



FACULTAD DE INGENIERÍAS Y CIENCIAS APLICADAS

DESARROLLO DE UN SISTEMA WEB PARA COMERCIALIZACIÓN DE LOS
PRODUCTOS DE LA GRANJA “NONO” DE LA UNIVERSIDAD DE LAS
AMÉRICAS

Trabajo de Titulación presentado en conformidad con los requisitos
establecidos para optar por el título de Ingeniero en Sistemas de Computación
e Informática

Profesor Guía

MSc. Pedro Manuel Nogales Cobas

Autor

Hernán Gustavo Verdezoto Feijóo

Año

2019

DECLARACIÓN DEL PROFESOR GUÍA

"Declaro haber dirigido el trabajo, Desarrollo de un sistema web para comercialización de los productos de la granja "Nono" de la Universidad de Las Américas, a través de reuniones periódicas con el estudiante Hernán Gustavo Verdezoto Feijóo, en el semestre 201920, orientando sus conocimientos y competencias para un eficiente desarrollo del tema escogido y dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación".

Pedro Manuel Nogales Cobas.

Master en Gestión de Proyectos Informáticos

CI: 1756760284

DECLARACIÓN DEL PROFESOR CORRECTOR

"Declaro haber revisado este trabajo, Desarrollo de un sistema web para comercialización de los productos de la granja "Nono" de la Universidad de Las Américas, de Hernán Gustavo Verdezoto Feijóo, en el semestre 201920 dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación".

Anita Elizabeth Yánez Torres

Magíster en Gerencia Empresarial

CI:1802462216

DECLARACIÓN DE AUTORÍA DEL ESTUDIANTE

“Declaro que este trabajo es original, de mi autoría, que se han citado las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen los derechos de autor vigentes.”

Hernán Gustavo Verdezoto Feijóo

CI: 1719838094

AGRADECIMIENTOS

A mi familia, amigos y
docentes por su apoyo y
guía

DEDICATORIA

A mis padres, hermanas, sobrinos, a Néstor, Mario, Santiago, Juan, Karla ya que, sin su apoyo constante y su cariño, esto no hubiera sido posible

RESUMEN

La Granja Nono elabora productos de manera tradicional y los comercializa dentro de la comunidad UDLA, estos productos tienen varios beneficios para la salud de las personas ya que estos son de sello verde. La falta de alcance y conocimiento de estos productos dentro de la comunidad UDLA permitió que se desarrolle el sistema descrito en este documento, partiendo desde un marco teórico, pasando por su desarrollo hasta llegar a la validación del sistema. Gracias a este software las personas podrán buscar, seleccionar y reservar los productos que ellos deseen, facilitando mucho la adquisición de estos productos y permitiendo que las personas puedan conocer mejor los productos que elabora la Granja UDLA. La solución fue desarrollada con el marco de trabajo Scrum, base de datos MySQL, el framework Laravel, PHP como lenguaje de programación y utilizando el patrón Modelo Vista Controlador.

ABSTRACT

The Nono Farm elaborates products in a traditional way and commercializes them within the UDLA community, these products have several benefits for the health of the people since these are of green seal. The lack of scope and knowledge of these products inside the UDLA community allowed the system described in this document to be developed, starting from a theoretical context, going through its development until it reaches the validation of the system. Thanks to this software, people can search, select and reserve the products they want, making it much easier to purchase these products and allowing people to better know the products produced by the UDLA Farm. The solution was developed with the Scrum framework, MySQL database, Laravel framework, PHP as programming language and using the Model View Controller pattern.

ÍNDICE

1. Capítulo 1. Introducción	1
1.2 Alcance	2
1.3 Justificación	3
1.4 Objetivo General	4
1.5 Objetivos específicos	4
2. Capítulo 2. Marco teórico	5
2.1 Introducción	5
2.2 Granja UDLA.....	5
2.3 Comercio electrónico	6
2.3.1 ¿Qué es comercio electrónico?	6
2.3.2 Ventajas del comercio electrónico	6
2.3.3 Herramientas de distribución y comercialización	7
2.4 Herramientas y aplicaciones similares	7
2.5 Marco de trabajo	9
2.5.1 Definición e historia	9
2.5.2 Roles, Artefactos y Actividades.....	10
2.5.3 Ciclo de vida de Scrum	12
2.5.4 Medición y estimación ágil	13
2.5.5 Herramientas para llevar Scrum	14
2.6 Tecnologías de desarrollo	14
2.6.1 Framework de desarrollo	14
2.6.2 Lenguaje de programación	16
2.6.3 Base de datos.....	17

2.7 Pruebas de verificación y validación	18
2.7.1 Apache Jmeter	19
3. Capítulo 3. Propuesta de Solución	20
3.1 Introducción	20
3.2 Product Backlog.....	20
3.3 Sprints.....	27
3.3.1 Flujo del primer Sprint.....	28
3.3.2 Flujo del segundo Sprint	29
3.4 Primer Sprint.....	30
3.4.1 Sprint Planning	30
3.4.2 Sprint Backlog	30
3.4.3 Daily Scrum	32
3.4.4 Sprint Review	33
3.4.5 Código del primer Sprint	33
3.4.6 Capturas de pantalla del incremento del primer Sprint.....	41
3.4.7 Sprint Retrospective	42
3.5 Segundo Sprint	42
3.5.1 Sprint Planning	42
3.5.2 Sprint Backlog	43
3.5.3 Daily Scrum	45
3.5.4 Sprint Review	45
3.5.5 Código segundo Sprint	45
3.5.6 Capturas de pantalla del incremento del segundo Sprint	48
3.5.7 Sprint Retrospective	49
3.6 Medición y Estimación.....	50

3.7 Gráfico Burn Up	51
3.8 Gráficos Burn Down	52
3.8.1 Gráfico Burn Down Primer Sprint	52
3.8.2 Gráfico Burn Down Segundo Sprint	53
4. Capítulo 4. Validación y Pruebas del Software	53
4.1 Introducción	53
4.2 Pruebas de aceptación del cliente	53
4.2.1 Demostración de las pruebas de aceptación del cliente	55
4.3 Pruebas unitarias	55
4.3.1 Demostración del cumplimiento de las pruebas unitarias	56
4.4 Pruebas de carga y estrés	56
5. Conclusiones y Recomendaciones	58
5.1 Conclusiones	58
5.2 Recomendaciones	59
Referencias	61
ANEXOS	67

ÍNDICE DE TABLAS

Tabla 1: Product backlog	21
Tabla 2: Sprint backlog del primer sprint	30
Tabla 3: Sprint backlog del segundo sprint.....	43
Tabla 4: Pruebas de aceptación del cliente	54
Tabla 5: Pruebas unitarias.	56

INDICE DE FIGURAS

Figura 1: ciclo de vida de scrum.....	12
Figura 2: flujo de trabajo realizado en el primer sprint	28
Figura 3: flujo de trabajo realizado en el segundo sprint	29
Figura 4: código fuente controlador global “categorías”	33
Figura 5: código fuente controlador con autorización “categorías” parte 1.....	34
Figura 6: código fuente controlador con autorización “categorías” parte 2.....	35
Figura 7: código fuente controlador con autorización “categorías” parte 3.....	36
Figura 8: código fuente controlador con autorización “categorías” parte 4.....	36
Figura 9: código fuente controlador global “productos”	37
Figura 10: código fuente controlador autenticado “productos” parte 1	38
Figura 11: código fuente controlador autenticado “productos” parte 2	39
Figura 12: código fuente controlador autenticado “productos” parte 3	40
Figura 13: código fuente controlador “carrito”	41
Figura 14: menú creado después del primer sprint	41
Figura 15: botón para agregar nuevo producto	42
Figura 16: forma de registro de producto completa	42
Figura 17: producto agregado	42
Figura 18: código fuente controlador “bienvenida”	45
Figura 19: código fuente controlador “ubicaciones”	46
Figura 20: código fuente controlador imágenes “productos” parte 1	46
Figura 21: código fuente controlador imágenes “productos” parte 2.....	47
Figura 22: código fuente controlador imágenes “productos” parte 3.....	47
Figura 23: menú de la aplicación luego del segundo sprint	48
Figura 24: listado de productos	48
Figura 25: agregar imágenes	48
Figura 26: imágenes agregadas.....	49
Figura 27: información del producto luego de imagen agregada.	49

Figura 28: gráfico burn up final.....	51
Figura 29: burn down primer sprint.....	52
Figura 30: burn down segundo sprint	53
Figura 31: vista de categorías	68
Figura 32: vista de productos por categoría	68
Figura 33: vista de categorías	69
Figura 34: resultado búsqueda.....	69
Figura 35: vista ubicaciones.....	70
Figura 36: vista mapa.....	70
Figura 37: forma de contacto	71
Figura 38: confirmación de envió mail	71
Figura 39: recepción de mail	71
Figura 40: vista login	72
Figura 41: vista header de administrador luego de autenticación	72
Figura 42: vista header de usuario luego de autenticación.....	72
Figura 43: vista de productos	73
Figura 44: vista de la información del producto	73
Figura 45: selección de cantidad del producto	74
Figura 46: vista del mensaje “producto agregado al carrito”	74
Figura 47: vista del contenido del carrito	74
Figura 48: formulario completo de nueva categoría	75
Figura 49: vista de la nueva categoría	75
Figura 50: forma de registro de producto completa.	75
Figura 51: producto agregado	76
Figura 52: listado de productos	76
Figura 53: vista de productos vendidos.	76
Figura 54: mensaje de error	77
Figura 55: mensaje de error	77
Figura 56: validación en la base.....	78
Figura 57: vista del menú de administrador luego de autenticación con usuario administrador	78
Figura 58: vista del menú de usuario luego de autenticación con	

usuario normal	79
Figura 59: mensaje de error	79
Figura 60: mensaje de error	79
Figura 61: mensaje de error	80
Figura 62: listado de productos ingresados en el carrito	80
Figura 63: valor total a pagar	80
Figura 64: configuración del thread group para la prueba de carga.....	81
Figura 65: configuración del http request	81
Figura 66: muestra del resultado de los http request realizados.....	82
Figura 67: grafico de resultados de los http request.....	82
Figura 68: imagen de tiempos de respuesta.	83

1. Capítulo 1. Introducción

El presente trabajo muestra el desarrollo de un proyecto web que busca mejorar la difusión de los productos de la granja “Nono” de la Universidad de las Américas, además de exponer los beneficios de contar con elaboración tradicional, sana y ecológica y permitir que estos se posicionen mejor dentro de la comunidad estudiantil UDLA y sus allegados.

Dentro de la fundamentación teórica se presenta un análisis del proceso de negocio y de las actividades realizadas en la granja, las tecnologías, la metodología de desarrollo y las herramientas usadas para la elaboración de este proyecto. Para finalizar se presentarán las conclusiones, las recomendaciones y las referencias.

1.1 Antecedentes

La Universidad de Las Américas (UDLA), fue fundada en 1994, y ha ido creciendo gradualmente agregando carreras a su oferta académica para cumplir con las necesidades del país, además promueve el desarrollo de proyectos y la vinculación con la comunidad. En la actualidad cuenta con diversos campus y recursos para los estudiantes como es el caso de la Granja “Nono” en la cual varias carreras de la Universidad tienen la apertura para emprender y desarrollar diferentes proyectos. Por lo general existen visitas de la Facultad de Ingeniería y Ciencias Agropecuarias (FICA), especialmente de las Ingenierías Ambiental y Agroindustrial y de Veterinaria, pero también han existido iniciativas de las carreras de Arquitectura, Gastronomía, Cine, Multimedia (UDLA, 2018).

La Granja UDLA está ubicada en la parroquia de Nono, en las faldas del volcán Pichincha, a 22 Km de la ciudad de Quito, tiene una temperatura promedio de 14° a 16° C a 2800 metros sobre el nivel del mar. Consta de 44,6 hectáreas de terreno a las afueras de la parroquia, donde se integra: producción agropecuaria, transformación de productos, prácticas académicas, investigación y proyectos de vinculación con la comunidad, todos impulsados por la Universidad de las Américas con el objetivo de que estas actividades puedan formar parte del desarrollo integral y mejorar la calidad de vida de los habitantes de la zona.

La Granja UDLA es relativamente nueva, se encuentra en proceso de crecimiento y desarrollo. Está a disposición de las diferentes carreras de la UDLA que deseen participar en proyectos productivos, académicos y de vinculación en la comunidad de Nono.

Una particularidad de la granja es la producción y comercialización de productos cuyo fin es ser distribuidos en la comunidad UDLA ya que estos productos son elaborados de manera tradicional usando muy pocos químicos y además estos son de sello verde (UDLA, 2018). El principal problema es la falta de alcance de mercado y la incapacidad de llegar a más gente que puedan estar interesada en este tipo de productos, en la actualidad gran parte de la comunidad estudiantil desconoce de la existencia de estos productos, esto se debe principalmente a la falta de un medio dedicado a la venta e información sobre lo que ofrece la granja. En la actualidad los estudiantes del club Nono en acción son los encargados de gestionar la venta a la comunidad interna y externa de la UDLA, con lo cual recaudarán fondos para su actividad y para ayuda social (UDLA, 2018), y es debido a esto que se plantea este nuevo canal para dar más alcance a estos productos y ayudar a los estudiantes del club Nono en acción.

1.2 Alcance

Este proyecto de titulación propone desarrollar un sistema web que constituya un mecanismo más para que la comunidad UDLA pueda conocer y adquirir los productos de la Granja “Nono”, recalcando los beneficios de la fabricación tradicional, sana y ecológica que éstos presentan, es por esto que, se trabajará de manera muy estrecha con el/la encargado/a de la Granja, para obtener información más clara y precisa sobre los procesos y actividades que se desarrollan en la misma. Este sistema permitirá ingresar productos, su precio, una descripción y nombre. Además, estos productos serán distribuidos por categorías y se podrán buscar tanto por su nombre, parte de éste y/o su categoría, también, en este proyecto se crearán carritos de compras, los mismos que contarán con la opción de ingresar, modificar o eliminar los productos ingresados, también se mostrará el valor total a pagar y la opción de reserva. Una vez hecha la reserva estos productos reservados serán entregados una vez que se recepte el pago correspondiente en los puntos de venta designados.

Existirán roles de usuarios, lo cual permitirá que los Clientes (rol usuario) solo puedan ver y buscar productos, mientras que los Administradores (rol administrador) tendrán el control del sistema, pudiendo agregar, modificar, editar o actualizar, ya sea los productos (precio, nombre, descuento y descripción), como las categorías o incluso asignar permisos para un usuario nuevo.

Todos los usuarios deben estar creados en el sistema, para lo cual se deben registrar en el mismo con un correo electrónico válido y una clave. En caso de que este usuario requiera ser un Administrador, éste debe solicitar los permisos al o los usuarios administradores del sistema, para que sean ellos quienes validen la información y puedan asignar los permisos correspondientes.

No se implementará una pasarela de pagos ya que en la actualidad los pagos son únicamente en efectivo y en los puntos de ventas.

Se agregará además una página de información que servirá para informar a las personas dónde pueden retirar estos productos, indicando la dirección de los campus de la universidad, y ellos decidirán el lugar donde van a retirarlos

A manera de información se contará con espacios en los cuales se podrán ingresar, actualizar o eliminar información sobre la Granja, sus procesos o productos, con el objetivo de que las diferentes carreras puedan publicar su quehacer en prácticas propias. Esto sólo lo podrán hacer los Administradores del sistema.

Tanto los Usuarios como los Administradores tendrán la opción de cambiar su clave de acceso al sistema y los últimos además podrán asignar permisos de usuario.

Esta solución será implementada utilizando el lenguaje de programación PHP y se desarrollará en el framework LARAVEL, contará con una base de datos MySQL y Scrum como marco de trabajo.

1.3 Justificación

Una vez recopilados los datos de la situación actual y la información de contexto del proyecto, se puede apreciar que la Granja cuenta con diversas actividades

que producen productos finales y listos para su comercialización, los cuales son elaborados de una manera tradicional, evitando el uso excesivo de químicos, pesticidas, y tampoco han sido modificados genéticamente para darles una cualidad que no tenían como es el caso de los transgénicos (Martín, 2018), es por esto que se plantea un método más para la difusión de estos productos, permitiendo que los mismos se posicionen entre la comunidad estudiantil UDLA y sus allegados y posteriormente en el país. También permitirá que se obtengan más beneficios que con una comercialización tradicional, como la que se maneja actualmente. Es importante tener en cuenta que este proyecto solventará las necesidades de mercado al permitir que los productos de la granja tengan más alcance y por ende estos se vendan más. Después de analizar la situación con respecto al conocimiento de los productos que comercializa la granja, se puede apreciar que los estudiantes en general no tienen mayor conocimiento de los productos que ofrece, por lo que se considera pertinente tener un sistema web para lograr una inclusión total de estos los mismos en el mercado estudiantil, ya que éste no está siendo abarcado en su totalidad en la actualidad. (BBVA.S. A, 2019).

1.4 Objetivo General

Desarrollar un sistema web, para apoyar a la comercialización de los productos de la Granja “NONO” de la Universidad de las Américas.

1.5 Objetivos específicos

- Realizar fundamentación teórica para determinar el ambiente en que será usado el sistema.
- Realizar un análisis de requerimientos, tanto funcionales como no funcionales, para dimensionar de manera correcta y adecuada los entregables.
- Implementar la lógica de negocio para el desarrollo y comercialización de los productos de la granja “Nono”
- Validar la propuesta de la solución utilizando pruebas de calidad de software.

2. Capítulo 2. Marco teórico

2.1 Introducción

En este capítulo, se expondrá la fundamentación teórica del proyecto tomando como base el planteamiento del problema, y se tratará exclusivamente los temas relacionados al problema presentado, con la finalidad de que la unión de la información y los conocimientos sean expuestos de manera práctica, sencilla y altamente útiles.

