



FACULTAD DE INGENIERÍA Y CIENCIAS AGROPECUARIAS

DISEÑO DE UNA TIENDA EN LÍNEA DE PRODUCTOS BÁSICOS PARA
EL HOGAR UTILIZANDO GEO LOCALIZACIÓN A TRAVÉS DE UNA
APLICACIÓN MÓVIL



AUTOR

JORGE RAMIRO ALEGRÍA VELASCO

AÑO

2017



FACULTAD DE INGENIERÍA Y CIENCIAS AGROPECUARIAS

DISEÑO DE UNA TIENDA EN LÍNEA DE PRODUCTOS BÁSICOS PARA EL
HOGAR UTILIZANDO GEO LOCALIZACIÓN A TRAVÉS DE UNA
APLICACIÓN MÓVIL

Trabajo de Titulación presentado en conformidad con los requisitos
establecidos para optar por el título de Ingeniero en Electrónica y Redes de
Información

Profesor guía
MSc. Carlos Andrés Guaita Ayala

Autor
Jorge Ramiro Alegría Velasco

Año
2017

DECLARACIÓN DEL PROFESOR GUÍA

“Declaro haber dirigido este trabajo a través de reuniones periódicas con el estudiante, 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”.

Carlos Andrés Guaita Ayala

Máster Universitario en Ingeniería Biomédica

CC: 1715607071

DECLARACIÓN DEL PROFESOR CORRECTOR

“Declaro haber revisado este trabajo, dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación”

Carlos Andrés Muñoz Cueva

Magister en Gerencia de Sistemas

CC: 1712981511

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.”

Jorge Ramiro Alegría Velasco

CC: 0201931557

AGRADECIMIENTOS

Agradezco de manera muy especial a las personas ilustres J. Ramiro Alegría Calero y Silvia G. Velasco Zavala, mis padres, quienes me han apoyado incondicionalmente con cada consejo, aliento, conocimiento, y cariño para poder alcanzar otra meta. Esta meta no es mía, es de todos junto a mis hermanos que los guardo profundamente en mi corazón.

RESUMEN

El presente trabajo contiene el diseño y desarrollo de una aplicación móvil para teléfonos inteligentes con Sistema Operativo Android; la cual permitirá a través del uso de geo localización, realizar la compra de productos básicos para el hogar haciendo uso de un catálogo en línea. La utilización de este sistema de compra permitirá brindar al usuario un mayor confort debido a que no tendrá que salir de su casa para poder obtener sus artículos de consumo básico. Así también, ayudará a los pequeños y medianos centros de expendio (tiendas) a mejorar sus ventas al ofertar y vender sus productos de forma telemática, propiciando el uso de la tecnología a los clientes como a los propietarios de las tiendas.

Palabras Clave: Sistema Operativo, Android, geo localización, catálogo en línea, telemática.

ABSTRACT

The present work contains a mobile application design and development for smartphones with an Android Operative System, which will through the geo localization use, make the purchase of basic products for home using an online catalogue. The use of this system of purchase will allow to offer the user a better comfort because he will not have to leave his house to be able to obtain his articles of basic consumption. It will also help small and medium retail outlets to improve their sales by offering and selling their products in a telematics way, encouraging the use of technology to customers as well as store owners.

Keywords: Operative System, Android, geo localization, online catalogue, telematics.

ÍNDICE

1. Capítulo I. Introducción	1
1.1 Presentación del Trabajo	1
1.2 Organización del Trabajo	1
2. Capítulo II. Antecedentes	2
2.1 Encuesta	3
2.2 Tiendas en línea en Quito	4
2.2.1 Mercado Libre.....	4
2.2.2 OLX	5
2.2.3 Patio Tuerca	6
2.2.4 A Domicilio Ya	7
3. Capítulo III. Objetivos	8
3.1 Objetivo general	8
3.2 Objetivos Específicos	8
4. Capítulo IV. Métodos y Herramientas	8
4.1 Metodología Scrum	8
4.2 Casos de uso	10
4.3 Arquitectura Cliente-Servidor	11
4.3.1 Cliente	12
4.3.2 Servidor	12
4.4 JavaScript Object Notation – JSON	12
4.5 Modelo Vista Controlador	14
4.5.1 Modelo.....	15

4.5.2	Vista.....	15
4.5.3	Controlador.....	16
4.6	API.....	16
4.7	Android Studio.....	16
4.8	Php.....	17
4.9	Slim.....	17
4.10	Bootstrap.....	18
4.11	JavaScript.....	18
4.12	Google Maps API.....	18
4.13	Volley.....	19
4.14	MySQL.....	19
4.15	Plivo.....	19
4.16	Draw.IO.....	19
5.	Capítulo V. Desarrollo Scrum.....	20
5.1	Product Backlog.....	20
5.2	Sprint Backlog.....	21
5.2.1	Sprint 1.....	21
5.2.1.1	BNUIOS1-1 Diagrama de servidores.....	22
5.2.1.2	BNUIOS1-2 Diseño de la arquitectura.....	23
5.2.1.3	BNUIOS2 Contratación de servidor.....	24
5.2.1.4	BNUIOS3 Base de Datos.....	25
5.2.2	Sprint 2.....	26
5.2.2.1	BNUIOS4 Creación de Modelos.....	26

