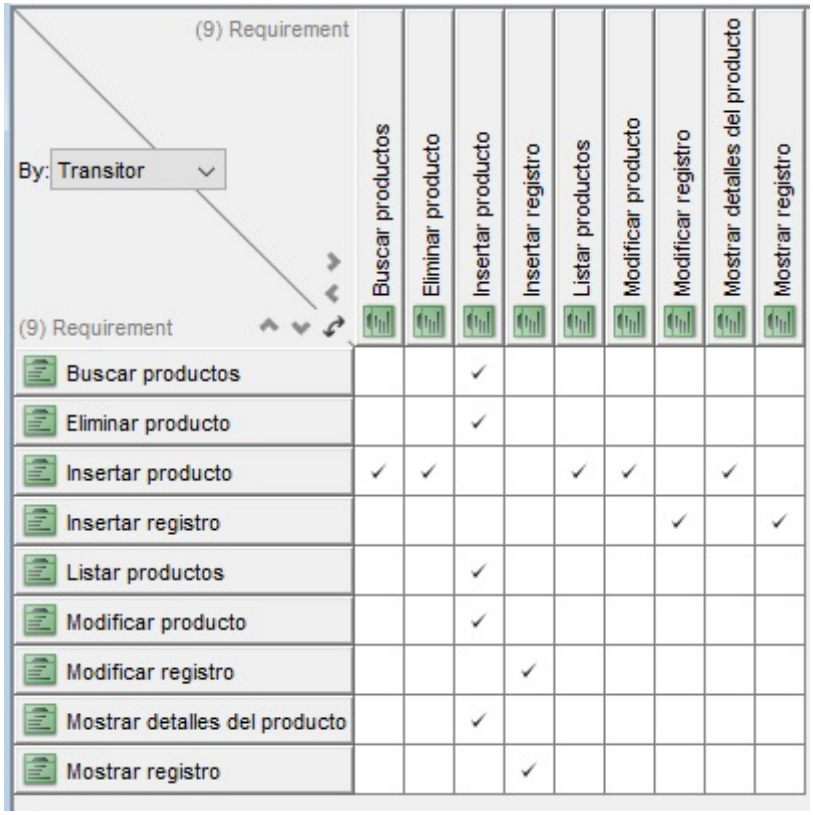


Imagen 11: Matriz de trazabilidad: Casos de usos-Caso de prueba



(9) Requirement / By: Transitor / (9) Requirement	Buscar productos	Eliminar producto	Insertar producto	Insertar registro	Listar productos	Modificar producto	Modificar registro	Mostrar detalles del producto	Mostrar registro
Buscar productos			✓						
Eliminar producto			✓						
Insertar producto	✓	✓			✓	✓		✓	
Insertar registro							✓		✓
Listar productos			✓						
Modificar producto			✓						
Modificar registro				✓					
Mostrar detalles del producto			✓						
Mostrar registro				✓					

Imagen 12: Matriz de trazabilidad: Requisito Funcional-Requisito Funcional

Anexo 4 Caso de prueba

Tabla 8: Caso de Prueba: Crear producto

Escenario	Descripción	Nombre del producto	Precio	Categoría	Tienda	Cantidad	Cadena	Dirección	Provincia	Municipio	Respuesta del módulo	Flujo Central
EC 1.1 Crear Producto	El escenario de prueba permite al cliente buscar un producto	V	V	V	V	V	V	V	V	V	El módulo verifica que los datos introducidos cumplan con los parámetros y si todo es satisfactorio inserta el producto en la bases de datos.	1. Presionar botón 'Nuevo Producto', 2. Insertar datos en el formulario, 3. Presionar botón 'Guardar'
		aceite	2	Alimentos	Ultra	200	TRD Caribe	Avenida Reina	La Habana	Centro Habana		
EC 1.2 Datos incorrectos	El escenario de prueba permite al usuario validar campos incorrectos.	I	V	V	V	V	V	V	V	V	El módulo muestra un mensaje: "El valor entrado contiene caracteres no válidos".	
		aceite2	2	Alimentos	Ultra	200	TRD Caribe	Avenida Reina	La Habana	Centro Habana		
		V	V	V	V	I	V	V	V	V	El módulo muestra un mensaje:	

EC 1.2 Datos incorrectos	El escenario de prueba permite al usuario validar campos incorrectos.	aceite	2	Alimentos	Ultra	paa	TRD Caribe	Avenida Reina	La Habana	Centro Habana	“El valor entrado contiene caracteres no validos”.	
EC 1.3 Campos obligatorios vacíos	El escenario de prueba permite al usuario validar campos vacíos.	I	V	V	V	V	V	V	V	V	El módulo muestra un mensaje: “El campo está vacío”.	
			2	Alimentos	Ultra		TRD Caribe	Avenida Reina	La Habana	Centro Habana		
EC 1.3 Campos obligatorios vacíos	El escenario de prueba permite al usuario validar campos vacíos.	I	V	V	I	V	V	V	V	V	El módulo muestra un mensaje: “El campo está vacío”.	
		aceite	2	Alimentos		200	TRD Caribe	Avenida Reina	La Habana	Centro Habana		