Se expondrá la situación actual de la Granja “Nono” UDLA, y los medios actuales que con los que esta cuenta para dar a conocer sus productos y cómo estos medios realmente apoyan para la granja, así como los sistemas similares existentes en la actualidad. Posteriormente se describirán las necesidades existentes y los requerimientos planteados por los interesados. Se describirá la metodología que será usada para desarrollar la solución y también se presentarán las tecnologías y herramientas de software que serán usadas para solventar las necesidades planteadas en la introducción de este documento.

2.2 Granja UDLA

La Granja UDLA está ubicada en la parroquia de Nono, en las faldas del volcán Pichincha, a 22 Km de la ciudad de Quito, tiene una temperatura promedio de 14° a 16° C a 2800 metros sobre el nivel del mar. Consta de 44,6 hectáreas de terreno a las afueras de la parroquia, donde se integra: producción agropecuaria, transformación de productos, prácticas académicas, investigación y proyectos de vinculación con la comunidad, todos impulsados por la Universidad de las Américas con el objetivo de que estas actividades puedan formar parte del desarrollo integral y mejorar la calidad de vida de los habitantes de la zona.

La Granja UDLA es relativamente nueva, se encuentra en proceso de crecimiento y desarrollo. Está a disposición de las diferentes carreras de la UDLA que deseen participar en proyectos productivos, académicos y de vinculación en la comunidad de Nono.

Algunos de los productos que la Granja UDLA comercializa desde el año 2018 para la Comunidad UDLA son: huevos de gallina, frutilla, mora, uvilla, lechugas, rábanos, pimientos, babaco, taxo, miel de abeja, etc.

Si bien los productos no son completamente orgánicos, estos utilizan muy pocos químicos y además son de sello verde. Es importante resaltar que la meta principal de la Granja es avanzar hacia una producción 100% orgánica. (UDLA, 2018).

2.3 Comercio electrónico

2.3.1 ¿Qué es comercio electrónico?

“El comercio electrónico, conocido también como e-commerce, es una manera de comprar y vender productos o servicios, a través de redes informáticas y del internet, principalmente. Gracias a la masificación del uso del internet, que en Ecuador abarca a más de la mitad de la población. Este tipo de comercio creció vertiginosamente en el mundo, moviendo millones de dólares.

En el país, por ejemplo, las personas pueden efectuar la compra de un celular en tiendas electrónicas en el extranjero y recibirlas en la comodidad de su hogar, gracias a los servicios de paquetería de empresas de courier, como Correos del Ecuador E.P. De esta misma manera, miles de transacciones se efectúan por internet estimulando la creación y utilización de innovaciones en: la administración de cadenas de suministros, el marketing en Internet, el procesamiento de transacciones en línea, los sistemas de administración del inventario y los sistemas automatizados de recolección de datos.” (Ministerio de Telecomunicaciones y de la Sociedad de la Información, SF.)

2.3.2 Ventajas del comercio electrónico

Entre las principales ventajas del comercio electrónico encontramos:

- “Agiliza la compra, venta y distribución de bienes o servicios.
- Facilita las comunicaciones comerciales, ya que al estar disponible las 24 horas, las empresas pueden fidelizar a sus clientes, mediante un diálogo que sucede a conveniencia de las dos partes a través de la web, donde levantan requerimientos o simplemente hacen comentarios con relación a los productos o servicios que ofrecen.
- Beneficia a las operaciones comerciales, reduciendo errores, tiempo y sobrecostos en el tratamiento de la información.” (Ministerio de Telecomunicaciones y de la Sociedad de la Información, SF.)

2.3.3 Herramientas de distribución y comercialización

En la actualidad existen varias Herramientas para la distribución y comercialización en el Ecuador. Algunos ejemplos de estas herramientas son Glovo, Uber Eats, Mercado Libre, OLX. todas estas herramientas son importantes y su análisis permite comprender el comercio electrónico, lo cual sirve como punto de partida para el desarrollo de la solución.

Dentro de la UDLA no existen aplicaciones o herramientas de distribución, pero si comercialización. en el caso puntual de la Granja, se utiliza una comercialización tradicional y para clientes seleccionados se maneja un sistema de reserva de pedidos por medio de un correo electrónico.

2.4 Herramientas y aplicaciones similares

Actualmente en el mercado existen herramientas y aplicaciones similares a la solución propuesta. Entra las principales tenemos a OLX, Glovo, Mercado Libre, Booking, Despegar, Expedia.

“OLX es la plataforma líder mundial de clasificados, enfocada en mercados emergentes, tanto móviles como en línea.

OLX opera a nivel mundial y está presente en más de 40 países.” (OLX, 2016).

“Glovo es la app que te permite pedir los mejores productos de tu ciudad en minutos. Dinos lo que quieres y te lo entregamos donde tu digas.

¿Los mejores productos de la ciudad? ¿Como qué? Un cargador, comida, un smoothie, un plato caliente de tu restaurante preferido, las llaves de casa...

Solo piensa que deseas...” (Glovo, 2016).

Mercado Libre es “la compañía tecnológica de comercio electrónico líder de América Latina y que tiene como propósito democratizar el comercio y el dinero para impactar en el desarrollo de la región.

Nacida en 1999 y actualmente opera en 19 países de la región. En pocos años, logró convertirse en la séptima plataforma de comercio electrónico del mundo.

A través de un ecosistema completo compuesto por Mercado Pago, Mercado Shops, Mercado Libre Publicidad y Mercado Envíos” (Mercado Libre, 2018).

“Booking.com se fundó en Ámsterdam en 1996 y ha pasado de ser una pequeña start-up holandesa a una de las mayores empresas e-commerce de viajes de todo el mundo. Booking.com forma parte de Booking Holdings Inc. (NASDAQ: BKNG), y cuenta con más de 17.000 trabajadores en 198 oficinas de 70 países de todo el mundo.

Booking.com, cuya misión es poner el mundo al alcance de todos, apuesta por la tecnología digital para que viajar sea más fácil. En Booking.com conectamos a los viajeros con la oferta más amplia de alojamientos increíbles, como apartamentos, casas, resorts de lujo, bed & breakfasts de gestión familiar y hasta iglús y casas en los árboles. La web y apps de Booking.com están disponibles en más de 40 idiomas, y ofrecen 28.397.079 opciones de alojamiento en total, en 148.142 destinos de 228 países y territorios en todo el mundo.” (Booking.com, 2019)

“Despegar.com es la mayor agencia de viajes de América Latina. Con sede central en Miami y presencia en 21 países de América Latina, la compañía cuenta con 4.500 empleados. En su sitio web, www.despegar.com, se pueden comprar billetes para 750 compañías aéreas o realizar reservas en 150.000 hoteles, cruceros y agencias de alquiler de vehículos de todo el mundo. Con independencia de si compra a través de Internet o mediante su centro de atención al cliente, Despegar.com siempre le garantiza el precio más bajo del mercado.” (Digitalelement, 2014)

“Expedia.mx te permite organizar fácilmente las vacaciones a tu medida. En línea desde noviembre de 2009, Expedia.mx es el líder mundial en viajes online, con todo lo necesario para satisfacer tus necesidades. Nunca tuviste tan cerca:

- Un directorio de más de 510,000 hoteles en todo el mundo, con cuatro millones de habitaciones
- Alquiler de coches de las principales compañías

- Tecnología para fácilmente organizar tu viaje a medida” (Expedia.mx, 2019)

Todas estas aplicaciones son similares, pero a la vez diferentes de la solución propuesta en este documento. Se seleccionaron estas aplicaciones como base para entender los procesos que se presentan en un sistema de comercio electrónico, como por ejemplo el proceso de reserva de Booking.com o el proceso de búsqueda de Expedia o Mercado Libre.

Todas estas aplicaciones son similares, pero ninguna se alinea con los procesos de negocio propios de la granja Nono, por lo que la solución presentada en este documento es diferente y se alinea mejor con la problemática.

2.5 Marco de trabajo

El marco de trabajo que se utilizará para este proyecto será Scrum.

2.5.1 Definición e historia

“Scrum (n): Un marco de trabajo por el cual las personas pueden acometer problemas complejos adaptativos, a la vez que entregar productos del máximo valor posible productiva y creativamente. Scrum es:

- Ligerio
- Fácil de entender
- Extremadamente difícil de llegar a dominar.”. (Schwaber & Sutherland, 2013)

Scrum es un marco de trabajo que ha sido utilizado para la gestión del desarrollo de proyectos complejos desde principios de los 90. Scrum es un marco de trabajo en el cual se pueden emplear diversas técnicas y procesos. Scrum muestra el modo de mejorar las prácticas de gestión de producto y las prácticas de desarrollo.

El marco de trabajo Scrum contiene Equipos Scrum, roles, eventos, artefactos y reglas asociadas a cada uno de estos componentes, los cuales sirven a un propósito específico y esencial para el uso exitoso de Scrum. (Schwaber & Sutherland, 2013)

2.5.2 Roles, Artefactos y Actividades

Scrum propone:

- **Roles:**
 - **Scrum Máster (SM):** es el líder del equipo, y el encargado de eliminar cualquier obstáculo que impida la consecución de los objetivos. Es el encargado de delegar las tareas a sus colaboradores para que éstos se autoorganicen y alcancen un nivel de coordinación y colaboración exitoso.
 - **Product Owner (PO):** es el representante del cliente. En ocasiones es él mismo, aunque en otras ocasiones es un representante o tercero designado por el cliente. (Schwaber & Sutherland, 2013)
 - **Scrum Team (ST):** es el que se encarga de ejecutar las acciones previstas. Para el que el proceso sea exitoso, lo ideal podrían ser un mínimo de 5 personas y un máximo de 9, también se lo conoce como equipo de desarrollo. (Butler, 2018)
- **Artefactos:**
 - **Product Backlog** (Pila del producto): es la lista de requerimientos del usuario a partir de los cuales se inicia, crecimiento y evolución del producto durante el desarrollo de este.
 - **Sprint Backlog** (Pila del Sprint): es la lista de tareas y actividades que el equipo de desarrollo debe realizar durante el Sprint para lograr alcanzar el incremento establecido previamente.
 - **Incremento:** es el resultado de cada Sprint, el cual puede ser o no usado para ser puesto en uso, pruebas o como parte de la documentación.
- **Actividades:**
 - **Sprint Planning** (Planificación del Sprint): es una reunión establecida dentro de una jornada laboral en la cual, se define el trabajo a realizarse durante el Sprint, este plan es creado con el apoyo del todo el ST. Se pueden usar técnicas para la planeación como por ejemplo el Planning Poker, el cual es un proceso que es

efectuado por el equipo y que utiliza los puntos de historia para valorar el proyecto. (Najarro, 2017)

- **Sprint:** es la parte más importante de Scrum, es un espacio de tiempo de duración de más o menos 2 o 4 semanas, el cual entrega un incremento del producto, utilizable y potencialmente desplegable. Cada Sprint comienza cuando termina el anterior.
- **Daily Scrum** (Scrum diario): es una reunión de no más de 15 minutos de duración en la cual el ST informa de las actividades realizadas el día anterior, las actividades a realizarse el día de “hoy” y los principales problemas presentados, cuando estas reuniones terminan se actualiza el Sprint Backlog.
- **Sprint Review** (Revisión del Sprint): al final de cada Sprint se lleva a cabo una revisión informativa del Sprint con el fin de analizar el incremento y adoptar las recomendaciones o sugerencias para el próximo Sprint.
- **Sprint Retrospective** (Retrospectiva del Sprint): permite al ST realizar una introspectiva sobre su desempeño para poder encontrar sus fortalezas y debilidades con el fin de reforzarlas o elaborar un plan de mejora. (Schwaber & Sutherland, 2013) Para realizar una buena retrospectiva del Sprint se pueden usar técnicas como la de la estrella de mar, la cual permite identificar: que se debe seguir haciendo, que se debe empezar a hacer, que se debe dejar de hacer, las cosas que deben mejorar y las cosas que no aportaron valor alguno. (Veliz. Bernaola, 2009)

Los roles permiten que todas las personas que intervienen, o tienen relación directa o indirecta con el proyecto puedan entender cómo se dividirán las responsabilidades que estos tienen durante el desarrollo del proyecto.

Los artefactos son herramientas gracias a los cuales el equipo de trabajo puede desarrollar y controlar el proyecto. La principal ventaja del framework es su facilidad para adaptarse a las necesidades del equipo de trabajo y al contexto del proyecto ya que el uso de artefactos no es mandatorio.

Las actividades y los tiempos son parte del proceso de realización del proyecto, los cuales permiten la generación de ciclos autoorganizados y evaluables con el fin de dar en todo momento un enfoque sobre el producto. (Gutiérrez, S.F)

2.5.3 Ciclo de vida de Scrum

Dentro del marco de trabajo Scrum todo proyecto comienza con el Product Owner, el cual es el responsable de crear e informar el producto backlog, el cual está constituido por historias de usuario que son las responsables de mostrar un aspecto o funcionalidad que debe tener el producto. luego el propietario del producto debe priorizar estas historias y les brinda un orden.

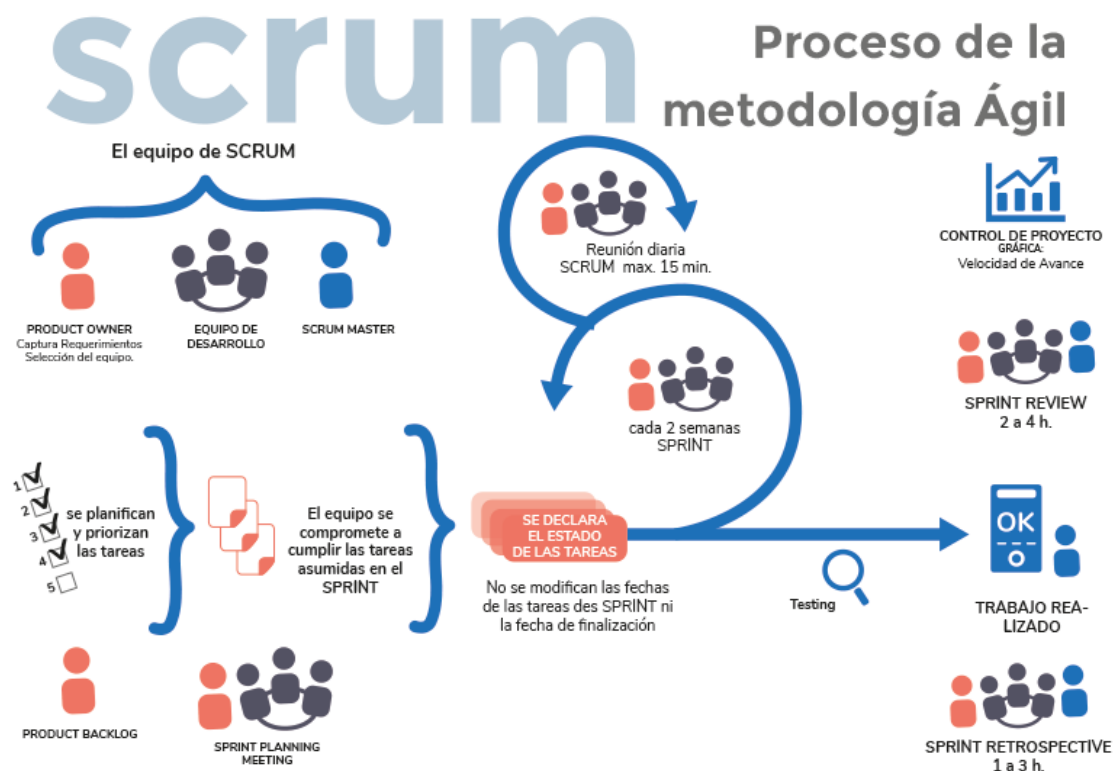


Figura 1: Ciclo de vida de Scrum

(Adaptado de Grupo Garatu Development)

Para cada Sprint del proyecto, el ST debe seleccionar cuales son las funcionalidades del Product Backlog que van a ser presentadas en el Sprint Backlog presente en esa interacción, posteriormente el equipo de desarrollo convierte las historias de usuario en tareas y se estima el tiempo de duración de cada tarea, una vez completada el Sprint Planning y mientras dure el Sprint, se

elaboran las Daily Meetings para presentar las tareas que se realizaron el día anterior, lo que se hará ese día y los problemas presentes, una vez que las reuniones terminan se procede a realizar las tareas asignadas.

El proceso se repite hasta que termina el Sprint y es entonces cuando se lleva a cabo el Sprint Review en el cual el usuario es el encargado de aceptar o rechazar el incremento. Por último, el SM y el ST llevarán a cabo el Scrum Retrospective en el cual se analizarán las áreas a mejorar, e iniciarán un nuevo Sprint. (ScrumManager, 2016)

2.5.4 Medición y estimación ágil

La medición y la estimación son necesarios para poder analizar el rendimiento, la organización, la eficiencia del ST y la satisfacción del cliente. Tenemos 2 niveles de análisis, uno es el nivel de desarrollo, en el cual están presentes factores como: la cantidad de fallas, la disponibilidad, el tiempo empleado y la complejidad del proyecto, y por otro lado tenemos el nivel del proyecto, en el cual los factores presentes son: costos, tiempos y esfuerzos.

esta información es importante porque permite tomar decisiones, ya que entrega criterios claros de control y gestión, también ayudan a planificar mejor los Sprints, tener un mayor control del estado del proyecto y a cumplir con los tiempos de ejecución de cada una de las tareas.

Como principal insumo de este sistema de estimación tenemos 2 tipos de gráficos: Burn Up Chart y Burn-Down Chart.