5.2.2.2	BNUIOS5 Controladores	27
5.2.2.3	BNUIOS6 Composer y Helper.....	27
5.2.2.4	BNUIOS7 API.....	28
5.2.3	Sprint 3	28
5.2.3.1	BNUIOS 8-1 Creación de JavaScripts.....	30
5.2.3.2	BNUIOS 8-2 Elaboración estructura HTML de inicio	31
5.2.3.3	BNUIOS 8-3 Pantalla para cambiar la contraseña	32
5.2.3.4	BNUIOS 8-4 Pantalla para el inicio de sesión	33
5.2.3.5	BNUIOS 8-5 Opción para cerrar sesión	33
5.2.3.6	BNUIOS 8-6 Pantalla donde se muestra el mapa	34
5.2.3.7	BNUIOS 8-7 Pantalla para crear tiendas.....	35
5.2.3.8	BNUIOS 8-8 Pantalla para administrar tiendas	36
5.2.3.9	BNUIOS 8-9 Pantalla de perfil de usuario	37
5.2.3.10	BNUIOS 8-10 Pantalla para crear productos	38
5.2.3.11	BNUIOS 8-11 Pantalla para administrar productos.....	39
5.2.3.12	BNUIOS 8-12 Pantalla para registrar usuarios.....	40
5.2.3.13	BNUIOS 8-13 Pantalla para reporte de mensajes de texto	41
5.2.3.14	BNUIOS 8-14 Pantalla de solicitudes de compra.....	42
5.2.3.15	BNUIOS 8-15 Pantalla para el detalle de compra	43
5.2.3.16	BNUIOS 8-16 Pantalla para el reporte de ventas.....	44
5.2.3.17	BNUIOS 8-17 Pantalla de compras realizadas	45
5.2.3.18	BNUIO 8-18 Pantalla para crear usuarios	46
5.2.3.19	BNUIO 8-19 Pantalla para editar usuarios	47
5.2.3.20	BNUIOS 8-20 Pantalla de olvido de contraseña	48
5.2.4	Sprint 5	49

5.2.4.1	BNUIOS 9-1 Flujo de Aplicación Móvil	50
5.2.4.2	BNUIOS 9-2 Pantalla de información acerca de la aplicación ...	51
5.2.4.3	BNUIO 9-3 Pantalla del historial de compras del usuario.....	52
5.2.4.4	BNUIOS 9-4 Pantalla para el inicio de sesión del usuario.....	53
5.2.4.5	BNUIOS 9-5 Pantalla principal	54
5.2.4.6	BNUIOS 9-6 Pantalla de tienda y sus productos.....	55
5.2.4.7	BNUIOS 9-7 Pantalla que permite registrar usuarios.	56
5.2.4.8	BNUIOS 9-8 Pantalla de elección de productos a comprar.....	57
5.2.4.9	BNUIOS 9-9 Pantalla de inicio o carga del sistema móvil	58
5.2.4.10	BNUIOS 9-10 Pantalla de la lista de los productos a elegir ...	58
5.2.4.11	BNUIO 9-11 Pantalla para recordar contraseña.....	59
6.	Conclusiones Recomendaciones	60
6.1	Conclusiones.....	60
6.2	Recomendaciones.....	61
	Referencias	62
	Anexos	65

ÍNDICE DE FIGURAS

Figura 1. Resultado de productos más vendidos a través de la encuesta	4
Figura 2. Logo de Mercado Libre	5
Figura 3. Logo de OLX	6
Figura 4. Logo de Patio Tuerca	6
Figura 5. Logo A Domicilio Ya	7
Figura 6. Metodología Scrum	9
Figura 7. Diseño de los casos de uso	10
Figura 8. Arquitectura Cliente-Servidor	11
Figura 9. Ejemplo de objeto JSON	14
Figura 10. Arquitectura y diseño del patrón MVC.....	15
Figura 11. Ejemplo de código Php	17
Figura 12. Ejemplo de código JavaScript	18
Figura 13. Diagrama de Servidores.....	22
Figura 14. Arquitectura del sistema.....	23
Figura 15. Modelo Base de Datos	25
Figura 16. Pantalla de inicio de la aplicación web.....	31
Figura 17. Pantalla web para cambiar la contraseña	32
Figura 18. Pantalla web para el inicio de sesión	33
Figura 19. Pantalla web donde muestra el localizador de tiendas.....	34
Figura 20. Pantalla web para creación de tiendas.....	35
Figura 21. Pantalla web para administrar tiendas	36
Figura 22. Pantalla web para ver el perfil del usuario.....	37
Figura 23. Pantalla web para crear productos.....	38
Figura 24. Pantalla web para administrar productos	39

Figura 25. Pantalla web para registro de usuarios	40
Figura 26. Pantalla web para el reporte de mensajes de texto	41
Figura 27. Pantalla web para las solicitudes de compra	42
Figura 28. Pantalla web para del detalle de compra	43
Figura 29. Pantalla web para el reporte de ventas	44
Figura 30. Pantalla web para ver el historial de compras realizadas.....	45
Figura 31. Pantalla web para crear usuarios	46
Figura 32. Pantalla web para administrar usuarios	47
Figura 33. Pantalla para resetear contraseña	48
Figura 34. Diagrama de flujo Aplicación Móvil.....	50
Figura 35. Pantalla móvil con información de la aplicación	51
Figura 36. Pantalla móvil para el historial de compras	52
Figura 37. Pantalla móvil para el inicio de sesión.....	53
Figura 38. Pantalla móvil principal.....	54
Figura 39. Pantalla móvil de la tienda con sus productos a escoger.....	55
Figura 40. Pantalla móvil para registrar usuarios	56
Figura 41. Pantalla móvil para seleccionar la cantidad de productos a elegir ..	57
Figura 42. Pantalla móvil de inicio.....	58
Figura 43. Pantalla móvil para resetear la contraseña	59

ÍNDICE DE TABLAS

Tabla 1. Product BackLog	20
Tabla 2. Sprint 1	21
Tabla 3. Sprint 2	26
Tabla 4. Modelos del Sistema	26
Tabla 5. Sprint 3	28
Tabla 6. Java Scripts del Sistema	30
Tabla 7. Sprint 4	49

1. Capítulo I. Introducción

1.1 Presentación del Trabajo

El presente Trabajo de Titulación, expone el diseño y desarrollo de un sistema de tipo web tanto como móvil para dispositivos Android, el mismo que permitirá la compra y venta de productos básicos para el hogar; dicho sistema se mostrará como una tienda online orientado a pequeñas y medianas empresas. La aplicación haciendo uso de la geo localización mostrará a las tiendas más cercas de la localización del usuario que use el dispositivo móvil. Así permitiéndole realizar la compra en cualquiera de las tiendas mencionadas, después de realizar la compra se enviará un mensaje de texto con los datos del cliente al igual que los productos comprados. Dichos productos tienen que tener un stock disponible, el cual se lo puede ver utilizando su sistema web.