Tabla 9: Caso de Prueba: Modificar producto

Escenario	Descripción	Nombre del producto	Precio	Categoría	Tienda	Cantidad	Cadena	Dirección	Provincia	Municipio	Respuesta del módulo	Flujo Central
EC 1.1 Crear Producto	El escenario de prueba permite al cliente buscar un producto	V	V	V	V	V	V	V	V	V	El módulo verifica que los datos introducidos cumplan con los parámetros y si todo es satisfactorio modifica los datos del producto en la bases de datos.	1. Presionar botón 'Modificar Producto', 2. Insertar datos en el formulario, 3. Presionar botón 'Guardar'
		aceite	2	Alimentos	Ultra	200	TRD Caribe	Avenida Reina	La Habana	Centro Habana		
EC 1.2 Datos incorrectos	El escenario de prueba permite al usuario validar campos incorrectos.	I	V	V	V	V	V	V	V	V	El módulo muestra un mensaje: "El valor entrado contiene caracteres no válidos".	
		aceite2	2	Alimentos	Ultra	200	TRD Caribe	Avenida Reina	La Habana	Centro Habana		
EC 1.2 Datos incorrectos	El escenario de prueba permite al	V	V	V	V	I	V	V	V	V	El módulo muestra un mensaje:	

	usuario validar campos incorrectos.	aceite	2	Alimentos	Ultra	paa	TRD Caribe	Avenida Reina	La Habana	Centro Habana	“El valor entrado contiene caracteres no válidos”.	
EC 1.3 Campos obligatorios vacíos	El escenario de prueba permite al usuario validar campos vacíos.	I	V	V	V	V	V	V	V	V	El módulo muestra un mensaje: “El campo está vacío”.	
			2	Alimentos	Ultra		TRD Caribe	Avenida Reina	La Habana	Centro Habana		
EC 1.3 Campos obligatorios vacíos	El escenario de prueba permite al usuario validar campos vacíos.	I	V	V	I	V	V	V	V	V	El módulo muestra un mensaje: “El campo está vacío”.	
		aceite	2	Alimentos		200	TRD Caribe	Avenida Reina	La Habana	Centro Habana		

Tabla 10: Caso de prueba: Autenticar usuario

Escenario	Descripción	Usuario	Contraseña	Respuesta del módulo	Flujo Central
EC 1.Autenticar usuario	El escenario de prueba permite al usuario la entrada al módulo.	V mdcollazo	V …….	El módulo verifica los datos y autentica con los que coinciden.	1. Insertar los datos en el formulario de acceso, 2. Presionar botón 'Acceder'
EC 1.2 Datos incorrectos	El escenario de prueba permite al usuario validar campos incorrectos.	V mdcollazo	I ….	El módulo muestra un mensaje: "La contraseña no tiene la cantidad de caracteres requeridos".	
EC 1.2 Datos incorrectos	El escenario de prueba permite al usuario validar campos incorrectos.	V	I	El módulo muestra un mensaje: "El contraseña no es correcta".	

		mdcollazo	..……		
EC 1.2 Datos incorrectos	El escenario de prueba permite al usuario validar campos incorrectos.	I mdcollazo1	V …….	El módulo muestra un mensaje: “El usuario no es correcto”.	
EC 1.3 Campos obligatorios vacíos	El escenario de prueba permite al usuario validar campos vacíos.	V mdcollazo	I	El módulo muestra un mensaje: “El campo está vacío”.	
EC 1.3 Campos obligatorios vacíos	El escenario de prueba permite al usuario validar campos vacíos.	I	V ……..	El módulo muestra un mensaje: “El campo está vacío”.	

Tabla 11: Caso de prueba: Crear registro

Escenario	Descripción	Entidad	Cadena	Usuario	Contraseña	Respuesta del módulo	Flujo Central
EC 1.1 Crear Registro	El escenario de prueba permite al proveedor registrarse en el módulo.	V	V	V	V	El módulo verifica los datos y registra si todos son válidos.	1. Presionar el botón 'Registrarse', 2. Insertar los datos en el formulario, 3. Presionar el botón 'Registrar'
		Cuevita	Cuenta Propia	cuevita12	…….		
EC 1.2 Datos incorrectos	El escenario de prueba permite al usuario validar campos incorrectos.	V	I	V	V	El módulo muestra un mensaje: "La cadena no está establecida".	
		Cuevita	pepito	cuevita12	…….		
EC 1.3 Campos	El escenario de prueba permite	V	I	V	V	El módulo muestra un	

obligatorios vacíos	al cliente validar campos vacíos.	cuevita		Cuvita12	……	mensaje: “El campo está vacío”.	
EC 1.3 Campos obligatorios vacíos	El escenario de prueba permite al cliente validar campos vacíos.	I	V	V	V	El módulo muestra un mensaje: “El campo está vacío”.	
			Cuenta Propia	Cuevita12	.....		