- Burn Up Chart o gráfico del producto: está a cargo del PO y una herramienta que ayuda a la planeación, ya que presenta de forma visual la evolución previsible del proyecto, midiendo el esfuerzo necesario para elaborar las historias de usuario, ya sean en tiempo real o en Sprints.
- Burn-Down Chart o gráfico de avance: está a cargo del ST ya que ellos son los encargados de actualizar este gráfico en las reuniones de Sprint Review. Con él, se puede monitorear el ritmo de progreso, y/o detectar posibles desviaciones en el proceso. este gráfico es esencial para el Daily

Scrum ya que permite calcular el esfuerzo restante. (ScrumManager, 2016)

2.5.5 Herramientas para llevar Scrum

La herramienta administrativa Azure DevOps fue usada para llevar el proceso de Scrum, ya que es una herramienta que permite trabajar con hasta 10 trabajos en paralelo y de forma gratuita, permite utilizar paneles Kanban, permite registrar el trabajo hecho y el pendiente, permite el trabajo en equipo y también permite generar informes personalizados por cada miembro del equipo de trabajo. (Microsoft Azure, 2019)

2.6 Tecnologías de desarrollo

2.6.1 Framework de desarrollo

Un framework o marco de trabajo es usado ampliamente en el desarrollo de software. Framework “significa que ya se ha diseñado un marco y que el desarrollador debe funcionar en ese marco para cumplir con los requisitos de su proyecto. Es solo una herramienta que ayuda al desarrollador a codificar mejor y más rápido.” (Ortiz, 2018)

Los frameworks no son más que estructuras de software compuestos de componentes personalizables que son usados para el desarrollo de un sistema informático. (Gutiérrez, S.F).

Las principales ventajas de utilizar un framework son:

“El programador no necesita plantearse una estructura global de la aplicación, sino que el framework le proporciona un esqueleto que hay que “rellenar”.

Facilita la colaboración. Cualquiera que haya tenido que “pelearse” con el código fuente de otro programador (¡jo incluso con el propio, pasado algún tiempo!) sabrá lo difícil que es entenderlo y modificarlo; por tanto, todo lo que sea definir y estandarizar va a ahorrar tiempo y trabajo a los desarrollos colaborativos.

Es más fácil encontrar herramientas (utilidades, librerías) adaptadas al framework concreto para facilitar el desarrollo.” (Jordisan.net, 2006)

Para el desarrollo de este proyecto se empleó un framework web con patrón Modelo - Vista – Controlador (MVC). Este patrón de diseño separa a la solución en 3 capas. En la primera capa (modelo) están los datos de la aplicación y las reglas de negocio. En la segunda capa (vista) están todos los formularios, vistas, salidas y entradas de datos de la aplicación esto quiere decir que es la parte visual de la aplicación. Por último, tenemos la tercera capa (controlador) en la cual se encuentran todos los controladores del sistema, y que son los responsables de procesar las peticiones de los usuarios, permitiendo que el flujo de ejecución del sistema sea el adecuado (Gutiérrez, S.F).

2.6.1.1 Laravel

Para este sistema se seleccionó el framework Laravel debido a su extensa documentación.

“Laravel es uno de los frameworks de código abierto más fáciles de asimilar para PHP. Es simple, muy potente y tiene una interfaz elegante y divertida de usar. Fue creado en 2011 y tiene una gran influencia de frameworks como Ruby on Rails, Sinatra y ASP.NET MVC.” (García, 2015).

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

Fue creado en 2011 y actualmente está en continuo desarrollo. Este framework usa el paradigma Orientado a objetos, permite el uso del patrón MVC, Mapeo objeto-relacional (ORM). Gran parte de Laravel está formado por dependencias, especialmente de Symfony, esto implica que el desarrollo de Laravel dependa del desarrollo de sus dependencias.” (García, 2015).

Entre los principales componentes de este framework tenemos:

- **Eloquent:** es el ORM que incluye Laravel para administrar de forma fácil y sencilla los procesos correspondientes al manejo de bases de datos del proyecto, gracias a las funciones que provee se pueden realizar consultas

complejas y peticiones a la base de datos sin escribir una sola línea de código SQL. (Styde, 2015)

- **Composer:** es un manejador de dependencias. Este siempre es instalado de manera local, y contiene las librerías que están ubicadas en la ruta /vendor. Composer es capaz de instalar cualquier librería que sea necesario para el proyecto y en la versión que sea y resuelve todas las dependencias indirectas y descarga automáticamente la versión correcta de cada paquete (Styde, 2014)
- **Symfony:** es un proyecto PHP de software libre que permite crear aplicaciones y sitios web rápidos y seguros de forma profesional (Symfony.es, SF)

2.6.1.1.2 Patrón Modelo Vista Controlador

MVC es un acrónimo de modelo vista controlador, en donde el modelo es una representación de una instancia u objeto de la vida real en el código base, la vista es la interfaz a través de la cual el usuario interactúa con la aplicación y el controlador es el encargado de procesar las acciones del usuario. (Laravel, 2018)

Laravel implementa este patrón desde el momento que se crea un proyecto. Al ser Laravel un Framework que usa el paradigma de programación orientado a objetos, se puede hacer uso del patrón MVC, además, al contar muchas dependencias entre sus componentes, se puede hacer uso de este patrón para que se desarrollen sus dependencias de una manera óptima y refinada, cumpliendo así, con la filosofía de Laravel. Pero Laravel no solo plantea esto, sino que también plantea el uso de Routes with Closures, que reemplaza al patrón MVC tradicional con el fin de generar un código más claro. (Laravel, 2018)

2.6.2 Lenguaje de programación

El lenguaje de programación seleccionado para la elaboración de esta solución es PHP (Hypertext Preprocessor), el cual es un lenguaje de Código abierto y “del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación.

Un lenguaje del lado del servidor es aquel que se ejecuta en el servidor web, justo antes de que se envíe la página a través de Internet al cliente. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente. El cliente solamente recibe una página con el código HTML resultante de la ejecución de la PHP. Como la página resultante contiene únicamente código HTML, es compatible con todos los navegadores. Podemos saber algo más sobre la programación del servidor y del cliente en el artículo “qué es DHTML”. (Álvarez, 2001).

Una de las principales ventajas de PHP es la simplicidad que tiene, lo cual permite que incluso un principiante pueda trabajar con este lenguaje, pero también cuenta con varias opciones avanzadas para los desarrolladores con bastante experiencia. (PHP, 2018).

2.6.3 Base de datos

“Una base de datos es una colección estructurada de datos. Puede ser cualquier cosa, desde una simple lista de compras hasta una galería de imágenes o la gran cantidad de información en una red corporativa. Para agregar, acceder y procesar los datos almacenados en una base de datos de computadora, se necesita un sistema de administración de base de datos como MySQL Server. Dado que las computadoras son muy buenas para manejar grandes cantidades de datos, los sistemas de administración de bases de datos desempeñan un papel central en la informática, como utilidades independientes o como parte de otras aplicaciones” (MySQL, 2018). Y cuenta con el respaldo de Oracle Corporation, ya que ellos son los encargados de su distribución, mantenimiento y desarrollo.

El sistema administrador de base de datos seleccionado para este proyecto es MySQL, el cual emplea el lenguaje SQL (Structured Query Language o lenguaje de consulta estructurada), este lenguaje es el lenguaje estandarizado más utilizado para acceder a las bases de datos y que está definido por el standard ANSI/ISO, el cual ha ido evolucionando desde su primera aparición en 1986.

MySQL es una base de datos relacional y al ser una base de este tipo los datos son almacenados en tablas separadas para evitar colocar todos los datos dentro de un solo dispositivo de almacenaje. Dentro de las estructuras de la base de datos existen ficheros físicos que están optimizados (tunning) para brindar una mejor velocidad y un mejor procesamiento de la información. Además de estos ficheros físicos existe también, el modelo lógico. El modelo lógico está formado por los objetos en la base de datos cómo, por ejemplo: tablas, vistas, filas y columnas, etc.

Una de las principales ventajas de usar MySQL es que es código abierto lo cual permite que cualquier usuario lo use y si desea modifique el software.

Otra de las principales ventajas de MySQL es su rapidez, escalabilidad, facilidad de uso, confiabilidad, la habilidad de ser ejecutado desde cualquier computadora, sea esta una computadora con altas o bajas prestaciones y su capacidad de integración con otras aplicaciones o servidores.

MySQL es usado, generalmente, en sistemas cliente-servidor, el cual permite que varios clientes se conecten al servidor de back-end, ya sea con la ayuda de herramientas o administrativas (DBMS), aplicaciones o programas de cliente, librerías o cualquiera de las herramientas de programación. (MySQL 2018)

2.7 Pruebas de verificación y validación

Las pruebas de verificación y validación sirven para comprobar que los procesos de comprobación y análisis que aseguran que el software que se desarrolla está acorde a la especificación y cumple con las necesidades de los clientes. Se inicia con la revisión de los requerimientos y continúa con el diseño y con la inspección del código, para terminar con las pruebas del producto. Existen actividades de este tipo en cada etapa del proceso de desarrollo del software.

Verificación: ¿Se está construyendo el producto correctamente?

El papel de la verificación consiste en comprobar que el software está de acuerdo con su especificación. Se comprueba que el sistema cumple los requerimientos funcionales y no funcionales que se le han especificado.

Validación: ¿Se está construyendo el producto concretamente?

La validación es un proceso más general. Se debe asegurar que el software cumple las expectativas del cliente. Va más allá de comprobar si el sistema está acorde con su especificación. (Drake y López, 2009)

Alguna de las pruebas que se pueden ejecutar son por ejemplo las pruebas de caja negra, caja blanca y casos de prueba.

Pruebas de caja negra: Estas pruebas permiten obtener un conjunto de condiciones de entrada que involucren a todos los requerimientos funcionales. En ellas se ignora la estructura de control, concentrándose en los requisitos funcionales del sistema y ejercitándolos. Son un complemento de las pruebas de caja blanca. (EcuRed, SF)

Pruebas de caja blanca: Estas pruebas se basan en el diseño de casos de prueba que usan la estructura de control del diseño procedimental para derivarlos. Es por esto, que se considera a las pruebas de caja blanca como uno de los tipos de pruebas más importantes que se le aplican al software, logrando como resultado, que se disminuya en un gran porcentaje el número de errores existentes y por ende se entrega software con mayor calidad y confiabilidad. (EcuRed, SF)

Casos de prueba: incluyen todas las funcionalidades del programa. Los casos de prueba deben tener en cuenta el uso de todo tipo de datos de entrada/salida, cada comportamiento esperado, todos los elementos de diseño, y cada clase de defecto. (Testeando Software, 2013)

Para este proyecto se usarán las pruebas de caja negra ya que estas serán usadas para que las personas encargadas de la granja Nono puedan aceptar el sistema.

2.7.1 Apache Jmeter

Apache JMeter es un software de código abierto, diseñado 100% en Java. Esta herramienta puede ser usada para probar el rendimiento de recursos los dinámicos y estáticos de un sistema web. Este software puede simular una carga pesada de trabajo en el servidor o grupos de servidores, redes u objetos para probar su resistencia, y poder analizar posibles cuellos de botella y problemas

de rendimiento de una aplicación web con diferentes cargas de trabajo (Apache, 2019)

3. Capítulo 3. Propuesta de Solución

3.1 Introducción

En este capítulo, se expondrá la propuesta de la solución planteada con el objetivo de mejorar y apoyar a la difusión de los productos de la granja Nono de la UDLA, además se demostrará el uso de Scrum como marco de trabajo para el desarrollo de esta aplicación. Se documentará la realización de los Product Backlog y los Sprint Backlogs obtenidos durante la realización de este proyecto, también se presentarán las gráficas Burn up y Burn down las cuales serán actualizadas durante el desarrollo de esta solución. Al completar cada Sprint se presentará el resultado obtenido (incremento), así como también un análisis de este. Se presentará el diagrama entidad-relación de la base de datos, y por último se presentarán vistas funcionales de la aplicación.

3.2 Product Backlog

A continuación, se presenta en una tabla denominada tabla 1, la lista completa de las historias de usuario que fueron requeridas para el desarrollo de esta solución.

La tabla 1 consta de 5 columnas:

1. La columna 1 contiene el identificador único para la historia de usuario
2. La columna 2 contiene una descripción de la historia de usuario.
3. La columna 3 contiene la Prioridad que indica el nivel de importancia que tiene cada historia de usuario según el Product Owner.
4. La columna 4 es una estimación del esfuerzo requerido para completar dicha tarea o requerimiento, para esta estimación se propone el uso de números.
5. La columna 5 contiene los resultados esperados, los cuales permiten saber si lo que desarrollado está acorde con las necesidades.

El total de puntos de historia (PH) es la sumatoria de todos los valores presentes en la columna 4, que en este caso es de 38 PH.

Tabla 1.

Product Backlog

Id	Nombre	Descripción	Prioridad	Pre-estimación	Criterio de validación
1	CRUD de categorías	Como administrador se debe ser capaz de ingresar, eliminar o modificar categorías dentro del sistema, así como también deberá ser capaz de subir una imagen representativa para la categoría, para permitir una mejor búsqueda e identificación de la categoría	Alta	5	<ul style="list-style-type: none"> • Se debe poder visualizar las actualizaciones y/o creaciones inmediatamente en el sistema. • Se debe poder la imagen ingresada
2	CRUD productos	Como administrador se debe poder ser capaz de ingresar, eliminar o modificar los productos dentro del sistema para	Alta	4	<ul style="list-style-type: none"> • Se debe poder visualizar las actualizaciones y/o creaciones inmediatamente en el sistema.

		tener un catálogo actualizado			
3	Búsqueda sin autenticación	Como usuario se debe poder ingresar a la página de “bienvenida” para poder buscar productos ya sea por su nombre o por categoría, pero no debería ser capaz de reservar ningún producto hasta que inicie sesión.	Media	3	<ul style="list-style-type: none"> • Vista de la página de bienvenida • búsqueda de productos • Página de login • Usuario registrado puede hacer login
4	Ingresar cantidad	Como cliente se debe poder seleccionar la cantidad de un mismo producto que desea para que esta cantidad sea reservada.	Media	3	<ul style="list-style-type: none"> • El usuario puede indicar la cantidad de producto que desea adquirir y esta cantidad debe verse reflejada en el carrito de compras

5	Carrito de compras	Como usuario se debe poder visualizar los productos ingresados al carrito de compra antes de proceder a realizar la reserva, y de ser el caso poder eliminarlos, o revisar la información de este para poder realizar una reserva acorde con las necesidades que tenga.	Alta	5	<ul style="list-style-type: none"> El usuario autenticado en el sistema debe poder revisar la lista de productos que agregó al carrito, así como la cantidad deseada de estos productos, el precio individual subtotal, el descuento, el total a pagar y la posibilidad de eliminar o de revisar la información de cada producto
6	Confirmación	Como administrador se debe poder recibir un correo con la	Alta	5	<ul style="list-style-type: none"> El administrador y el cliente

		información del cliente, la lista de los productos reservados y el valor a pagar, para poder separar su pedido.			deben recibir un correo con los siguientes datos del cliente: Nombre E-mail Teléfono Dirección Fecha Listado de productos El valor que debe pagar
7	Puntos de venta	Como usuario se debe poder revisar la ubicación de los puntos de venta, así como también ver un mapa para poder llegar al punto de venta.	Baja	1	<ul style="list-style-type: none"> • Poder visualizar la ubicación y los horarios de atención de los carros de la granja • Debe desplegarse en una ventana aparte un mapa con la

					ubicación del campus
8	Ingreso imágenes	Como administrador se debe ingresar imágenes de un producto y poder resaltar una imagen como la más importante del mismo, para resaltar el producto	Alta	5	<ul style="list-style-type: none"> • El administrador del sistema debe poder subir una o varias imágenes al sistema, así como poder visualizar las imágenes y también poder seleccionar una como destacada
9	Contacto	Como usuario se debe ser capaz de comunicarse con el administrador del sistema a través de un correo electrónico para solventar sus dudas o necesidades	Baja	2	<ul style="list-style-type: none"> • Un usuario puede enviar un correo desde la forma de "contáctenos" para solicitar información, o si tiene

					<p>alguna inquietud.</p> <ul style="list-style-type: none"> • El Administrador debe recibir el correo y poder responder.
10	Información	Como usuario se debe poder ingresar a una página informativa para conocer más sobre la granja.	Baja	2	<ul style="list-style-type: none"> • Las encargadas de la granja deben poder crear, actualizar o eliminar la información de la granja en una página web.
11	Productos más solicitados	Como administrador se debe poder revisar un histórico de los productos reservados para poder analizar mejor que productos son los más reservados	Media	3	<ul style="list-style-type: none"> • El administrador debe poder revisar la lista de todos los productos reservados hasta la fecha.

Esta tabla se muestra la pre-estimación hecha por el Product Owner para el desarrollo de la solución. Esta pre-estimación no es la definitiva y esta puede variar dependiendo de diversos factores que se pueden presentar, por ejemplo: que un integrante del equipo de desarrollo se enferme, que ocurran desastres naturales, el factor humano o la habilidad que tiene un equipo de desarrollo para desarrollar una historia de usuario.

3.3 Sprints

Tomando en cuenta las historias de usuario y la prioridad de estas se decidió realizar 2 Sprints con una duración de 2 semanas cada uno.

- Primer Sprint: este Sprint está compuesto por las historias de usuario con identificador: 1,2,3,4,5. Estas historias de usuario tiene un total de 19 PH
- Segundo Sprint: Sprint: este Sprint está compuesto por las historias de usuario con identificador: 6,7,8,9,10,11. Estas historias de usuario tiene un total de 19 PH.

El primer Sprint contiene todas las historias de usuario relacionadas a la gestión de productos, categorías y el carrito de compras, y también la búsqueda de productos.

El segundo Sprint contiene las historias de usuario que entregan información, y mejora del aspecto visual del software.