Existen ciertas tiendas en línea en el país, donde ofrecen productos, pero en este caso el diseño mencionado va a ser la primera propuesta como ayuda a las personas que están en casa y por motivos exteriores tengan la necesidad de no acudir a una tienda a hacer la compra de productos básicos; sino, más bien, acudir a esta herramienta.

1.2 Organización del Trabajo

El presenta trabajo nace con una idea simple y sencilla; la cual es, proporcionar a un segmento de personas que por diferentes motivos, ya sean por: frio, incapacidad física, impedimentos laborales, impedimentos familiares, o simplemente por comodidad; no puedan visitar a una tienda cercana, puedan hacer su pedido de productos básicos para su hogar a través de una aplicación móvil.

Esta herramienta móvil por un lado se presentará como una tienda en línea, la cual gestionará datos a través de un servidor web, que, a su vez, las tiendas gestionarán sus productos a través de una página web.

Una vez conocidos los antecedentes como son la encuesta, y el levantamiento de información de las tiendas en línea existentes; las mismas que son plataformas con un sistema muy cercano a la propuesta, se ha procedido a plantear los objetivos del trabajo de grado.

Dentro del material y metodología se podrán observar las fases del trabajo, desde el diseño, desarrollo, estructuras. Haciendo énfasis a las conclusiones y recomendaciones de este diseño y desarrollo del sistema móvil y web se plantearán posibles trabajos futuros; los mismos que son derivados del mismo.

En el capítulo de referencias estará disponible el material principal utilizado para la realización de este trabajo.

2. Capítulo II. Antecedentes

En la familia han surgido siempre necesidades de ciertos artículos básicos en el hogar, los cuales en altas horas de la noche son difíciles de conseguir, ya sea porque está lloviendo, el clima no es el apropiado como para salir, o simplemente por comodidad de parte del cliente, al salir a una tienda y comprar el producto.

Para personas embarazadas, con enfermedades, o con niños en casa, es muy difícil salir de su hogar y comprar ciertos productos que se agotaron o debido a circunstancias ajenas, no fueron comprados con anterioridad.

En julio del 2010 fue presentado en la Facultad de Ciencias de la Comunicación de la Universidad Abierta Interamericana, en la ciudad autónoma de Buenos Aires, el trabajo especial de grado: "Marketing Móvil" por Cinthia Varas y revisada por el profesor Marcelo Savazzini.

Este estudio describe lo esencial e imprescindible que se ha vuelto un teléfono celular en la vida de las personas, siendo aparte de la billetera y llaves del hogar, un artículo necesario en el día a día. Debido a la importancia de dicho accesorio, el marketing puede aprovechar los diferentes mercados, marcas, etc. Ofreciendo el servicio de una tienda al cliente, ofertando promociones, beneficios, o simplemente la compra del producto en sí.

En el 2010 existían alrededor de 2,4 billones de teléfonos móviles usados en el mundo, hoy en día las cifras han quintuplicado su número. Añadiendo lo cómoda que se ha vuelto la gente actualmente, al igual que dependiente de soluciones de aspecto tecnológico. ¡Qué mejor cosa que hacerlo a través de aplicaciones móviles! (Varas, 2010)

De esta manera se pretende estimular las visitas o compras a tiendas, buen servicio y fidelidad de los clientes; y lo principal para los negocios, mejorar sus ingresos.

Debido a los problemas antes mencionados, el marketing móvil es una ruta muy afinada para llegar a este sector de personas que por ciertas razones no acceden a tiendas cercanas, y las aplicaciones móviles tienden a ser un gran mecanismo para su soporte.

2.1 Encuesta

Para empezar con esta plataforma se realizó una encuesta a un segmento de tiendas en el sector de nombre Jipijapa en la ciudad de Quito con la siguiente pregunta; ¿Le gustaría usar la internet para poder vender sus productos? Obteniendo una respuesta muy parecida entre los dueños de los respectivos locales comerciales; las cuales, daban como respuesta un sí rotundo al uso de la tecnología para la venta de sus productos.

Además, dentro de la encuesta a estas cinco tiendas con respecto a los productos que más venden son los que se muestran en el gráfico de la Figura 1 a continuación.

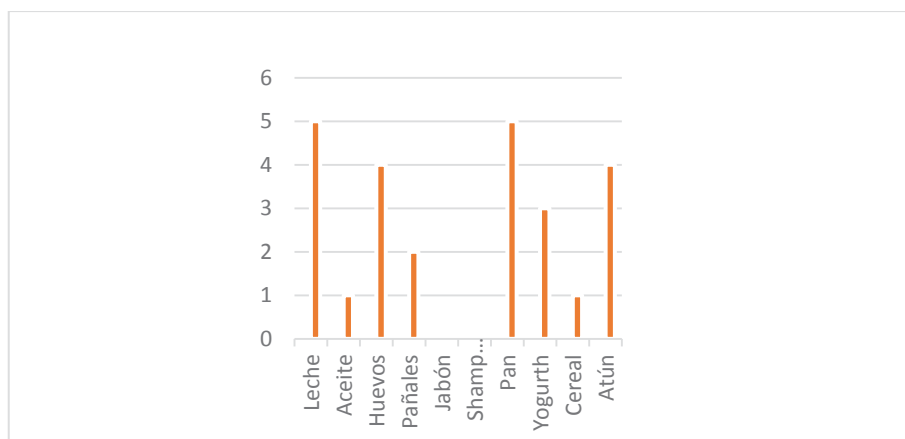


Figura 1. Resultado de productos más vendidos a través de la encuesta

Como consecuencia a la encuesta se puede ver que es factible usar la tecnología para hacer las compras de los productos básicos del hogar; además de tener una lista de los ítems que más se venden en este tipo de lugares comerciales.

2.2 Tiendas en línea en Quito

2.2.1 Mercado Libre

Mercado Libre es una tienda en línea, la cual es conocida y funciona a nivel internacional. Esta tienda, a través de un usuario vende y compra productos; hoy en día esta tienda tiene una aplicación móvil la cual permite un mejor acceso a las personas a utilizar este medio.