Tabla 12: Caso de prueba: Modificar registro

Escenario	Descripción	Entidad	Cadena	Usuario	Contraseña	Respuesta del módulo	Flujo Central
EC 1.1 Modificar Registro	El escenario de prueba permite al proveedor modificar sus datos de registro en el módulo.	V	V	V	V	El módulo verifica los datos y actualizar los datos si todos son válidos.	Presionar el botón 'Modificar', Insertar los datos en el formulario, Presionar el botón 'Modificar'
		Cuevita	Cuenta Propia	cuevita12	…….		
EC 1.2 Datos incorrectos	El escenario de prueba permite al usuario validar campos incorrectos.	V	I	V	V	El módulo muestra un mensaje: "La cadena no está establecida".	
		Cuevita	pepito	cuevita12	…….		

EC 1.3 Campos obligatorios vacíos	El escenario de prueba permite al cliente validar campos vacíos.	V	I	V	V	El módulo muestra un mensaje: “El campo está vacío”.	
		cuevita		Cuvita12	……		
EC 1.3 Campos obligatorios vacíos	El escenario de prueba permite al cliente validar campos vacíos.	I	V	V	V	El módulo muestra un mensaje: “El campo está vacío”.	
			Cuenta Propia	Cuevita12	……		

Anexo 5 Pruebas a los servicios

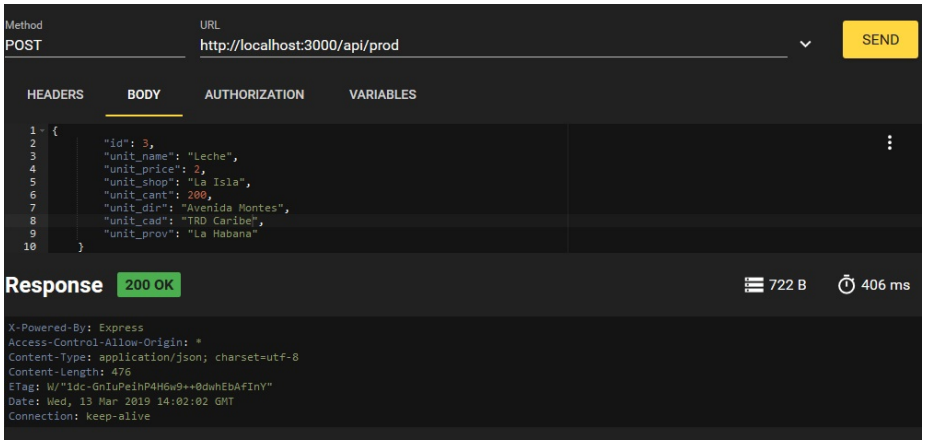


Imagen 13: Petición Post al servidor

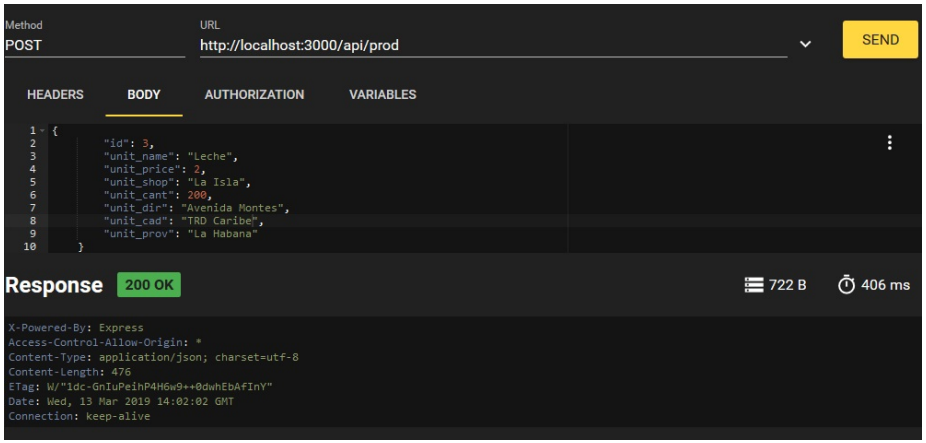


Imagen 14: Respuesta del servidor

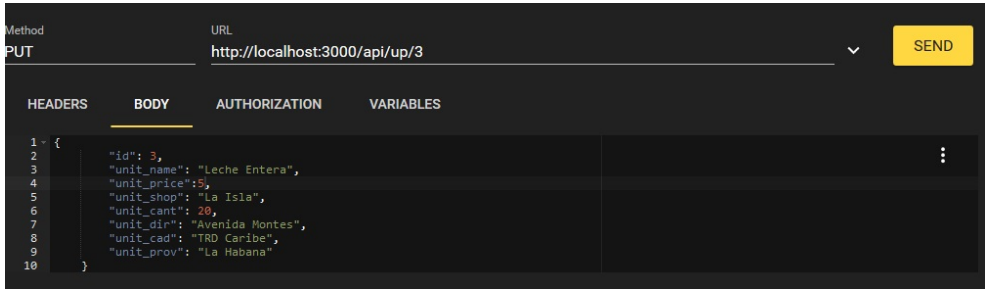


Imagen 15: Petición Put al servidor

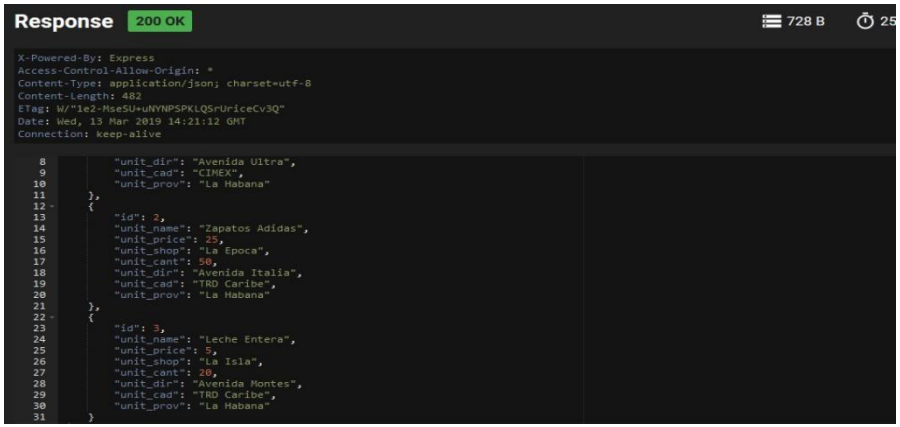
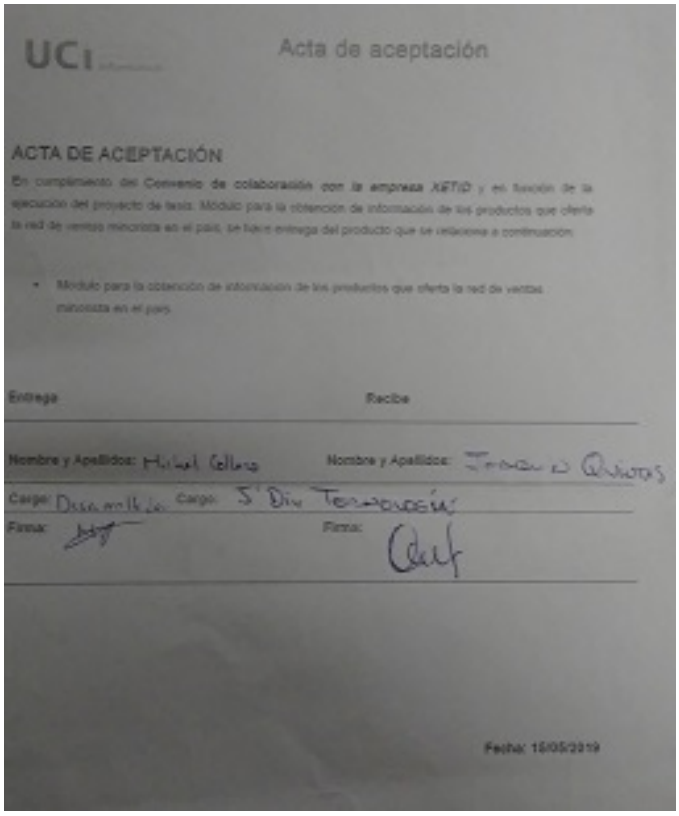


Imagen 16: Respuesta del servidor

Anexo 6 Acta de Aceptación



UCI

Acta de aceptación

ACTA DE ACEPTACIÓN

En cumplimiento del Convenio de colaboración con la empresa XETID y en función de la ejecución del proyecto de tesis. Módulo para la obtención de información de los productos que oferta la red de ventas minorista en el país, se hace entrega del producto que se relaciona a continuación:

- Módulo para la obtención de información de los productos que oferta la red de ventas minorista en el país.

Entrega	Recibe
Nombre y Apellidos: Michel Collera	Nombre y Apellidos: Ernesto Quintos
Cargo: Desarrollador	Cargo: Jefe Div Tecnologías
Firma:	Firma:

Fecha: 15/05/2019