3.3.1 Flujo del primer Sprint

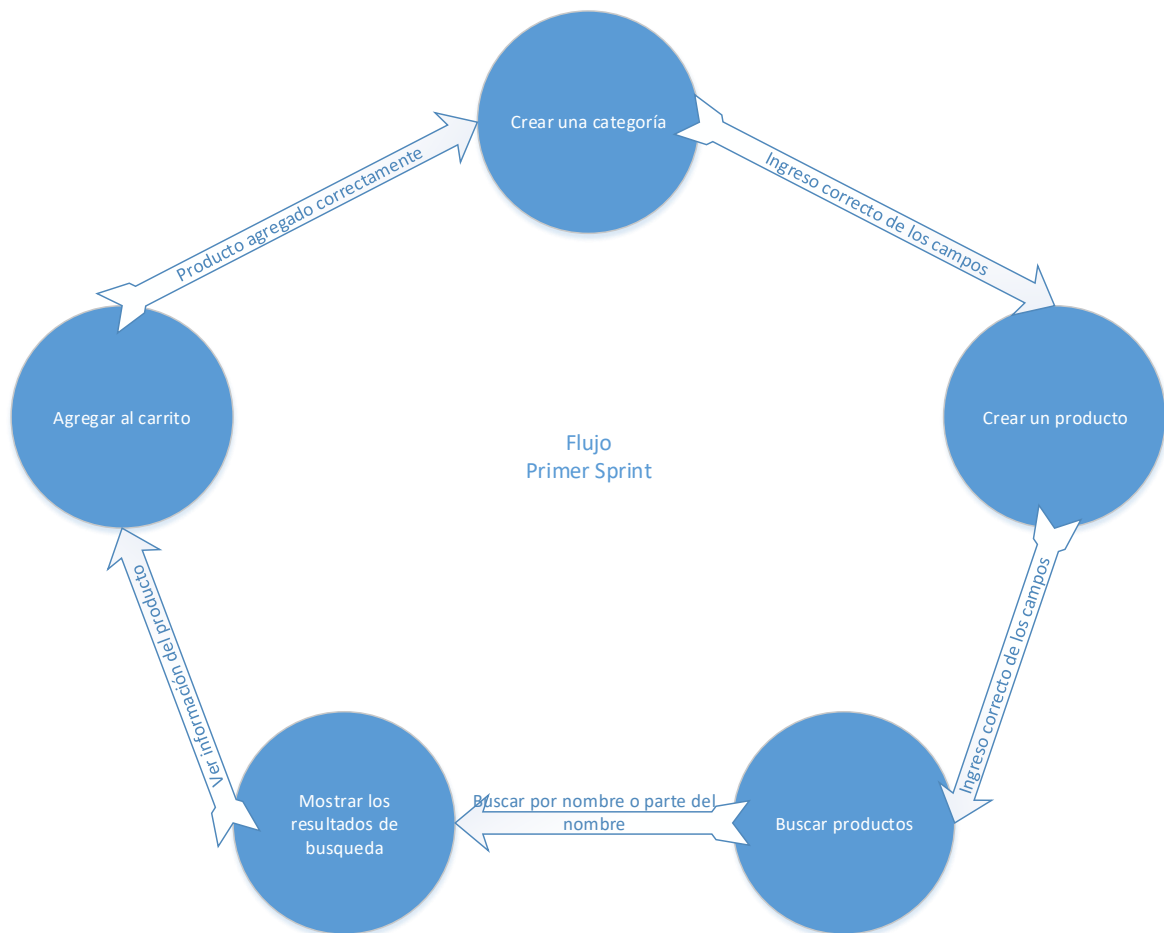


Figura 2: flujo de trabajo realizado en el primer Sprint

3.3.2 Flujo del segundo Sprint

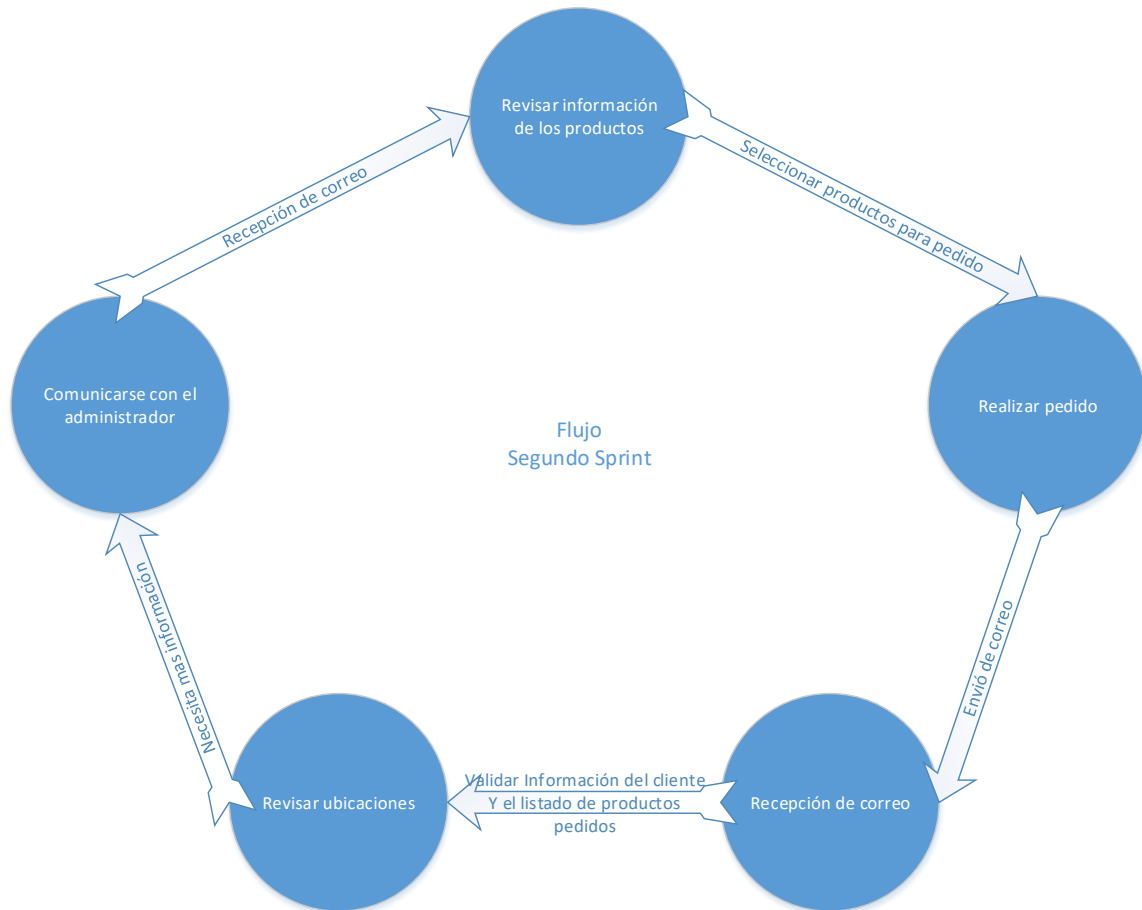


Figura 3: flujo de trabajo realizado en el segundo Sprint

3.4 Primer Sprint

3.4.1 Sprint Planning

Aplicando la técnica Planning Poker se pudo estimar y seleccionar las historias de usuario que van a ser desarrolladas en el primer Sprint, considerando las opiniones de todo el equipo, así como también permitir que los integrantes tengan más compromiso con el proyecto, para este primer Sprint se seleccionaron las historias de usuario 1, 2, 3, 4, 5.

3.4.2 Sprint Backlog

En el Sprint Backlog del primer Sprint se encuentran listadas las historias de usuario con identificador 1, 2, 3, 4, 5 y que están descritas en la tabla 1 ya que estas se encuentran agrupadas por actividades que permiten generar un incremento

A continuación, se muestra la tabla correspondiente al Sprint Backlog del primer Sprint.

Dentro de la tabla se tiene las siguientes columnas: id, descripción, esfuerzo, y prioridad.

- La columna id indica el número de historia de usuario, la columna nombre muestra el nombre de la historia de usuario
- La columna descripción contiene la información de la historia de usuario desarrollada.
- La columna esfuerzo indica el esfuerzo real que fue necesario para completar el desarrollo de esa historia de usuario.
- La columna prioridad indica que tan crítico es desarrollar esa historia de usuario.

Tabla 2:

Sprint Backlog del primer Sprint

ID	Nombre	Descripción	Estado	Esfuerzo	Prioridad	Responsable
1	Administrador de categorías	Como administrador se debe ser	Completado	6	Alta	Hernán Verdezoto

		capaz de ingresar, eliminar o modificar categorías dentro del sistema, así como también deberá ser capaz de subir una imagen representativa para la categoría, para permitir una mejor búsqueda e identificación de la categoría				
2	Administrador de productos	Como administrador se debe poder ser capaz de ingresar, eliminar o modificar los productos dentro del sistema para tener un catálogo actualizado	Completo	5	Alta	Hernán Verdezoto
3	Búsqueda	Como usuario se debe poder ingresar a la página de "bienvenida" para poder buscar productos ya sea por su nombre o por categoría, pero no debería ser capaz de reservar ningún producto hasta que inicie sesión.	Completo	4	Media	Hernán Verdezoto

4	Seleccionar cantidad	Como cliente se debe poder seleccionar la cantidad de un mismo producto que deseo para que esta cantidad sea reservada.	Completo	4	Media	Hernán Verdezoto
5	Información carrito de compras	Como usuario se debe poder visualizar los productos ingresados al carrito de compra antes de proceder a realizar la reserva, y de ser el caso poder eliminarlos, o revisar la información de este para poder realizar una reserva acorde con las necesidades que tenga.	Completo	6	Alta	Hernán Verdezoto

3.4.3 Daily Scrum

Cada día el equipo se reunió durante 15 minutos con el fin de responder a las 3 inquietudes, ¿Qué se hizo ayer?, ¿Qué se va a hacer hoy? y ¿Qué problemas se presentaron? Una vez respondidas las 3 interrogantes se procede a actualizar la gráfica Burn Down de Sprint, para poder observar el comportamiento y avance logrado.

En el primer Sprint no se presentaron inconvenientes en el avance durante los primeros días, pero posteriormente se presentaron demoras en el avance, por lo cual en el tercer día cuando se empezaron a evidenciar estas demoras, por lo que el Scrum Master tomó acciones para tratar de estabilizar el proceso.

3.4.4 Sprint Review

Al finalizar este Sprint se llevó a cabo un Sprint Review, en donde revisó lo que se trabajó y el incremento presentado que aporta valor para el cliente, además se obtienen conclusiones, recomendaciones y experiencias que fueron compartidos con el equipo, y que servirán para los siguientes Sprints o proyectos futuros.

En el primer Sprint se obtuvo el “core” de la aplicación.

3.4.5 Código del primer Sprint

```
CategoryController.php
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6  use App\Category;
7
8  class CategoryController extends Controller
9  {
10
11     public function show(Category $category)
12     {
13         $products = $category->products()->paginate(10);
14         return view('categories.show')->with(compact('category', 'products'));
15     }
16
17
18 }
```

Figura 4: código fuente controlador global “categorías”


```
CategoryController.php
1  <?php
2
3  namespace App\Http\Controllers\Admin;
4
5  use Illuminate\Http\Request;
6  use App\Http\Controllers\Controller;
7  use App\Category;
8  use File;
9  use App\Product;
10 use Illuminate\Support\Facades\Session;
11
12 class CategoryController extends Controller
13 {
14     public function index()
15     {
16         $categories = Category::orderBy('id')->paginate(10);
17         return view('admin.categories.index')->with(compact('categories'));
18     }
19
20     public function create()
21     {
22         return view('admin.categories.create');
23     }
24
25     public function store(Request $request)
26     {
27         $this->validate($request, Category::$rules);
28         $category = Category::create($request->only('name', 'description'));
29         if($request->hasFile('image')) {
30             $file = $request->file('image');
```

Figura 5: código fuente controlador con autorización “categorías” parte 1

```
CategoryController.php
31     $path = public_path() . '/images/categories';
32     $fileName = uniqid() . '-' . $file->getClientOriginalName();
33     $moved = $file->move($path, $fileName);
34
35     if ($moved)
36     {
37         $category->image = $fileName;
38         $category->save();
39     }
40 }
41 return redirect('admin/categories');
42 }
43
44
45 public function edit(Category $category)
46 {
47     return view('admin.categories.edit')->with(compact('category'));
48 }
49
50
51 public function update(Request $request, Category $category)
52 {
53
54     $this->validate($request, Category::$rules);
55
56     $category->update($request->only('name','description'));
57
58     if($request->hasFile('image')) {
59         $file = $request->file('image');
```

Figura 6: código fuente controlador con autorización “categorías” parte 2

```

CategoryController.php
60     $path = public_path() . '/images/categories';
61     $fileName = uniqid() . '-' . $file->getClientOriginalName();
62     $moved = $file->move($path, $fileName);
63
64     if ($moved)
65     {
66         $previousPath = $path . '/' . $category->image;
67         $category->image = $fileName;
68         $saved = $category->save();
69
70         if($saved)
71             File::delete($previousPath);
72     }
73 }
74
75 return redirect('admin/categories');
76 }
77
78
79 public function destroy(Category $category)
80 {
81     $msg='';
82     $cont = Product::where('category_id', $category->id)->count();
83     if ($cont <= 0){
84         $category->delete();
85         $msg = "Categoria eliminada";
86     }else{
87         $msg = "No es posible eliminar la categoria porque tiene productos relacionados";
88     }

```

Figura 7: código fuente controlador con autorización “categorías” parte 3

```

        Session::flash('msg', $msg);
        return back();
    }
}

```

Figura 8: código fuente controlador con autorización “categorías” parte 4

```
ProductController.php
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6  use App\Product;
7
8  class ProductController extends Controller
9  {
10     public function show($id)
11     {
12         $product = Product::find($id);
13         $images = $product->images;
14
15         $imagesLeft = collect();
16         $imagesRight = collect();
17         foreach ($images as $key => $image) {
18             if($key%2 == 0)
19                 $imagesLeft->push($image);
20             else
21                 $imagesRight->push($image);
22         }
23         return view('products.show')->with(compact('product', 'imagesLeft', 'imagesRight'));
24     }
25 }
26
27
28 }
```

Figura 9: código fuente controlador global “productos”

```
ProductController.php
5 use App\Http\Controllers\Controller;
6 use Illuminate\Http\Request;
7 use App\Product;
8 use App\Category;
9 use App\CartDetail;
10 use App\ProductImage;
11 use Illuminate\Support\Facades\Session;
12
13 class ProductController extends Controller
14 {
15     public function index()
16     {
17         $products = Product::paginate(10);
18         return view('admin.products.index')->with(compact('products'));
19     }
20
21     public function create()
22     {
23         $categories = Category::orderBy('name')->get();
24         return view('admin.products.create')->with(compact('categories'));
25     }
26
27     public function store(Request $request)
28     {
29
30         $rules = [
31             'name' => 'required|min:3',
32             'description' => 'required|max:200',
33             'price' => 'required|numeric|min:0',
```

Figura 10: código fuente controlador autenticado “productos” parte 1

```
ProductController.php
34     'discount' => 'required|numeric|min:0',
35     ];
36     $this->validate($request, $rules);
37
38
39     $product = new Product();
40     $product->name = $request->input('name');
41     $product->description = $request->input('description');
42     $product->price = $request->input('price');
43     $product->discount = $request->input('discount');
44     $product->long_description = $request->input('long_description');
45     $product->category_id = $request->category_id;
46     $product->save();
47
48     return redirect('admin/products');
49 }
50
51 public function edit($id)
52 {
53     $categories = Category::orderBy('name')->get();
54     $product = Product::find($id);
55     return view('admin.products.edit')->with(compact('product', 'categories'));
56 }
57
58 public function update(Request $request, $id)
59 {
60     //validacion
61     $rules = [
62         'name' => 'required|min:3',
63     ];
```

Figura 11: código fuente controlador autenticado “productos” parte 2

```
63     'description' => 'required|max:200',
64     'price' => 'required|numeric|min:0',
65 ];
66 $this->validate($request, $rules);
67
68     $product = Product::find($id);
69     $product->name = $request->input('name');
70     $product->description = $request->input('description');
71     $product->price = $request->input('price');
72     $product->discount = $request->input('discount');
73     $product->long_description = $request->input('long_description');
74     $product->category_id = $request->category_id;
75     $product->save();
76
77     return redirect('admin/products');
78 }
79
80 public function destroy($id)
81 {
82     CartDetail::where('product_id', $id)->delete();
83     ProductImage::where('product_id', $id)->delete();
84
85     $product = Product::find($id);
86     $product->delete();
87
88     Session::flash('msg', 'Se eliminó el producto y las imágenes asociadas');
89     return back();
90 }
91 }
```

Figura 12: código fuente controlador autenticado “productos” parte 3

```

3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use Carbon\Carbon;
7 use App\User;
8 use App\Mail\NewOrder;
9 use Mail;
10
11 class CartController extends Controller
12 {
13
14     public function update()
15     {
16         $client = auth()->user();
17         $cart = $client->cart;
18         $cart->status = 'Pending';
19         $cart->order_date = Carbon::now();
20         $cart->save();
21         $users = User::where('admin', false)->get();
22         $admins = User::where('admin', true)->get();
23         Mail::to($admins)->send(new NewOrder($client, $cart));
24         Mail::to($client)->send(new NewOrder($client, $cart));
25         $msg = 'Tu pedido se ha registrado correctamente. Te contactaremos pronto vía mail!';
26         return back()->with(compact('msg'));
27     }
28 }

```

Figura 13: código fuente controlador “carrito”

3.4.6 Capturas de pantalla del incremento del primer Sprint



Figura 14: menú creado después del primer Sprint

Listado de Productos



Figura 15: botón para agregar nuevo producto

Registrar nuevo producto

	Tomate de árbol malla x 20	\$ 2.65
	Tomate de árbol malla x 20	\$ 0

Frutas ▼

Tomate de árbol malla x 20

REGISTRAR PRODUCTO
CANCELAR

Figura 16: forma de registro de producto completa

Tomate de árbol malla x 20	Tomate de árbol malla x 20	Frutas	\$ 2.65	\$ 0				
----------------------------	----------------------------	--------	---------	------	--	--	--	--

Figura 17: producto agregado

3.4.7 Sprint Retrospective

Después de aplicar la técnica de estrella de mar se encontró que se debe disminuir el tiempo que se entrega a tareas que causan demoras como, por ejemplo: revisar mucho tiempo el celular. También se debe seguir incrementando el interés y la interacción con el Product Owner. Se debe seguir con los procesos de documentación y pruebas unitarias. Se debe empezar a tener más opciones para ser aplicadas en casos de eventos inesperados, como el caso de enfermedad o de problemas personales de los miembros del equipo de desarrollo que puedan demorar el proyecto.