Hoy, esta plataforma es uno de los medios más usados digitalmente para generar transacciones por compras de manera rápida y segura. Según la tesis doctoral “Proyecto para la mejora de la comunicación y relaciones internas en una empresa de comercio electrónico”, afirma el beneficio de la plataforma a la sociedad: “ Mercado Libre, compañía que surgió de un proyecto universitario, es hoy un generador y multiplicador de empleos: más de mil 295 personas trabajan en la compañía en forma directa en toda América Latina” (Betancourt Granillo & Sánchez Anguiano, 2010). El logo de Mercado Libre se lo muestra en la Figura 2.



Figura 2. Logo de Mercado Libre
Tomado de (Mercado Libre, s.f.)

2.2.2 OLX

OLX es una empresa que al igual que Mercado Libre, a través de un usuario puede hacer transacciones de compra y venta de artículos ya sean nuevos, o usados; permitiendo, un mejor comercio seguro e informal dentro de un país. Los artículos en venta son diversos, los cuales permiten una gran gama de oportunidad el momento de hacer una venta o comprar un producto.

Esta empresa ya cuenta con su aplicativo móvil permitiendo un mayor ingreso de usuarios a usar esta plataforma para compras en línea; siendo otra página más que está a la vanguardia de la tecnología en ventas. El logo de esta empresa se lo puede ver en la Figura 3.



Figura 3. Logo de OLX
Tomado de (OLX, s.f.)

2.2.3 Patio Tuerca

Patio Tuerca es una tienda en línea internacional, con su respectiva sucursal en Ecuador, con una oferta de producto en el área automotriz; emitiendo transacciones de compra y venta de: autos, motocicletas, tractores, remolques, motos de agua, yates, avionetas, entre otros.

La demanda de transacciones y el avance tecnológico han llevado a que Patio Tuerca no solo tenga su tienda en línea responsive; sino, además, tenga su respectiva aplicación móvil para ofrecer un mejor servicio a los usuarios. El logo de Patio Tuerca se lo muestra en la Figura 4.



Figura 4. Logo de Patio Tuerca
Tomado de (Patio Tuerca, s.f.)

2.2.4 A Domicilio Ya

Esta empresa internacional, con su tienda en Ecuador, permite elegir una variedad de restaurantes para elegir su plato deseado en línea. De esta manera facilitando al usuario al no tener que moverse de su casa, lugar de trabajo, lugar de reunión, etc. Para poder hacer el pedido de su comida favorita.

A Domicilio Ya, cuenta con su aplicativo móvil que permite ya sea desde la web o desde el teléfono celular elegir las tiendas más cercanas a su ubicación para garantizar la entrega de sus productos.

Además, cada transacción genera un mínimo recargo por entrega; esta empresa llega a ser un intermediario entre los restaurantes y el usuario. Haciendo la parte de cliente y usuario simultáneamente por transacción. Su logo se lo puede ver en la Figura 5.



Figura 5. Logo A Domicilio Ya
Tomado de (A Domicilio Ya, s.f.)

Después de haber visto las tiendas en línea que funcionan en el medio, se prueba la no existencia de una plataforma web y móvil como la planteada. Dando como resultado un único e innovador sistema para que los dueños de los mencionados start ups y PYMES puedan mejorar sus ingresos y satisfacer las necesidades del cliente a través de su aplicación brindada que hace uso de la tecnología.

3. Capítulo III. Objetivos

3.1 Objetivo general

Diseñar una aplicación móvil de compras en línea “e-commerce” con geo localización para productos básicos del hogar.

3.2 Objetivos Específicos

- Contratar un servidor, y dominio web para el servicio.
- Diseñar una de base de datos para: cliente, tienda, producto, mensaje, compra.
- Diseñar dicha aplicación móvil con las características adecuadas.
- Diseñar una arquitectura cliente – servidor entre el aplicativo y el componente web.
- Adjuntar la geo localización para elegir la tienda más cercana y obtener coordenadas.
- Establecer mediante JSON el intercambio de datos entre el servidor y la aplicación.
- Establecer un servicio de mensajería para enviar mensajes de texto a las tiendas cercanas por cada compra realizada por el cliente.
- Diseñar de página web para cargar productos por parte de las tiendas y mantener el stock actualizado.

4. Capítulo IV. Métodos y Herramientas

4.1 Metodología Scrum

En el diseño de la aplicación móvil, será empleada la metodología llamada “Scrum” o metodología ágil.

Scrum es un proceso en el cual se usan una serie de buenas prácticas para obtener el mejor resultado posible del mismo a través de trabajo colaborativo en equipo.

En Scrum se usan entregas parciales y regulares del producto final; las mismas que son priorizadas por el beneficio que aportan al receptor del mismo; siendo, por esta causa, la mejor vía para obtener resultados parciales donde los requisitos pueden ser cambiantes o poco definidos; donde es fundamental la competitividad, productividad, flexibilidad e innovación, (proyectosagiles.org)

El proceso para la metodología ágil Scrum, un proyecto se ejecuta en bloques temporales cortos y fijos con una iteración de aproximadamente dos semanas entre cada uno, los cuales pueden ser hasta de cuatro semanas a las cuales se les aplica retroalimentaciones y reflexiones.

Siendo esta, la mejor forma de tener incrementos parciales del producto final, el cual evidencia el esfuerzo constante al realizar el proyecto. Se puede ver el funcionamiento en la Figura 6.

El proceso parte de la lista de objetivos planteados; en esta lista el cliente prioriza a los objetivos balanceando las necesidades y quedan divididos en iteraciones y entregas. (proyectosagiles.org)

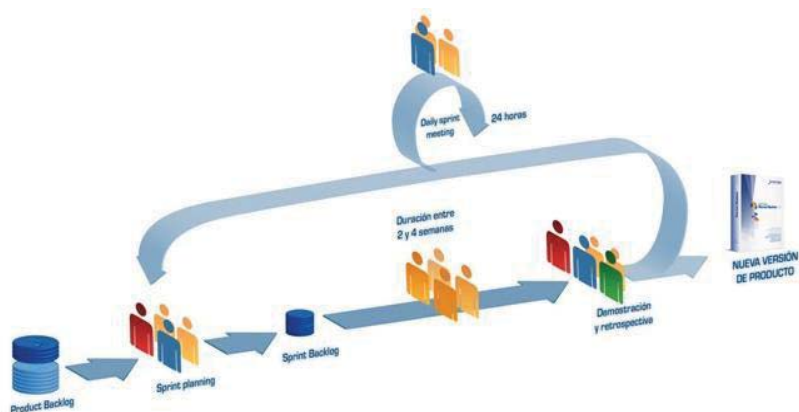


Figura 6. Metodología Scrum

Tomado de (Proyectos Ágiles, s.f.)