3.5 Segundo Sprint

3.5.1 Sprint Planning

Aplicando la técnica Planning Poker se pudo estimar y seleccionar las historias de usuario que van a ser desarrolladas en el segundo Sprint, considerando las

opiniones de todo el equipo, así como también permitir que los integrantes tengan más compromiso con el proyecto, para este primer Sprint se seleccionaron las historias de usuario 6, 7, 8, 9, 10, 11.

3.5.2 Sprint Backlog

En el Sprint Backlog del segundo Sprint se encuentran listadas las historias de usuario con identificador 6, 7, 8, 9, 10, 11 y que están descritas en la tabla 1 ya que estas se encuentran agrupadas por actividades que permiten generar un incremento

A continuación, se muestra la tabla correspondiente al Sprint Backlog del Segundo Sprint.

Dentro de la tabla se tiene las siguientes columnas: id, descripción, esfuerzo, y prioridad.

- La columna id indica el número de historia de usuario, la columna nombre muestra el nombre de la historia de usuario
- La columna descripción contiene la información de la historia de usuario desarrollada.
- La columna esfuerzo indica el esfuerzo real que fue necesario para completar el desarrollo de esa historia de usuario.
- La columna prioridad indica que tan crítico es desarrollar esa historia de usuario.

Tabla 3.

Sprint Backlog del segundo Sprint

ID	Nombre	Descripción	Estado	Esfuerzo	Prioridad	Responsable
6	Pedido	Como administrador se debe poder recibir un correo con la información del cliente, la lista de los productos reservados y el	Completo	6	Alta	Hernán Verdezoto

		valor a pagar, para poder separar su pedido				
7	Ubicaciones	Como usuario se debe poder revisar la ubicación de los puntos de venta, así como también ver un mapa para poder llegar al punto de venta	Completo	3	Baja	Hernán Verdezoto
8	Información del producto	Como administrador se debe ingresar imágenes de un producto y poder resaltar una imagen como la más importante del mismo, para resaltar el producto	Completo	5	Alta	Hernán Verdezoto
9	Dudas sobre los productos	Como usuario se debe ser capaz de comunicarse con el administrador del sistema a través de un correo electrónico para solventar sus dudas o necesidades	Completo	4	Baja	Hernán Verdezoto
10	Información	Como usuario se debe poder ingresar a una página informativa para conocer más sobre la granja	Completo	4	Baja	Hernán Verdezoto
11	Histórico	Como administrador se debe poder revisar un histórico de los productos reservados para poder analizar mejor que productos son los más reservados	Completo	3	Media	Hernán Verdezoto

3.5.3 Daily Scrum

Cada día el equipo se reunió durante 15 minutos con el fin de responder a las 3 inquietudes, ¿Qué se hizo ayer?, ¿Qué se va a hacer hoy? y ¿Qué problemas se presentaron? Una vez respondidas las 3 interrogantes se procede a actualizar la gráfica Burn Down de Sprint, para poder observar el comportamiento y avance logrado.

En el segundo Sprint se presentaron inconvenientes en el avance durante los primeros días, debido a esto el Scrum Master tomo acciones para tratar de estabilizar el proceso, pero debido a la mala estimación se presentaron demoras.

3.5.4 Sprint Review

Al finalizar este Sprint se llevó a cabo un Sprint Review, en donde revisó lo que se trabajó y el incremento presentado que aporta valor para el cliente, además se obtienen conclusiones, recomendaciones y experiencias que fueron compartidos con el equipo.

Al finalizar el segundo Sprint se obtuvo el proceso de negocio de la granja Nono.

3.5.5 Código segundo Sprint

```
namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Category;

class TestController extends Controller
{

    public function welcome()
    {

        $categories = Category::has('products')->get();
        return view('welcome')->with(compact('categories'));

    }

}
```

Figura 18: código fuente controlador “bienvenida”

```

2 namespace App\Http\Controllers;
3 use Illuminate\Http\Request;
4 use Illuminate\Support\Facades\DB;
5 use App\Http\Requests;
6 use App\Http\Controllers\Controller;
7
8 class LocationController extends Controller {
9
10 public function index(){
11
12
13 return view('locations.show');
14 }
15
16 }

```

Figura 19: código fuente controlador “ubicaciones”

```

ImageController.php
1 <?php
2
3 namespace App\Http\Controllers\Admin;
4 use App\Http\Controllers\Controller;
5 use Illuminate\Http\Request;
6 use App\Product;
7 use App\ProductImage;
8 use File;
9
10 class ImageController extends Controller
11 {
12     public function index($id)
13     {
14         $product = Product::find($id);
15         $images = $product->images()->orderBy('featured', 'desc')->get();
16         return view('admin.products.images.index')->with(compact('product', 'images'));
17     }
18
19     public function store(Request $request, $id)
20     {
21
22         $file = $request->file('photo');
23         $path = public_path() . '/images/products';
24         $fileName = uniqid() . $file->getClientOriginalName();
25         $moved = $file->move($path, $fileName);
26
27         if($moved)
28         {
29             $productImage = new ProductImage();
30             $productImage->image = $fileName;

```

Figura 20: código fuente controlador imágenes “productos” parte 1

```

ImageController.php
31     $productImage->product_id = $id;
32     $productImage->save();
33 }
34
35     return back();
36 }
37 public function destroy(Request $request, $id)
38 {
39     $productImage = ProductImage::find($request->image_id);
40     if(substr($productImage->image, 0, 4) === "http")
41     {
42         $deleted = true;
43     }else{
44         $fullPath = public_path() . '/images/products' . $productImage->image;
45         $deleted = File::delete($fullPath);
46     }
47     if($deleted)
48     {
49         $productImage->delete();
50     }
51
52     return back();
53
54 }
55 public function select($id, $image)
56 {
57     ProductImage::where('product_id', $id)->update([
58         'featured' => false
59     ]);

```

Figura 21: código fuente controlador imágenes “productos” parte 2

```

61     $productImage = ProductImage::find($image);
62     $productImage->featured = true;
63     $productImage->save();
64
65     return back();
66 }
67
68 }

```

Figura 22: código fuente controlador imágenes “productos” parte 3

3.5.6 Capturas de pantalla del incremento del segundo Sprint

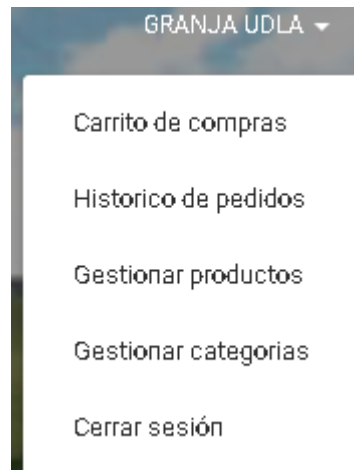


Figura 23: menú de la aplicación luego del segundo Sprint

Listado de Productos B

#	Nombre	Descripción	Categoría	Precio	Descuento	Acciones			
1	Huevos de gallina cubeta x 30	Huevos de gallina cubeta x 30	Huevos	\$ 3.75	\$ 0.05	i	📄	📷	✖
2	ECOFUNDAS UDLA	ECOFUNDAS UDLA	Generales	\$ 3	\$ 0	i	📄	📷	✖
4	Huevos de gallina cubeta x 15	Huevos de gallina cubeta x 15	Huevos	\$ 2.5	\$ 0	i	📄	📷	✖
5	Babaco Unidad	Babaco Unidad	Frutas	\$ 1.5	\$ 0	i	📄	📷	✖
6	Mora empacada 500g	Mora empacada 500g	Frutas	\$ 2.15	\$ 0	i	📄	📷	✖

Figura 24: listado de productos

Imágenes del producto "Huevos de gallina cubeta x 30"

No file chosen

SUBIR NUEVA IMAGEN

VOLVER A PRODUCTOS

Figura 25: agregar imágenes

Imágenes del producto "Huevos de gallina cubeta x 30"

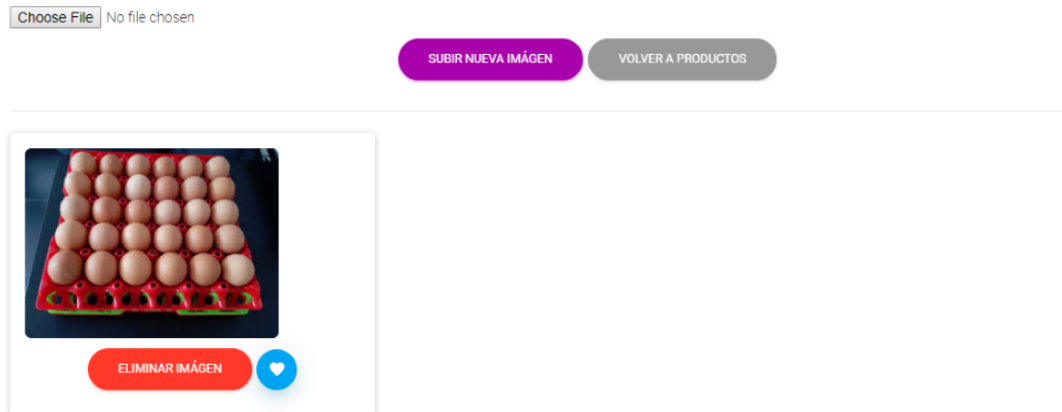


Figura 26: imágenes agregadas



Figura 27: información del producto luego de imagen agregada.

3.5.7 Sprint Retrospective

Después de aplicar la técnica de estrella de mar se encontró que se debe disminuir el tiempo de realización del trabajo diario. También se debe seguir con las interacciones con el Product Owner. Se debe continuar con los procesos de documentación y pruebas unitarias. Se debe dejar de lado los procesos lentos.

Y también se debe poder aplicar controles al desarrollo para que no se presenten demoras en el proceso de desarrollo

3.6 Medición y Estimación

Para poder llevar a cabo una evaluación (medición) del avance de esta solución con el uso del marco de trabajo Scrum se usan los gráficos Burn Up y Burn Down, debido a que estos gráficos permiten contrastar el avance real contra el estimado en el tiempo planteado. Estos gráficos muestran a detalle cómo se avanzó, y los tiempos planteados para entregar el sistema, en contraste con el alcance ideal que debía tener el equipo de trabajo.

3.7 Gráfico Burn Up

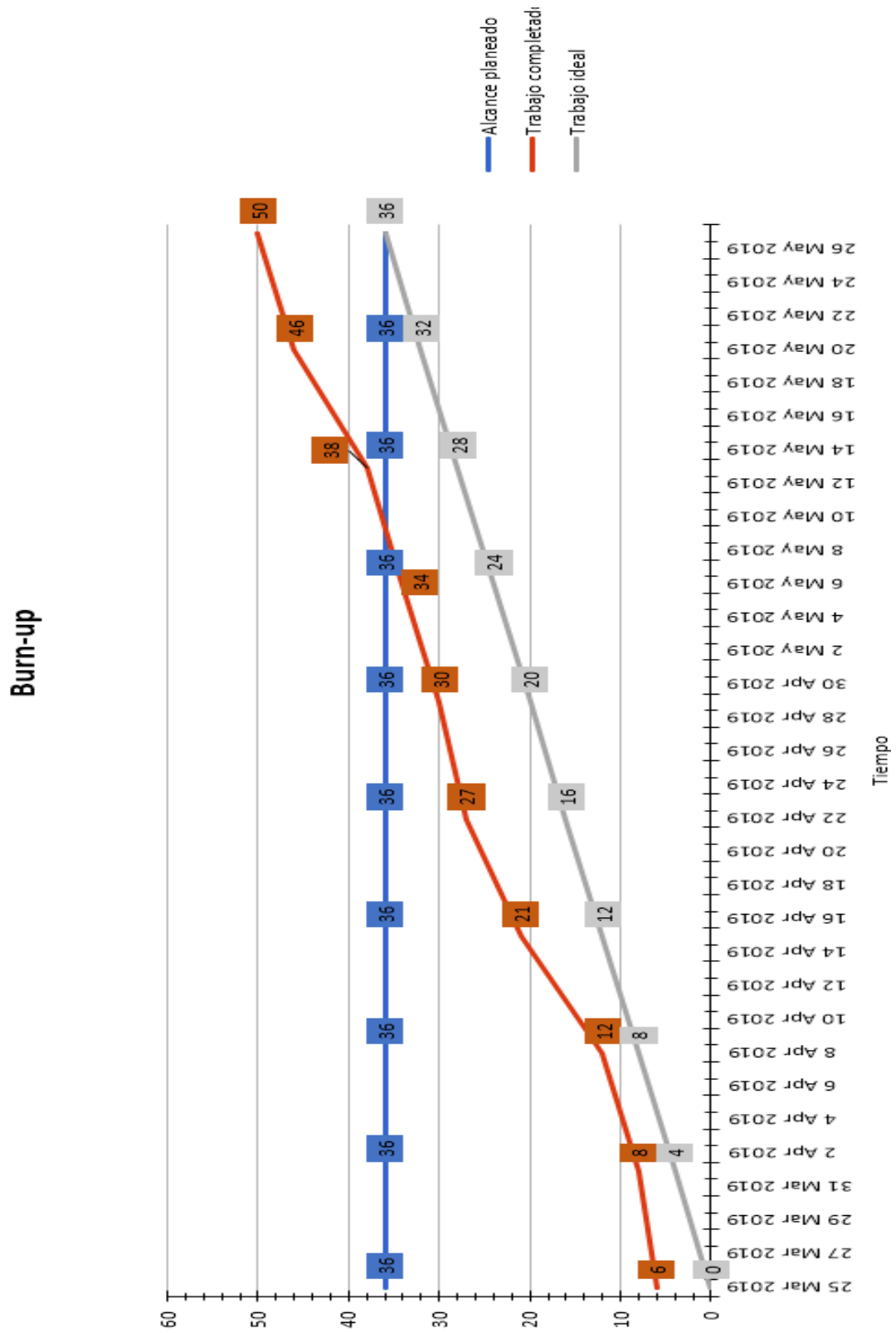


Figura 28: Gráfico Burn Up final

En el gráfico anterior se tiene 3 colores que representan el alcance planeado (azul), el trabajo completado (naranja), y el trabajo ideal (gris). El alcance planeado se mantiene constante durante todo el desarrollo ya que es la meta planteada, el trabajo ideal muestra el avance ideal planteado para cumplir con el desarrollo del sistema, y por último el trabajo completado muestra el esfuerzo real que fue necesario para poder cumplir con este proyecto. Como se puede apreciar en este gráfico la estimación planteada fue menor que la real. La estimación planteada fue de 38 puntos de historia, pero la real fue de 50 puntos de historia. El fallo en la estimación fue del 32%.

3.8 Gráficos Burn Down

3.8.1 Gráfico Burn Down Primer Sprint

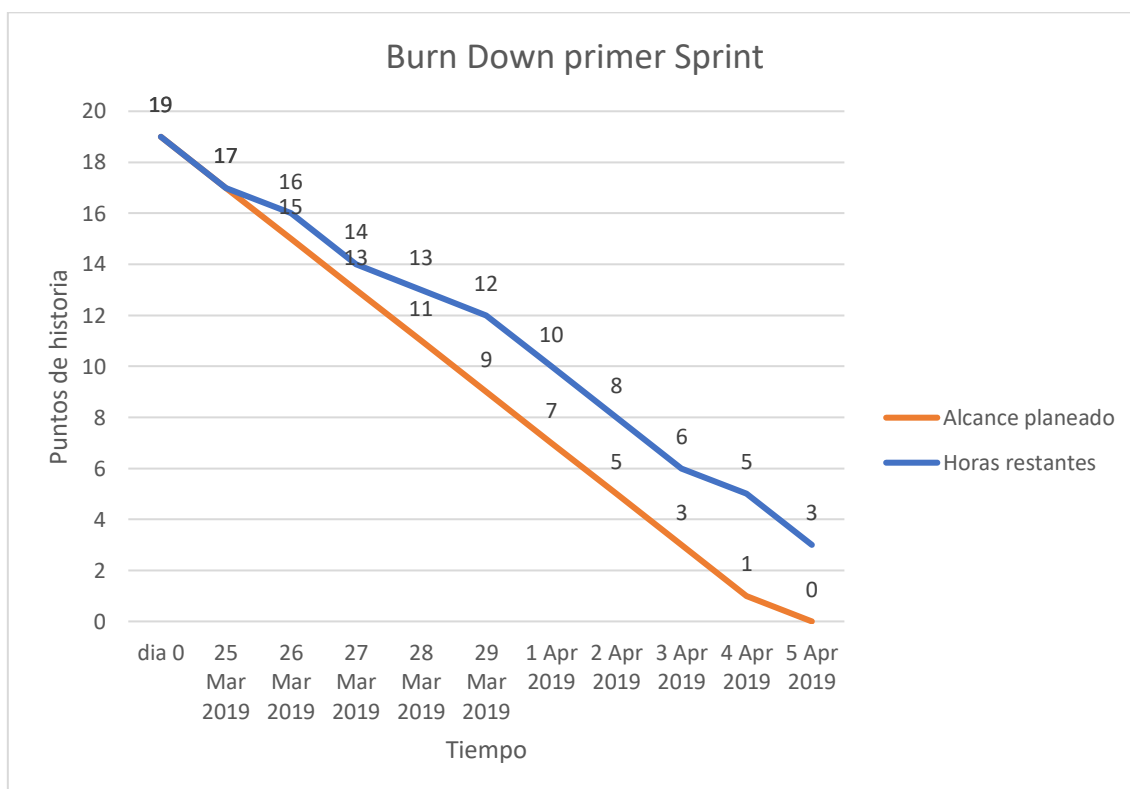


Figura 29: Burn Down primer Sprint

3.8.2 Gráfico Burn Down Segundo Sprint

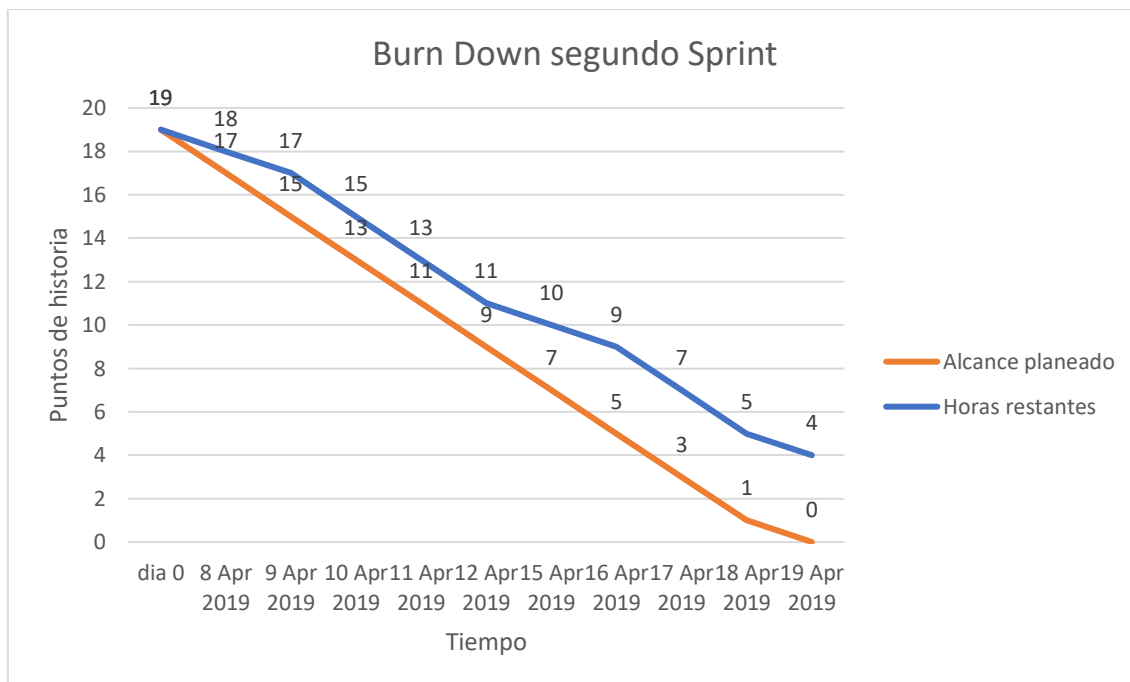


Figura 30: Burn Down segundo Sprint

4. Capítulo 4. Validación y Pruebas del Software

4.1 Introducción

En este capítulo, se expondrá las pruebas de validación de la solución planteada, para mejorar la comercialización de los productos de la granja Nono.

Las pruebas realizadas para validar esta solución fueron, unitarias y de aceptación del cliente, además de estas pruebas se ejecutaron pruebas de carga sobre el sistema usando la herramienta Jmeter. Los resultados de estas pruebas serán documentados y presentados en los siguientes epígrafes.

4.2 Pruebas de aceptación del cliente

Las pruebas de aceptación del cliente que fueron aplicadas están listadas en la siguiente tabla y representan a las historias de usuario presentadas anteriormente. La tabla de las pruebas tiene las siguientes columnas: un id, la descripción y un estado que puede ser "Fallido" o "Correcto".

Tabla 4

Pruebas de aceptación del cliente

id	Descripción	Estado
1	En la página de bienvenida hacer click en la lista de categorías y ver los productos de cada categoría	Correcto
2	En la página de bienvenida hacer click en el campo de campo de texto y escribir un nombre o parte de un nombre y hacer click en el botón de búsqueda y revisar los productos asociados a ese nombre o parte de nombre	Correcto
3	En la página de bienvenida hacer click en el hipervínculo “nuestras ubicaciones” y revisar la lista de sitios y horarios, también debe poder hacer click en el botón “mapa” y se debe desplegar el mapa con la ubicación del campus	Correcto
4	En la página de bienvenida hacer click en el hipervínculo “¿necesitas ayuda con un pedido?”, a continuación, ingresar el nombre, el mail, el mensaje y hacer click en el botón enviar.	Correcto
5	En la página de bienvenida hacer click en el hipervínculo “ingresar”, ingresar el correo y el password, y hacer click en el botón ingresar.	Correcto
6	Navegar hasta un producto y hacer click en el botón agregar al carrito, a continuación, indicar la cantidad de producto que se desea, y hacer click en agregar al carrito se debe presentar un mensaje que diga “producto agregado al carrito”.	Correcto
7	Navegar hasta el carrito de compra y revisar la lista de productos agregados al carrito, la cantidad del este, subtotal, descuento y la posibilidad de eliminar o revisar los detalles de este, y el valor a pagar.	Correcto

8	Como administrador navegar hasta el gestor de categorías y hacer click en el botón agregar categoría, ingresar el nombre de la categoría, seleccionar una imagen, y una descripción de la categoría y hacer click en el botón registrar categoría.	Correcto
9	Como administrador navegar hasta el gestor de productos y hacer click en el botón agregar producto, ingresar el nombre, el precio, una descripción corta, descuento, la categoría a la que pertenece y una descripción larga del producto y hacer click en el botón agregar producto.	Correcto
10	Como administrador navegar hasta el gestor de productos y hacer click en el botón imágenes, seleccionar una imagen, o varias imágenes y hacer click en el botón subir nueva imagen, y hacer click en la imagen a resaltar.	Correcto
11	Como administrador navegar hasta el histórico de pedidos y ver el histórico de pedidos.	Correcto

4.2.1 Demostración de las pruebas de aceptación del cliente

Prueba 1

la demostración de estas pruebas se encuentra en el anexo 1 de este documento

4.3 Pruebas unitarias

Las pruebas unitarias son pequeñas pruebas que son creadas con la única finalidad de comprobar que el código principal está funcionando correctamente, para evitar que el producto final presente errores o fallas en su programación, estas pruebas no toman como base los requerimientos de usuario. Estas pruebas permiten evitar que los clientes comenten errores dentro de la aplicación que se puedan presentar por no seguir correctamente los pasos establecidos y que puedan afectar al desempeño del código.

La siguiente tabla muestra la prueba realizada para verificar si un error está controlado o no (estado), por lo cual se muestra el identificador del error, una descripción de este error y el estado de ese error.

Tabla 5

Pruebas unitarias.

Id	Detalle	Estado
1	Validar que las credenciales ingresadas en la página de login sean las correctas	Controlado
2	Validar que los campos obligatorios estén completos antes de ingresar datos a la base de datos	Controlado
3	Validar el correcto ingreso de datos a la base de datos	Controlado
4	Validar la correcta visualización del menú correspondiente al rol asignado	Controlado
5	Validar si un email ya se encuentra en el sistema para evitar que se vuelva a registrar con ese mismo correo.	Controlado
6	Validar que las contraseñas ingresadas sean iguales	Controlado
7	Validar el tamaño mínimo de contraseñas	Controlado
8	Validar la correcta suma del valor total a pagar	Controlado

4.3.1 Demostración del cumplimiento de las pruebas unitarias

la demostración de estas pruebas se encuentra en el anexo 2 de este documento.

4.4 Pruebas de carga y estrés

Las pruebas de carga realizadas a la aplicación usando la herramienta Jmeter permitió identificar posibles cuellos de botella que afecten el rendimiento de la aplicación. Para esta prueba se generó un Thread Group con un total de 1000

usuarios con un periodo de solicitud de 1 segundo entre cada solicitud y con un loop de 5.

La ejecución de estas pruebas se encuentra en el anexo 3 de este documento

En estas pruebas se puede apreciar que el sistema al estar recibiendo altas cantidades de solicitudes el sistema empieza a presentar demoras en los tiempos de respuesta, más sin embargo es capaz de resolver todas las solicitudes sin presentar fallas, estas pruebas fueron planteadas para demostrar el comportamiento del sistema con una cantidad 10 veces mayor a la esperada. También podemos apreciar que el sistema tiene un rendimiento de 1335.063 respuestas de solicitudes por minuto, también se puede apreciar que el tiempo promedio de respuesta es casi constante.

5. Conclusiones y Recomendaciones

5.1 Conclusiones

Después de analizar los sistemas de compra y venta presentes en este documento, se puede apreciar que ninguno se acoplaba a las necesidades puntuales de la Granja, es por esto que se decidió desarrollar una aplicación desde cero, permitiendo de esta forma que la granja accediera a los beneficios del comercio electrónico, y que además de esto, permitiera mejorar la difusión de sus productos, poder estar disponible las 24 horas del día, generando una mayor fidelidad de sus clientes.

Scrum permite entregar parcialmente resultados funcionales y posiblemente desplegados, ya que se trata de generar un producto que aporte valor al finalizar cada SPRINT, para este proyecto se desarrollaron 2 SPRINT. Al finalizar el primer SPRINT se obtuvo el módulo de presentación de los productos, y al concluir el segundo SPRINT se desarrolló la lógica de negocio para la aplicación.

Laravel, es una herramienta de código abierto que cuenta con varios recursos como la consola Artisan, Eloquent, la creación de dependencias, la utilización del patrón MVC, la posibilidad de emplear PHP para el desarrollo, son elementos versátiles y útiles al momento de desarrollar software, además cuenta con una extensa documentación, la cual permite solucionar inquietudes o problemas de una manera fácil, rápida y eficaz. El uso de este framework facilitó el desarrollo de la solución, debido a las ventajas mencionadas anteriormente, al igual que su fácil uso y versatilidad, además de lo ya mencionado, se decidió usar un framework de código abierto, para que no existan inconvenientes con licencias.

La herramienta Jmeter permitió llevar a cabo pruebas de carga y estrés, una vez ejecutadas dichas pruebas, los resultados muestran que la aplicación presenta un incremento en los tiempos de respuesta de las solicitudes HTML; sin embargo, el sistema se mantiene estable frente a una gran cantidad de peticiones, ya que es capaz de resolver todas las solicitudes entregadas, además como se muestra en la figura 67 en color azul, el tiempo promedio de respuesta es casi constante para todas las solicitudes enviadas a través de la herramienta,

también, como muestra la figura 68, se puede apreciar un pico en el tiempo de respuesta del sistema, este pico de 17 segundos se debe a que el hosting gratuito donde se publicó el sistema realizó una optimización del servidor, posterior a esta optimización los tiempos de respuesta se estabilizaron, estas pruebas no solo mostraron los tiempos de respuesta de la solución sino que también demostraron que el sistema puede soportar hasta 10 veces más que el flujo transaccional que estaba considerado en un inicio, con lo cual se puede afirmar que el sistema es capaz de brindar alta transaccionalidad.

El Sprint Retrospective indicó que lo que más afectaba al avance y desarrollo del sistema era el tiempo que se empleaba para ciertas actividades distractoras, la falta de comunicación de los miembros del equipo de desarrollo, así como también la mala estimación del esfuerzo necesario para completar las historias de usuario.

Azure Devops es una herramienta de pago con la opción de ser gratuita con limitaciones, que permite llevar el control y el avance del proyecto, una de las características más importante de esta herramienta, es que está orientada al trabajo con Scrum, dentro de esta herramienta se puede definir el esfuerzo, el encargado de dicha actividad, así como también se puede controlar y monitorear el avance del proyecto, así como también aplicar acciones correctivas en el caso de ser necesarias para cumplir con los tiempos establecidos.

5.2 Recomendaciones

Se recomienda para una siguiente versión del sistema integrar una pasarela de pago, la cual debido a las políticas actuales de la granja y de la UDLA no fue posible realizar esta implementación. Se recomienda implementar Adyen, no solo porque grandes empresas como Vodafone, Uber o Netflix la usan, sino porque esta pasarela de pagos permite aceptar pagos de eCommerce, dispositivos móviles, o de múltiples opciones personalizadas, además de que reduce los costos ya que simplifica las operaciones, aparte es una plataforma robusta y segura.

Se recomienda que en un futuro se integre un módulo de distribución, el cual no fue implementado debido a la falta de capacidad de producción actual de productos.

Se recomienda que se pase el software al área de producción de la universidad, para este pueda contar con un dominio propio, se realice la integración con el servidor de correos de Office365, tanto para los correos de notificaciones tanto de pedidos o inquietudes recibidas y de respuesta a las mismas, y para que de esta forma, la universidad sea la encargada de administrar y gestionar el sistema

Referencias

- Álvarez, M.A (2001). Qué es PHP. Recuperado el 6 de abril de 2019 de:
<https://desarrolloweb.com/articulos/392.php>
- Apache (2019). *Apache Jmeter*. Recuperado el 12 de mayo de 2019 de:
<https://jmeter.apache.org/index.html>
- Booking (2019). Sobre Booking.com™. Recuperado el 7 de abril de 2019 de:
https://www.booking.com/content/about.es.html?aid=376374;label=esrowOtlvhU2CXhSVxek50Z_17wS267754636760%3Apl%3Aata%3Aap1%3Aap22%2C304%2C000%3Aac%3Aap1t1%3Aneg%3Afi%3Aati kwd-65526620%3Alp9069516%3Ali%3Adec%3Adm;sid=2bc278218b5e4fe2f5dccba48660040e
- But5ler, N. (2018). *Boost*. Recuperado el 3 de abril de 2019 de:
<https://www.boost.co.nz/blog/2018/05/stakeholders-in-Scrum-product-owner>
- CABRERA, F. M. (2000). El método inductivo, Monterrey, México. Recuperado el 12 enero de 2019 de:
<http://eprints.uanl.mx/5599/1/1080071376.PDF>
- Concepto.de. (2017). Método Inductivo. Recuperado el 12 de enero de 2019 de:
<https://concepto.de/metodo-inductivo/>
- conceptodefinicion.de. (2016). Método Inductivo. Recuperado el 12 de enero de 2019 de: <https://conceptodefinicion.de/metodo-inductivo/>
- Cruz, F. N. (2016). *gestiopolis*. Investigación Exploratoria. Recuperado el 12 de enero de 2019 de: <https://www.gestiopolis.com/la-investigacion-exploratoria/>
- Digitalelement (2014). Despegar.com ofrece servicios de planificación de viajes más personalizados en América Latina con la solución de geolocalización de Digital Element. Recuperado el 7 de abril de 2019 de: <https://www.digitalelement.com/translations/despegar-com->

ofrece-servicios-de-planificacion-de-viajes-mas-personalizados-en-america-latina-con-la-solucion-de-geolocalizacion-de-digital-element/

Drake, J.M y López, P. (2009) Computadores y Tiempo Real. Recuperado el 11 de abril de 2019 de: https://www.ctr.unican.es/asignaturas/Ingenieria_Software_4_F/Doc/M7_09_VerificacionValidacion-2011.pdf

EcuRed. Pruebas de caja blanca. Recuperado el 11 de abril de 2019 de: https://www.ecured.cu/Pruebas_de_caja_blanca

EcuRed. Pruebas de caja negra. Recuperado el 11 de abril de 2019 de: https://www.ecured.cu/Pruebas_de_caja_negra

Española, R. A. (2018). *enclave RAE*. Recuperado el 11 de enero de 2019 de: <http://dle.rae.es/srv/search?m=30&w=metodología>

Expedia (2019). Acerca de Expedia.mx. Recuperado el 7 de abril de 2019 de: <https://www.expedia.mx/g/rf/acerca-de-expedia>

García, J. M. (2015). ¿Qué es Laravel?

Glovo (2016). Preguntas frecuentes. Recuperado el 6 de abril de 2019 de: <https://glovoapp.com/es/faq>

Grupo Garatu Development. Grupo Garatu Development. Recuperado el 3 de abril de 2019 de: <https://development.grupogaratu.com/metodologia-Scrum-desarrollo-software/>

Laravel. (2018). *Laravel 5.7 From Scratch*. Recuperado el 13 de noviembre de 2018 de: <https://laracasts.com/series/laravel-from-scratch-2018>

Laravel. (2018). *Laravel Documentation*. Recuperado el 13 de noviembre de 2018 de: <https://laravel.com/docs/5.7/database>

Laravel. (2018). *What's New in Laravel 5.1*. Recuperado el 13 de noviembre de 2018 de: <https://laracasts.com/series/whats-new-in-laravel-5-1>

- Laravel. (2018). *Eloquent: Getting Started*. Recuperado el 15 de noviembre de 2018 de: <https://laravel.com/docs/5.7/eloquent#defining-models>
- Martín, B. (2018). ¿Qué tienen de malo los alimentos transgénicos? Recuperado el 21 de enero de 2019 de: https://elpais.com/elpais/2018/07/11/ciencia/1531299972_580652.html
- Mercado Libre (2018). ¿Qué es Mercado Libre? Recuperado el 7 de abril de 2019 de: <https://ideas.mercadolibre.com/ar/noticias/historia-de-mercado-libre/>
- Merino, J. P. (2017). Método Inductivo. Recuperado el 12 de enero de 2019 de: <https://definicion.de/metodo-inductivo/>
- Metodoss. (2015). Método Inductivo. Recuperado el 12 de enero de 2019 de: <https://metodoss.com/inductivo/>
- Microsoft Azure (2019). *Azure DevOps*. Recuperado el 29 de abril de 2019 de: <https://azure.microsoft.com/es-es/services/devops/>
- Ministerio de Telecomunicaciones y de la Sociedad de la Información. (s.f.). Comercio electrónico, una oportunidad para el desarrollo de negocios a través de la web. Recuperado el 7 de abril de 2019 de: <https://www.telecomunicaciones.gob.ec/comercio-electronico-una-oportunidad-para-el-desarrollo-de-negocios-a-traves-de-la-web/>
- Morales, N. (2017). Investigación Exploratoria. Recuperado el 11 de enero de 2019 de: <https://www.lifeder.com/investigacion-exploratoria/>
- MySQL (2018). *Reference Manual: 1.2.1 What is MySQL?* Recuperado el 9 de abril de 2019 de: <https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html>
- Najarro, M. (2017). *Planning Poker*. ¿Cómo realizar la estimación inicial de un proyecto de forma rápida y fiable? Recuperado el 4 de abril de 2019 de: <https://blog.interactius.com/planning-poker-c%C3%B3mo->

realizar-la-estimaci%C3%B3n-inicial-de-un-proyecto-de-forma-r%C3%A1pida-y-fiable-3576e9f1a943

OLX (2016). Acerca de OLX. Recuperado el 6 de abril de 2019 de:
<https://www.olx.com.ec/help/about>