En el apartado de Resultados se hace el uso de esta metodología, la cual fue escogida por su rápido proceso para que a través de iteraciones semanales se va realizando los cambios necesarios como también el avance de la plataforma.

4.2 Casos de uso

Los casos de uso se implantan con un Id o identificación del requerimiento funcional con su respectiva descripción además de su pre condición y post condición. Esta tabla a llenar con los casos de uso tiene un flujo normal como alterno, excepciones y notas como se muestra en la Figura 7 a continuación. (Toapanta, 2012)

ID	RF-02
Descripción	Proporciona funcionalidades para dar mantenimiento a los equipos (PK) utilizados para la recolección de información.
Precondición	
Postcondición	Información actualizada de los equipos móviles disponibles para la gestión de lecturas.
Flujo Normal	
	1 El usuario logeado utiliza el formulario para mantenimiento de equipos.
	2 Se ingresan los datos del nuevo equipo.
	3 El sistema almacena los datos proporcionados.
Flujos Alternos	
	* El equipo ya está registrado
	Si el equipo ya existe en el sistema y es necesario modificar la información actual, el sistema proporciona la funcionalidad para realizarlo.
Excepciones	
	1 Si se intenta registrar un equipo con Número de serie o Cód. ASISTECOM que ya se encuentren registrados, el sistema debe advertir la situación y evitar que se registre el PK.
Notas:	
	1 Se requiere los siguientes datos:
	1.1. Código (*): Este código es un identificador único propio con los que los equipos utilizados por ASISTECOM son actualmente identificados. Este código debe ser registrado en los equipos.
	1.2. Marca (*): Se seleccionará de un catalogo de marcas.
	1.3. Modelo
	1.4. Número celular
	1.5. IMEI: Se debe revisar si es posible obtener, en la Aplicación DEMO, no fue posible de utilizar en todos los modelos probados.
	1.6. Imagen: Debe existir la posibilidad de almacenar un registro gráfico del equipo.
	1.7 Número de serie.
	1.8 Fecha de registro.
	1.9 Chip servicio internet
	Los campos macados con (*) son obligatorios.

Figura 7. Diseño de los casos de uso

Tomado de (Scrum, 2012)

Los mismos casos de uso con los que se trabajaron previos a realizar el diseño y desarrollo del sistema móvil y web.

4.3 Arquitectura Cliente-Servidor

En el diseño de la plataforma móvil para compras en línea de nombre "BuyNowUIO", se deberá tomar en cuenta que la misma va a implementar la arquitectura Cliente-Servidor, debido a que la aplicación móvil va a consumir lo que el componente web expone; por esta razón, se explicará el funcionamiento de este tipo de arquitectura.

La arquitectura Cliente-Servidor es un modelo de aplicación de tipo distribuida entre múltiples procesadores en el que las tareas se reparten entre los proveedores de recursos o servicios, de nombre servidores; y los demandantes, llamados clientes, donde estos serán los encargados de realizar peticiones y los servidores responden a las mismas. Habitualmente, diferentes programas de tipo Cliente, interactúan con los servicios de un servidor, como se lo puede ver en la Figura 8. (Tanenbaum, 2008).

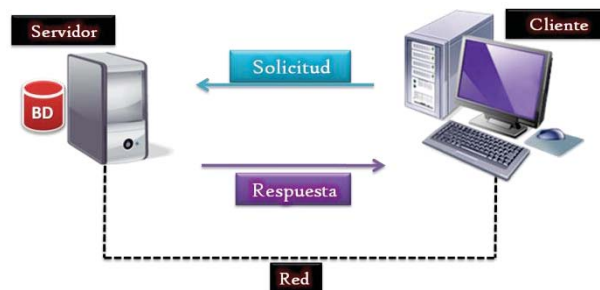


Figura 8. Arquitectura Cliente-Servidor

Tomado de (CharlieDaw2236, s.f.)

La arquitectura Cliente-Servidor cuenta con tres elementos sobre los cuales se desarrollan e implantan todos los sistemas que poseen la mencionada arquitectura.

La arquitectura comienza con el proceso cliente, que inicia el diálogo; el proceso que espera pasivamente a que lleguen peticiones de servicio llamado proceso de servidor; y, por último, el middleware, el cual es la interfaz que provee la conectividad entre el cliente y servidor.

4.3.1 Cliente

Un cliente o también llamado front-end, es aquel proceso que reclama servicios de otro; en otras palabras, es aquel proceso que permite al usuario formular los requerimientos y pasarlos al servidor. Este se encarga de manejar todas las funciones relacionadas con la manipulación y despliegue de datos, por lo que están desarrollados sobre plataformas que permiten construir interfaces gráficas de usuario o también conocidas GUI; además de acceder a los servicios distribuidos en cualquier parte de la red. (Tanenbaum, 2008)

4.3.2 Servidor

Un servidor o también llamado back-end, es aquel proceso que proporciona un servicio a otros procesos. Puede ser capaz de atender a múltiples clientes que realicen peticiones de algún recurso administrado por el mismo. El servidor normalmente maneja todas las funciones relacionadas con la mayoría de las reglas del negocio y los recursos de datos. Cabe mencionar, que puede darse el caso que un servidor actúe a su vez como cliente de otro servidor. (Tanenbaum, 2008)

4.4 JavaScript Object Notation – JSON

Para la aplicación móvil se ha utilizado JSON como formato para traer las respuestas que han hecho los pedidos a las Apis; la cual va a ser explicada a continuación.