Ortiz, A. (2018). Qué es un framework en informática o programación. Recuperado el 6 de abril de 2019 de:
<http://blog.hostdime.com.co/que-es-un-framework-informatica-programacion/>

Palacios, J. (2017). Guía fundamental de Scrum. Recuperado el 14 de enero de 2019 de: <https://jeronimopalacios.com/Scrum/>

PHP (2018). ¿Qué es PHP? Recuperado el 6 de abril de 2019 de 2019 de:
<http://php.net/manuales/es/intro-what-is.php>

S.A., B. B. (2019). Las estrategias de expansión y diversificación en la empresa. Recuperado el 14 de enero de 2019 de:
<https://www.bbva.es/general/finanzas-vistazo/empresas/expansion-diversificacion/index.jsp>

Sánchez, A. (2016). DESARROLLO DE UNA HERRAMIENTA INFORMÁTICA MODULAR BASADA EN PLATAFORMA WEB PARA LA OPTIMIZACIÓN DEL PROCESO. Recuperado el 14 de enero de 2019 de:
https://www.researchgate.net/publication/292962243_DESARROLLO_DE_UNA_HERRAMIENTA_INFORMATICA_MODULAR_BASADA_EN_PLATAFORMA_WEB_PARA_LA_OPTIMIZACION_DEL_PROCESO

Schawaber, k., & Sutherland, J. (2013). La Guía de Scrum. Recuperado el 3 de abril de 2019 de:
<https://www.Scrumguides.org/docs/Scrumguide/v1/Scrum-guide-es.pdf>

- ScrumManager.(2015). *ScrumManager*. Recuperado el 3 de abril de 2019 de:
https://www.Scrummanager.net/files/sm_proyecto.pdf
- Sierra, F., Acosta, J., Ariza, j., & Salas, M. (2017). Estudio y análisis de los framework en php basados en modelo vista controlador para el desarrollo de: software orientado a la web. *Revista Investigación y Desarrollo en TIC*, 4(2).
- SNIPPET, T. L. (n.d.). *Nobody Knows What the Hell They're Doing* [grabado por J. wade]. Recuperado el 17 de noviembre de 2018 de:
<https://laracasts.com/podcast/20273>
- Styde (2014). Qué es Composer y cómo usarlo. Recuperado el 7 de abril de 2019 de: <https://styde.net/que-es-composer-y-como-usarlo/>
- Styde (2015). Aprende a usar Eloquent el ORM de Laravel. Recuperado el 7 de abril de 2019 de: <https://styde.net/aprende-a-usar-eloquent-el-orm-de-laravel/>
- Symfony. Symfony para programadores. Recuperado el 8 de abril de 2019 de:
<https://symfony.es/pagina/que-es-symfony/>
- Testeando Software (2013). Recuperado el 11 de abril de 2019 de:
<https://testeandosoftware.com/casos-de-uso-vs-casos-de-prueba/>
- UDLA. (2013). GRANJA INTEGRAL Y SU BIODIVERSIDAD. Recuperado el 13 de noviembre de 2018 de:
<https://www.udla.edu.ec/internacional/cursos-internacionalidad/agricultural-management-and-biodiversity/granja-integral-y-su-biodiversidad/>
- UDLA. (2018). HISTORIA. Recuperado el 21 de enero de 2019 de:
<https://www.udla.edu.ec/la-udla/historia/>
- UDLA. (2018). PRODUCTOS DE LA GRANJA UDLA SE VENDERÁN EN LOS CAMPUS. Recuperado el 21 de enero de 2019 de:
<https://www.udla.edu.ec/2016/08/23/productos-de-la-granja-udla-se-venderan-en-los-campus/>

Veliz. B. G. (2009). Retrospectiva de estrella de mar (Starfish retrospective).
Recuperado el 4 de abril de 2019 de:
<https://proyectosagiles.org/2009/06/14/retrospectiva-estrella-mar-starfish-retrospective-Scrum/>

ANEXOS

Anexo 1 Demostración de las pruebas de aceptación

Visita nuestras categorías



Figura 31: vista de categorías

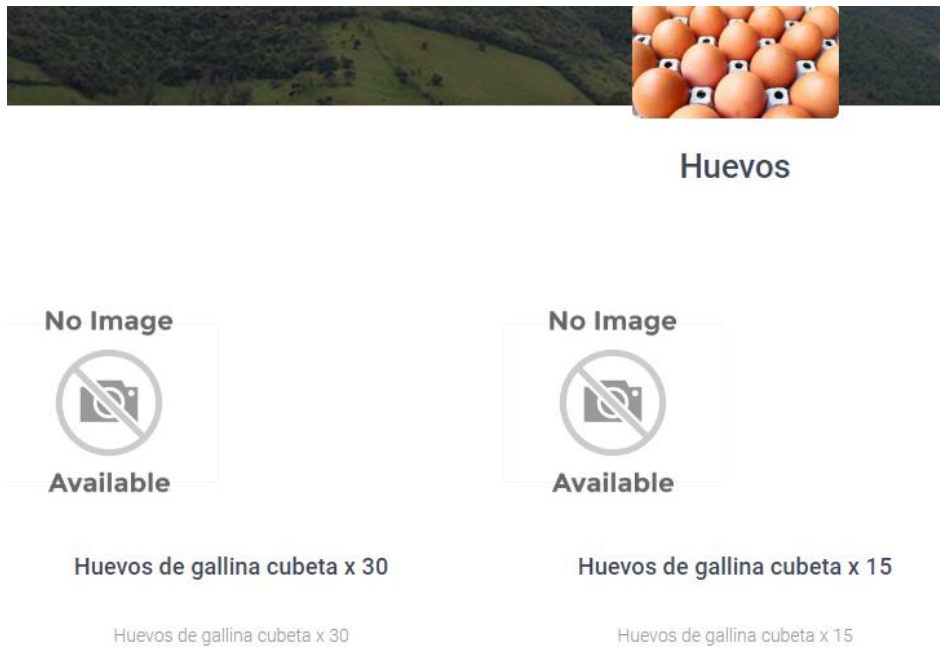


Figura 32: vista de productos por categoría

Prueba 2

Visita nuestras categorías

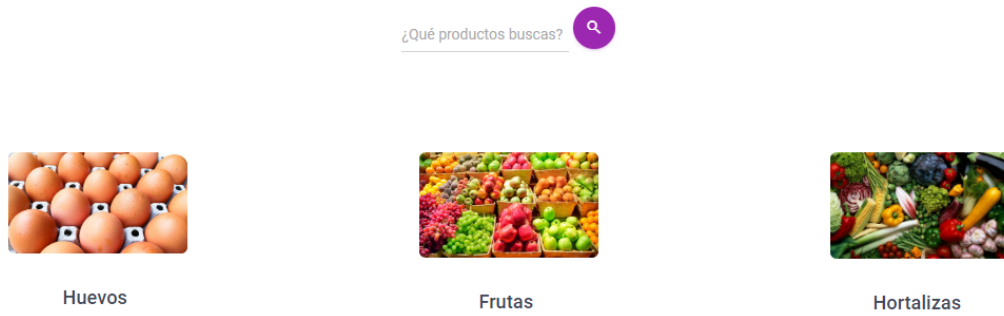


Figura 33: vista de categorías

Prueba 3

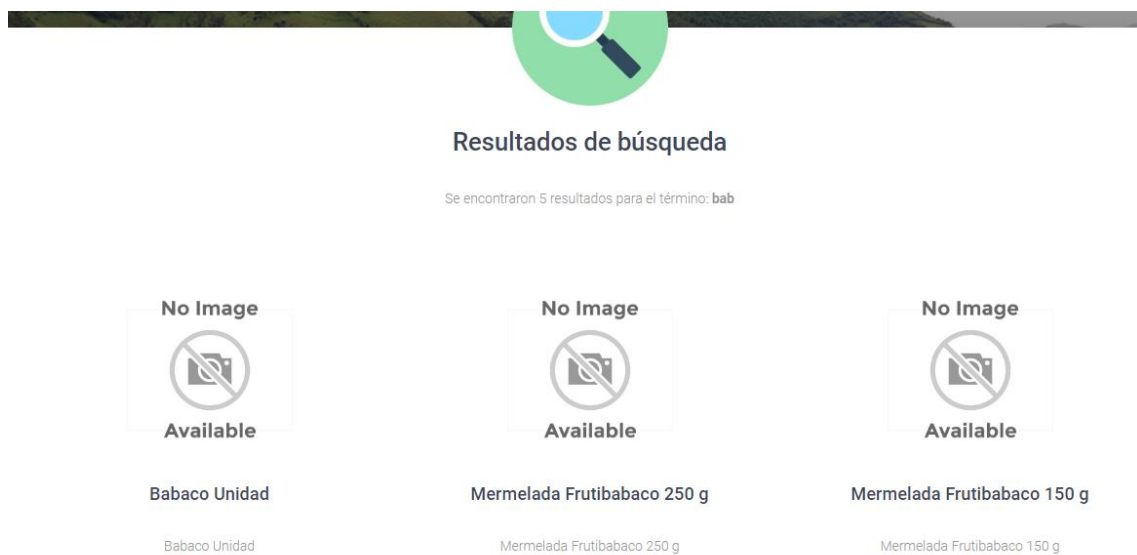


Figura 34: resultado búsqueda

Ubicaciones

Te esperamos los lunes, martes y miércoles de 11:00 a 15:00 en los siguientes puntos de atención:

Nombre	Horario de atención	Dirección
Querí	Lunes, entrada principal, bloque 1	MAPA
UdlaPark	Martes, en el Patio Amazónico	MAPA
Granados	Miércoles, junto a La Corteza	MAPA

Y los días jueves de 10:30 a 14:30 en el siguiente punto de atención:

Figura 35: vista ubicaciones

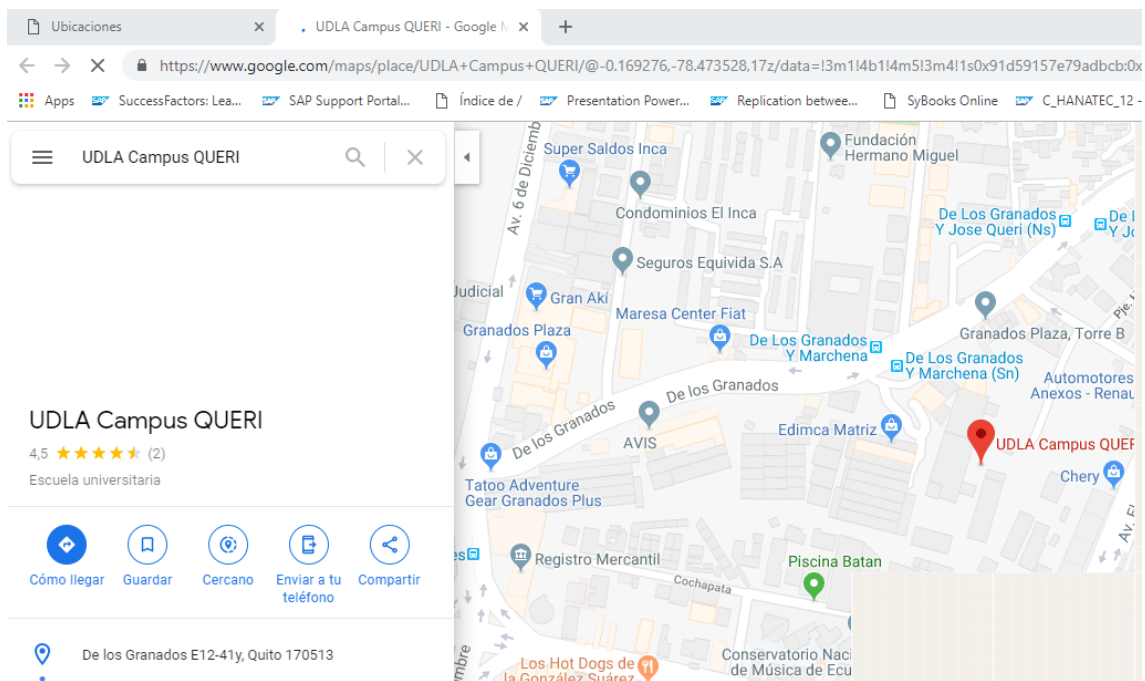


Figura 36: vista mapa

Prueba 4

Contacto

Nombre Email

Inserta tu nombre Inserta un email válido

Mensaje

email de prueba

Inserta tu mensaje

✓ ENVIAR REGRESAR A LA PAGINA PRINCIPAL

Figura 37: forma de contacto

Contacto

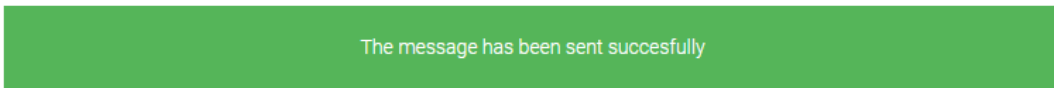


Figura 38: confirmación de envió mail



New message from Hernan Verdezoto

email de prueba

Laravel

Figura 39: recepción de mail

Prueba 5

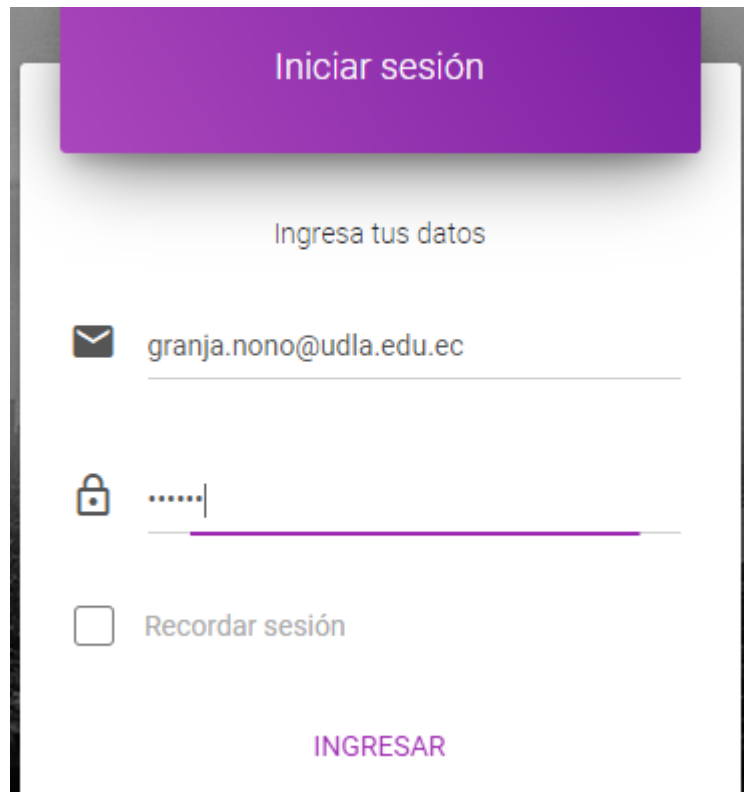


Figura 40: vista login

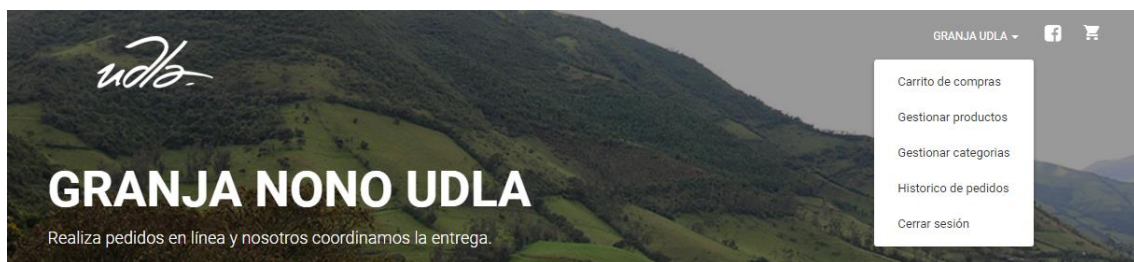


Figura 41: vista header de administrador luego de autenticación

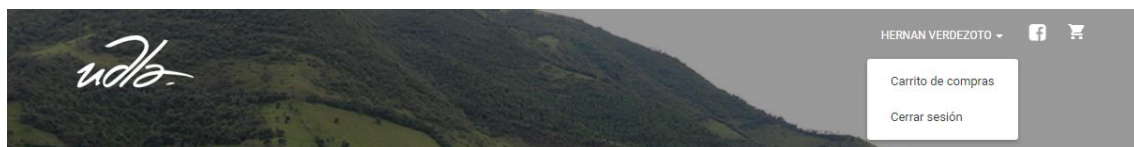


Figura 42: vista header de usuario luego de autenticación

Prueba 6

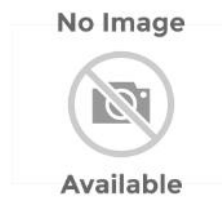


Huevos



Huevos de gallina cubeta x 30

Huevos de gallina cubeta x 30



Huevos de gallina cubeta x 15

Huevos de gallina cubeta x 15

Figura 43: vista de productos



Huevos de gallina cubeta x 30

HUEVOS

valor: 3.75 \$

+ AGREGAR AL CARRITO

IR AL CARRITO

← REGRESAR

Figura 44: vista de la información del producto

Selecciona la cantidad que deseas agregar ×

7| ⬆️ ⬆️

CANCELAR AGREGAR AL CARRITO

Figura 45: selección de cantidad del producto

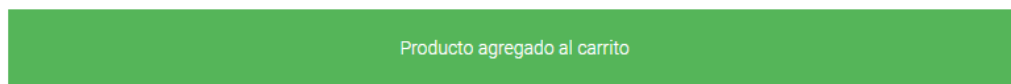


Figura 46: vista del mensaje “producto agregado al carrito”

Prueba 7

Carrito de Compras

CARRITO DE COMPRAS

SEGUIR AGREGANDO PRODUCTOS

Tu carrito de compras tiene 1 productos

#	Nombre	Precio	Cantidad	Subtotal	Descuento	Acciones
 Available	Huevos de gallina cubeta x 30	\$ 3.75	7	\$26.25	\$0.35	i ×

Total a pagar: 25.9


✓ REALIZAR PEDIDO

AGREGAR PRODUCTOS

Figura 47: vista del contenido del carrito

Prueba 8

Registrar nuevo categoría

 Manjar Imágen de la categoría
 No file chosen

Descripción de la categoría




Figura 48: formulario completo de nueva categoría












Yogurt			
Manjar			
Natillas			


Figura 49: vista de la nueva categoría

Prueba 9:

Registrar nuevo producto

 Tomate de árbol malla x 20 \$ 2.65

 Tomate de árbol malla x 20 \$ 0

Frutas 

Tomate de árbol malla x 20




Figura 50: forma de registro de producto completa.