JavaScript Object Notation, es un formato ligero para el intercambio de datos derivado del lenguaje JavaScript; y sirve para representar de forma simple estructuras de datos y arrays asociativos llamados objetos. A pesar de su relación con JavaScript, puede ser entendido por varios lenguajes de programación, ya que está concebido para ser fácil de leer y escribir tanto para seres humanos como para máquinas. (Json.org)

Dicho formato tiene la funcionalidad de serializar como transmitir estructuras de datos sobre una conexión de red, el cual es aplicado principalmente en aplicaciones Cliente-Servidor, lugar donde el flujo de datos entre los mencionados es de mucha importancia. Como consecuencia a su simplicidad y ligereza, el formato JSON se ha convertido en una opción totalmente viable hacia XML.

El formato JSON consta de dos estructuras, las cuales son:

- Colección de pares (nombre/valor)
- Lista ordenada de valores

Las estructuras son universales; todos los lenguajes de programación los soportan virtualmente de cualquier forma. Es razonable que un formato de intercambio de datos que es independiente del lenguaje se base en estas estructuras. (Json.org)

JSON como objeto, que comienza con una llave de apertura y termina con otra llave de cierre. Además, cada nombre tiene a continuación un signo de puntuación: Dos puntos. Y están separados por el signo de puntuación: coma. En la Figura 9 a continuación se muestra un ejemplo.

```
"tesisRamiroAlegriaUsoJSON"
{
  "empleados": [
    {
      "nombre": "Ramiro Alegria",
      "edad": 25,
    },
    {
      "nombre": "Andrés Calamaro",
      "edad": 45,
    }
  ]
}
```

Figura 9. Ejemplo de objeto JSON

El objeto JSON es el formato utilizado para trabajar en el intercambio de datos con las API.

4.5 Modelo Vista Controlador

Se ha decidido elegir uno de los patrones más utilizados para el desarrollo de software llamado Modelo Vista Controlador (MVC).

Este patrón para el desarrollo de software se basa en diferenciar los datos y la toda la parte del usuario en lo que tiene que ver a la lógica del negocio. El Modelo Vista Controlador recomienda construir tres componentes distintos; los cuales, se los puede ver en la Figura 10. (Reenskaug & Colien, 2009)

Este modelo está diseñado para facilitar el desarrollo mediante la reutilización del código y separación de conceptos; de esta manera ayudando al mantenimiento y escalabilidad de dichas aplicaciones.

Debido a que es utilizado en mayoría para aplicaciones web, ha sido elegido para este diseño donde la vista es la página HTML; el modelo es el sistema de gestión de base de datos, incluyendo la lógica interna; y el controlador es quien recibe los eventos y da una solución a los mismos.

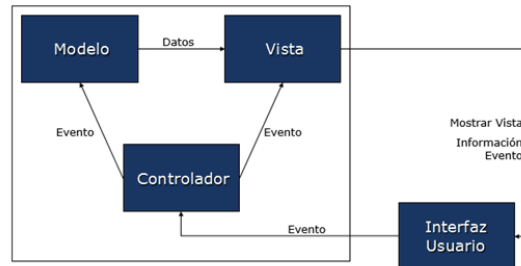


Figura 10. Arquitectura y diseño del patrón MVC

Tomado de (Páez, 2008)

A continuación, se va a definir los tres componentes del patrón MVC.

4.5.1 Modelo

Es quien gestiona todos los accesos de la información que el sistema opera, además trabaja para mostrar la información que el usuario solicita junto a la Vista y es accedido por el Controlador para realizar operaciones como: agregar, eliminar, buscar o actualizar datos.

4.5.2 Vista

La Vista representa el modelo para el usuario en un formato para que pueda interactuar adecuadamente con el mismo; el cual tiende a ser la interfaz de usuario en la mayoría de casos.

4.5.3 Controlador

El Controlador recibe, trata y responde a los eventos enviados por el usuario; al igual que a los eventos que han sido enviados por la aplicación. Además, invoca peticiones cuando se emite solicitudes sobre la información y permite enviar comandos a su Vista asociada; siempre y cuando, se solicite cambios en su forma. Este hace el papel de intermediario entre la Vista y el Modelo. Se lo conoce como el elemento más abstracto.

4.6 API

Una API (Application Programming Interface), es un conjunto de reglas y especificaciones que las aplicaciones pueden seguir para comunicarse entre ellas: sirviendo de interfaz entre programas diferentes de la misma manera que en la interfaz de usuario facilita la interacción humano-software. Las API son valiosas, ante todo, porque permiten hacer uso de funciones ya existentes en otro software para no estar reinventando la rueda constantemente, reutilizando así código. (Merino, 2014)

Dicho modelo es el utilizado en el diseño y desarrollo de este proyecto al tener la vista con el cliente, los controladores que trabajan con las API y el modelo que trabaja con la base de datos planteada.

4.7 Android Studio

Android Studio es un entorno de desarrollo integrado de la compañía JetBrains, que proporciona varias mejoras con respecto al plugin Android Developer Tools para Eclipse. Android Studio utiliza una licencia de software libre Apache 2.0, está programado en Java y es multiplataforma. Fue presentado en Google, con el objetivo de crear un entorno dedicado en exclusiva a la programación de aplicaciones para dispositivos Android, proporcionando a Google un mayor control sobre el proceso de producción. El mismo que es una alternativa a Eclipse con muchas mejoras que se puede descargar desde la Web. (Academia Android, 2014).

Android Studio es el software escogido para realizar la aplicación móvil debido a su fácil manejo ante las otras herramientas para su elaboración.

4.8 Php

Php es un lenguaje de programación, netamente para desarrollo web que usa código abierto; el cual puede ser embebido en HTML. Su acrónimo recurso es Hypertext Preprocessor. Lo que distingue a este lenguaje de programación de algo del lado del cliente como JavaScript es que el código es ejecutado en el servidor, generando HTML y enviándolo al cliente. El cliente recibirá el resultado de ejecutar el script, aunque no se sabrá el código subyacente anterior. El servidor web puede ser configurado para que procese HTML con PHP. (The Php Group)

Un ejemplo de Php se puede ver en la Figura 11 a continuación.

```
<html>
  <head>
    <title> Tesis Ramiro Alegría - Uso de PHP</title>
  </head>
  <body>
    <?php
      echo "Tesis UDLA 2016";
    ?>
  </body>
</html>
```

Figura 11. Ejemplo de código Php

4.9 Slim

Slim es un micro framework de Php que ayuda de una manera fácil y rápida a escribir código de aplicaciones Web al igual que Apis sencillas pero poderosas. Se puede decir que Slim es un despachador que recibe solicitudes HTTP, invocando rutinas de devolución de llamadas devolviendo una respuesta HTTP. (Lockhart, Smith, & Allen). Este micro framework es usado para las API.

4.10 Bootstrap

Twitter es la empresa que creó un framework denominado Bootstrap; en el cual se pueden crear interfaces de tipo web, soportando estilos (CSS) y JavaScript, con la diferencia de que incorpora la vista de un sitio web, proporcionalmente a la de un dispositivo en el cual se lo vaya a visualizar. Esta técnica se le conoce como diseño adaptativo. (Solis, 2014). El mismo framework usado para el sistema web.

4.11 JavaScript

El cliente usa el lenguaje de programación llamado JavaScript con el fin de crear efectos llamativos, innovadores, dinámicos, poniendo en juego la creación del usuario, a través de las páginas web. Los navegadores de hoy en día interpretan este código integrado en las páginas web. (González).

Un ejemplo de JavaScript se lo puede observar en la Figura 12.

```
<html>
  <head>
    <title> Tesis Ramiro Alegría - Uso de JavaScript</title>
  </head>
  <body>
    <script type="text/javascript">
      document.write('Tesis UDLA 2016');
    </script>
  </body>
</html>
```

Figura 12. Ejemplo de código JavaScript

4.12 Google Maps API

Para usar la Google Maps Android API, se requiere registrar a un proyecto con la aplicación en la Google Developers Console y obtener una clave de API Google; la cual se puede agregar a la aplicación mencionada. Este tipo de clave es una clave de Android. Todas las aplicaciones de Android tienen una firma cuyo certificado digital es la clave privada que se obtiene. Estas claves están vinculadas a pares de paquetes y certificados específicos. (Google, 2016)

4.13 Volley

Google es la empresa que desarrolló una librería llamada Volley, con el propósito único de mejorar el envío de peticiones de tipo Http que realizan las aplicaciones de Sistema Operativo Android, con rumbo a servidores externos. Así, actuando como una interfaz de alto nivel, Despreocupando a quien desarrolla de hilos y proceso de parsing, para permitir publicar de manera sencilla los resultados en el hilo principal. (Revelo, 2015).

Esta librería es la usada para conectarse entre la aplicación móvil y las API como se puede ver en la figura 14.

4.14 MySQL

MySQL es un gestor de base de datos de tipo relacional, basado en SQL, que es un lenguaje de consulta que cumple con una estructura. Se lo puede ejecutar en diferentes plataformas como son: Linux, UNIX, Windows. Se puede decir que MySQL trabaja de mayor manera en sistemas web y publicaciones en línea. (Rouse). La misma base de datos que es usada para el desarrollo de este sistema.

4.15 Plivo

Plivo es una plataforma API en la nube que sirve para el envío de SMS a nivel internacional; el mismo que, sirve para cualquier desarrollo en el lenguaje de programación Php. La misma plataforma que es usada en este desarrollo para el envío de mensajes de texto.

4.16 Draw.IO

Draw.IO es un software libre que se utiliza como herramienta para realizar diagramas de diferentes tipos como diagramas de flujo, diagramas de proceso, arquitecturas de software, diagramas de red, UML, entre otros. El cual es utilizado para realizar los esquemas y arquitecturas en el presente proyecto.

5. Capítulo V. Desarrollo Scrum

5.1 Product Backlog

En el Product Back Log se detallan todas las iteraciones como se muestra en la tabla 1 a continuación.

Tabla 1. Product BackLog

Product BackLog				
Id	Backlog ítem	Esfuerzo	Prioridad	Estado
BNUIO-1	Diseño de arquitectura	2	BAJA	Sin comenzar / Iniciado / Terminado
BNUIO -2	Contratación de servidor	1	BAJA	Sin comenzar / Iniciado / Terminado
BNUIO -3	Base de Datos	3	MEDIA	Sin comenzar / Iniciado / Terminado
BNUIO -4	Modelo	2	BAJA	Sin comenzar / Iniciado / Terminado
BNUIO -5	Controlador	2	BAJA	Sin comenzar / Iniciado / Terminado
BNUIO -6	Composer y Helper	2	BAJA	Sin comenzar / Iniciado / Terminado
BNUIO -7	API's	4	ALTA	Sin comenzar / Iniciado / Terminado
BNUIO -8	Aplicación Web	4	ALTA	Sin comenzar / Iniciado / Terminado
BNUIO -9	Aplicación Móvil	5	ALTA	Sin comenzar / Iniciado / Terminado

5.2 Sprint Backlog

5.2.1 Sprint 1

El Sprint 1 contempla los tres primeros product backlog BNUIO -1, BNUIO -2 y BNUIO-3 respectivamente como se muestra en la tabla a continuación.

Tabla 2. Sprint 1

Id	Descripción	Criterio de Aceptación
BNUIOS1-1	Diagrama de Servidores	Crear un diagrama donde se evidencie la interacción del usuario con los servidores
BNUIOS1-2	Diseño de la arquitectura	Crear un diagrama donde se pueda ver el funcionamiento de la aplicación
BNUIOS 2	Contratación del servidor	Obtener el mejor servidor de acuerdo a las necesidades de uso
BNUIOS	Creación de la base de datos	Crear una base de datos que cumpla con los requisitos para desarrollar la aplicación.

Desarrollo del Sprint

5.2.1.1 BNUIOS1-1 Diagrama de servidores

El diseño del sistema arranca con un diagrama donde se contrata al proveedor GoDaddy un web hosting con una base de datos MySQL y un servidor donde se encuentran las aplicaciones en Apache; ambos con sistema operativo Linux. Para un mejor entendimiento se presenta la Figura 13.