Tomate de árbol malla x 20	Tomate de árbol malla x 20	Frutas	\$ 2.65	\$ 0				
----------------------------	----------------------------	--------	---------	------	--	--	--	--

Figura 51: producto agregado

Prueba 10

Listado de Productos

#	Nombre	Descripción	Categoría	Precio	Descuento	Acciones			
1	Huevos de gallina cubeta x 30	Huevos de gallina cubeta x 30	Huevos	\$ 3.75	\$ 0.05				
2	ECOFUNDAS UDLA	ECOFUNDAS UDLA	Generales	\$ 3	\$ 0				
4	Huevos de gallina cubeta x 15	Huevos de gallina cubeta x 15	Huevos	\$ 2.5	\$ 0				
5	Babaco Unidad	Babaco Unidad	Frutas	\$ 1.5	\$ 0				
6	Mora empacada 500g	Mora empacada 500g	Frutas	\$ 2.15	\$ 0				

Figura 52: listado de productos

Prueba 11:

Historico pedidos

Número de pedido	Email	User_id	telefono cliente	cantidad de productos pedidos	lista de productos
1	herniverde@gmail.com	2	0999815150	4	PULPA DE FRUTILLA
2	herniverde@gmail.com	2	0999815150	2	Huevos de gallina cubeta x 30
3	herniverde@gmail.com	2	0999815150	1	Mora empacada 500g
4	herniverde@gmail.com	2	0999815150	7	Huevos de gallina cubeta x 30

Figura 53: vista de productos vendidos.

Anexo 2 Demostración de las pruebas unitarias

Prueba 1



Figura 54: mensaje de error

Prueba 2

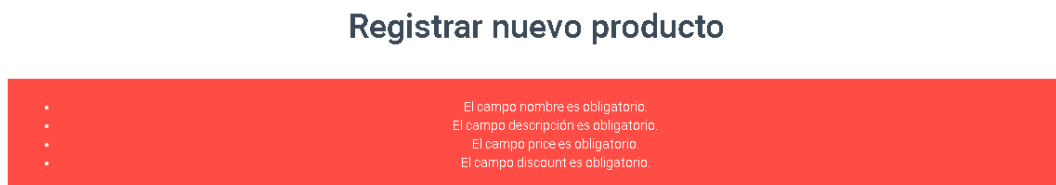


Figura 55: Mensaje de error

Prueba 3

id	name	description	long_description	price	discount
85	BROCOLI	BROCOLI	BROCOLI	0.50	0.00
86	Perejil	Perejil	Perejil	0.40	0.00
87	Perejil Crespo	Perejil Crespo	Perejil Crespo	0.50	0.50
88	Culantro	Culantro	Culantro	0.40	0.00
89	COLIFLOR	COLIFLOR	COLIFLOR	0.50	0.00
90	ACELGA	ACELGA	ACELGA	0.75	0.00
91	Alcachofa	Alcachofa	Alcachofa	0.45	0.00
92	ESPINACA	ESPINACA	ESPINACA	0.65	0.00
93	Zuchini Verde	Zuchini Verde	Zuchini Verde	1.15	0.00
94	Zuchini Amarillo	Zuchini Amarillo	Zuchini Amarillo	1.20	0.00
95	remolacha	remolacha	remolacha	0.65	0.00
96	VAINITA	VAINITA	VAINITA	1.00	0.00
97	VAINITA CHINA	VAINITA CHINA	VAINITA CHINA	1.00	0.00
98	COL	COL	COL	0.65	0.00
99	LECHUGA CRESPA	LECHUGA CRESPA	LECHUGA CRESPA	0.50	0.00
100	Prueba	Producto de prueba	prueba de ingreso a la base	0.00	0.00

cted:  Edit  Copy  Delete  Export

Figura 56: validación en la base

Prueba 4



Figura 57: vista del menú de administrador luego de autenticación con usuario administrador

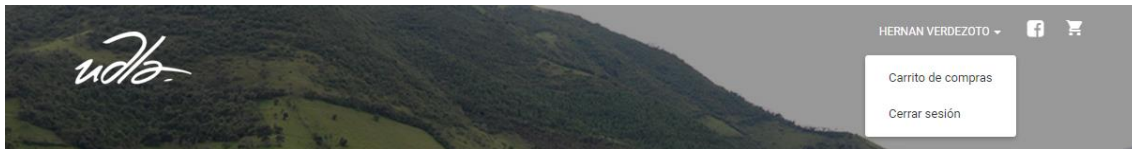


Figura 58: vista del menú de usuario luego de autenticación con usuario normal

Prueba 5



Figura 59: mensaje de error

Prueba 6

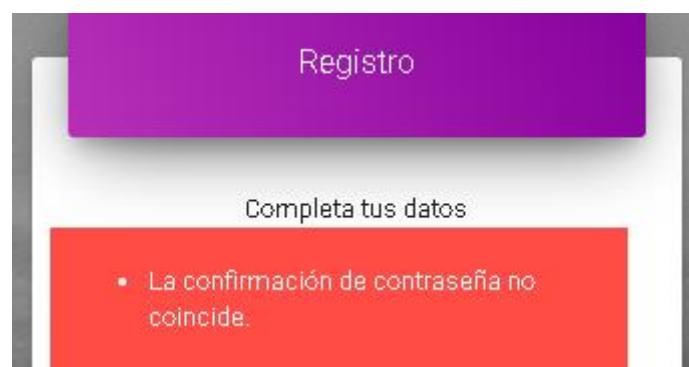


Figura 60: mensaje de error

Prueba 7

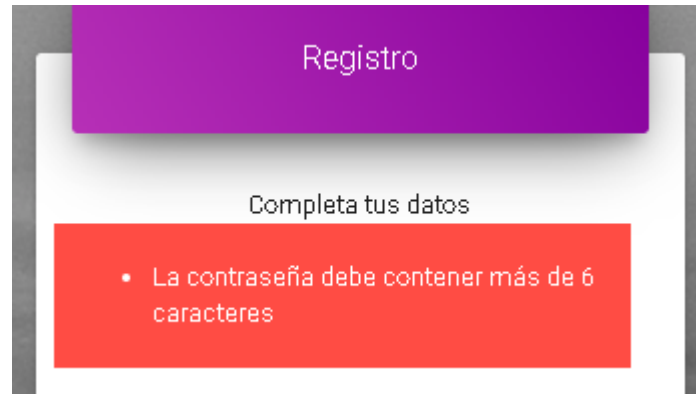



Figura 61: mensaje de error

Prueba 8

Carrito de Compras

CARRITO DE COMPRAS [SEGUIR AGREGANDO PRODUCTOS](#)

Tu carrito de compras tiene 3 productos

#	Nombre	Precio	Cantidad	Subtotal	Descuento	Acciones
	Huevos de gallina cubeta x 30	\$ 3.75	7	\$26.25	\$0.35	i x
	PULPA DE FRUTILLA	\$ 1.79	3	\$5.37	\$0	i x
	Pepinillo	\$ 0.5	2	\$1	\$0	i x

Total a pagar: 32.27

Figura 62: listado de productos ingresados en el carrito

Total a pagar: 32.27

Figura 63: valor total a pagar

Anexo 3 Ejecución de las pruebas de carga y estrés

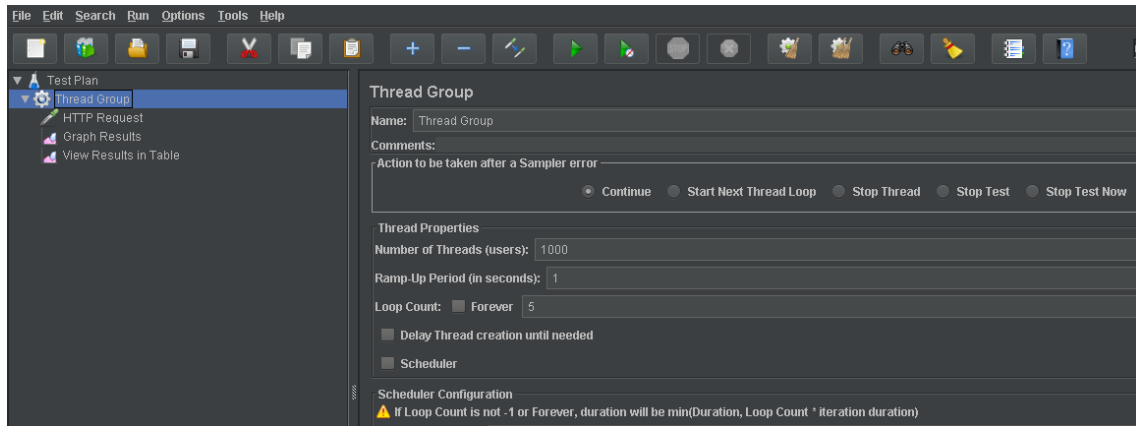


Figura 64: configuración del thread group para la prueba de carga

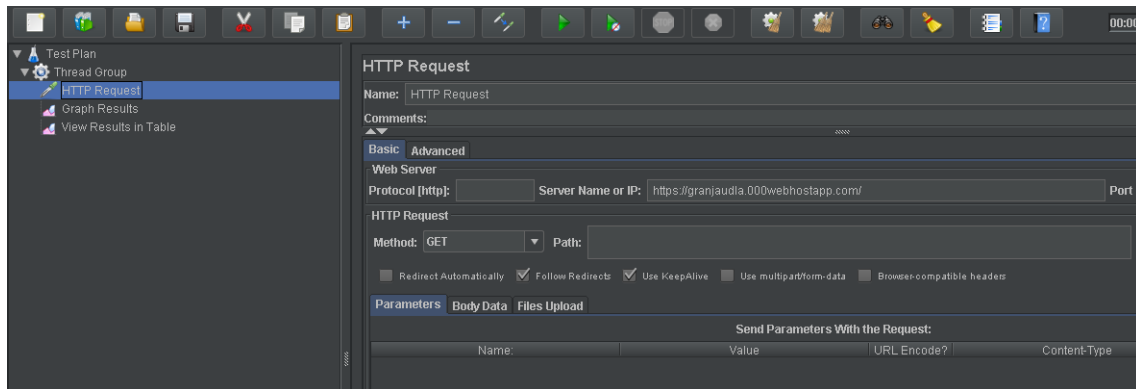


Figura 65: configuración del HTTP Request

Sample #	Start Time	Thread Name	Label	Sample Time(m...	Status	Bytes	Sent Bytes	Latency	Connect Time(...
9712	14:35:51.224	Thread Group 1...	HTTP Request	24739	🟢	20857	118	24739	1
9713	14:35:51.351	Thread Group 1...	HTTP Request	24757	🟢	20851	118	24757	0
9714	14:35:51.486	Thread Group 1...	HTTP Request	24755	🟢	20855	118	24755	1
9715	14:35:51.619	Thread Group 1...	HTTP Request	24763	🟢	20855	118	24763	1
9716	14:35:51.749	Thread Group 1...	HTTP Request	24785	🟢	20855	118	24785	1
9717	14:35:51.880	Thread Group 1...	HTTP Request	24792	🟢	20853	118	24792	1
9718	14:35:52.011	Thread Group 1...	HTTP Request	24829	🟢	20851	118	24829	1
9719	14:35:52.144	Thread Group 1...	HTTP Request	24848	🟢	20857	118	24848	1
9720	14:35:52.306	Thread Group 1...	HTTP Request	25013	🟢	20865	118	25013	1
9721	14:35:52.439	Thread Group 1...	HTTP Request	25008	🟢	20851	118	25008	1
9722	14:35:52.576	Thread Group 1...	HTTP Request	24994	🟢	20855	118	24994	1
9723	14:35:52.709	Thread Group 1...	HTTP Request	24984	🟢	20855	118	24984	1
9724	14:35:52.841	Thread Group 1...	HTTP Request	24975	🟢	20859	118	24975	1
9725	14:35:52.976	Thread Group 1...	HTTP Request	24966	🟢	20855	118	24966	1
9726	14:35:53.105	Thread Group 1...	HTTP Request	24960	🟢	20849	118	24960	1
9727	14:35:53.240	Thread Group 1...	HTTP Request	24962	🟢	20851	118	24962	1
9728	14:35:53.367	Thread Group 1...	HTTP Request	24956	🟢	20857	118	24956	1
9729	14:35:53.504	Thread Group 1...	HTTP Request	24944	🟢	20861	118	24944	1
9730	14:35:53.663	Thread Group 1...	HTTP Request	24915	🟢	20857	118	24915	1
9731	14:35:53.793	Thread Group 1...	HTTP Request	24908	🟢	20853	118	24908	1
9732	14:35:53.926	Thread Group 1...	HTTP Request	24896	🟢	20859	118	24896	1
9733	14:35:54.064	Thread Group 1...	HTTP Request	24886	🟢	20851	118	24886	1
9734	14:35:54.195	Thread Group 1...	HTTP Request	24882	🟢	20851	118	24882	0
9735	14:35:54.332	Thread Group 1...	HTTP Request	24869	🟢	20851	118	24869	1
9736	14:35:54.469	Thread Group 1...	HTTP Request	24854	🟢	20861	118	24854	1
9737	14:35:54.602	Thread Group 1...	HTTP Request	24846	🟢	20853	118	24846	1

Scroll automatically?
 Child samples?
No of Samples 10000
Latest Sample 23719
Average 5149
Deviation 8930

Figura 66: muestra del resultado de los HTTP Request realizados

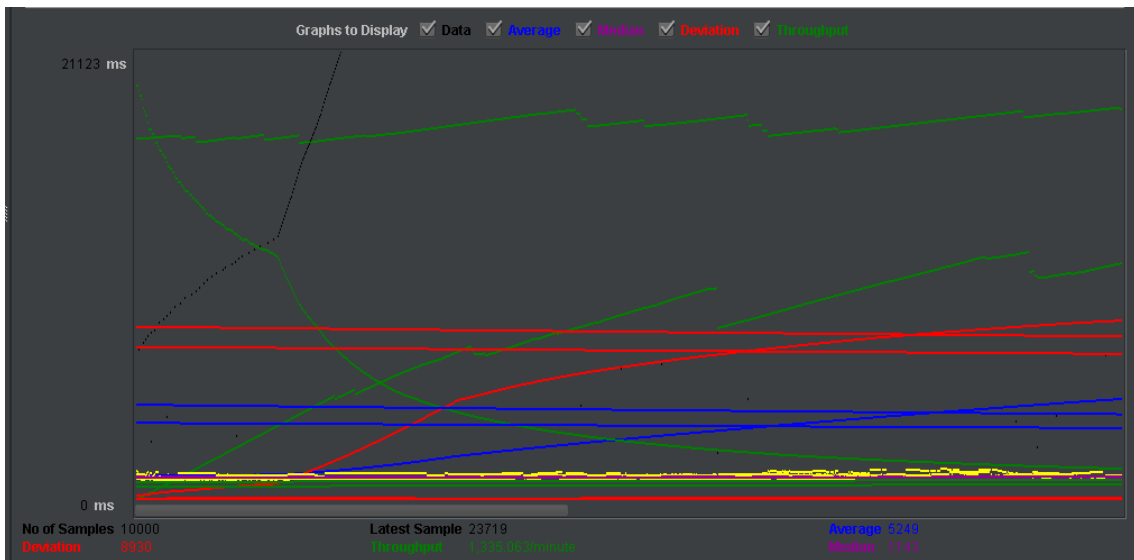


Figura 67: grafico de resultados de los HTTP Request

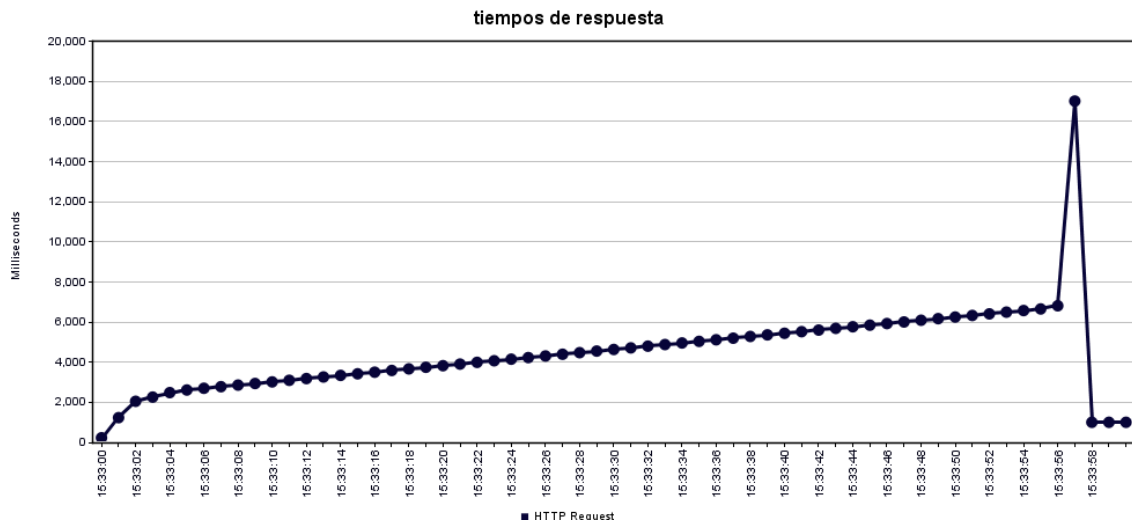


Figura 68: imagen de tiempos de respuesta.