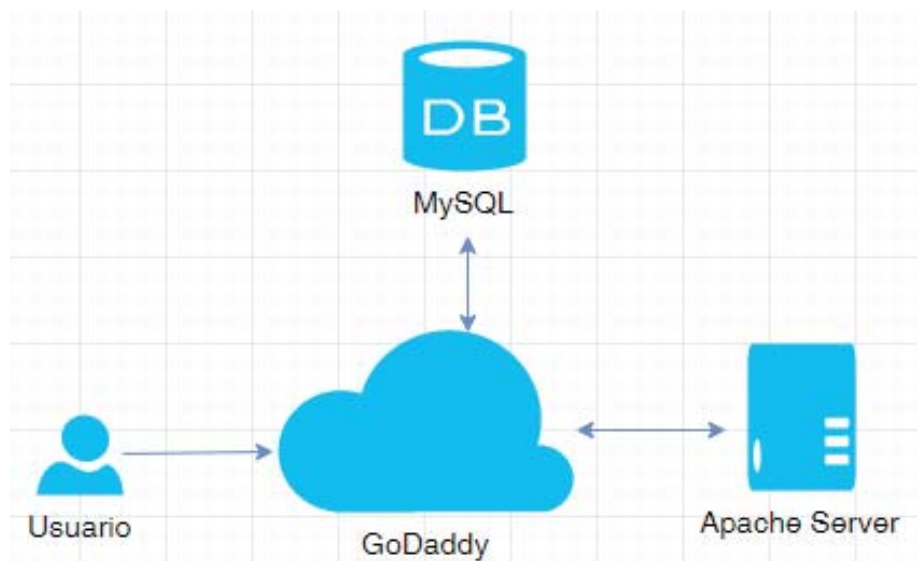


Figura 13. Diagrama de Servidores

5.2.1.2 BNUIOS1-2 Diseño de la arquitectura

Después de tener el diagrama de servidores, se pensó en la arquitectura del sistema; la misma que se la puede apreciar en la Figura 14, presentada a continuación.

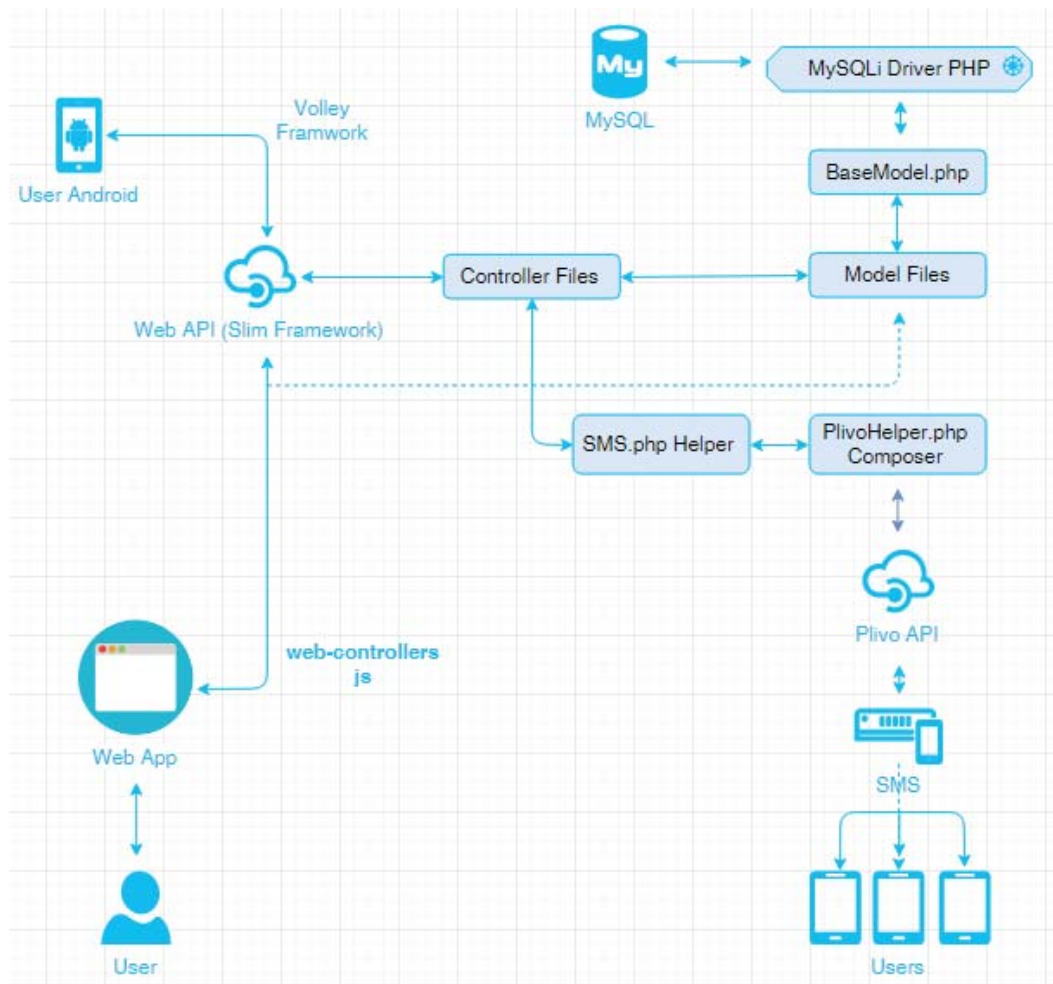


Figura 14. Arquitectura del sistema

5.2.1.3 BNUIOS2 Contratación de servidor

Después de haber diseñado el sistema, se procedió al desarrollo del mismo. El entorno de desarrollo es Android Studio el cual tiene las herramientas necesarias para desarrollar esta aplicación móvil. Además, para el desarrollo web se utilizó Php, manejando Bootstrap para hacer responsiva a la aplicación; por otro lado, MySQL fue la herramienta usada para la base de datos.

Se procedió a la adquisición de un servidor web a través del proveedor GoDaddy con un plan para Linux económico con 100GB de almacenamiento y ancho de banda limitada. Este servidor web tiene un dominio del sitio el cual es: www.buynowuio.com.

5.2.1.4 BNUIOS3 Base de Datos

A continuación, se procedió a crear la base de datos MySQL utilizando phpMyAdmin; la misma que contiene las tablas: usuario, tienda, mensaje, producto, compra, mensaje como el detalle de la compra como se muestra en la Figura 15.

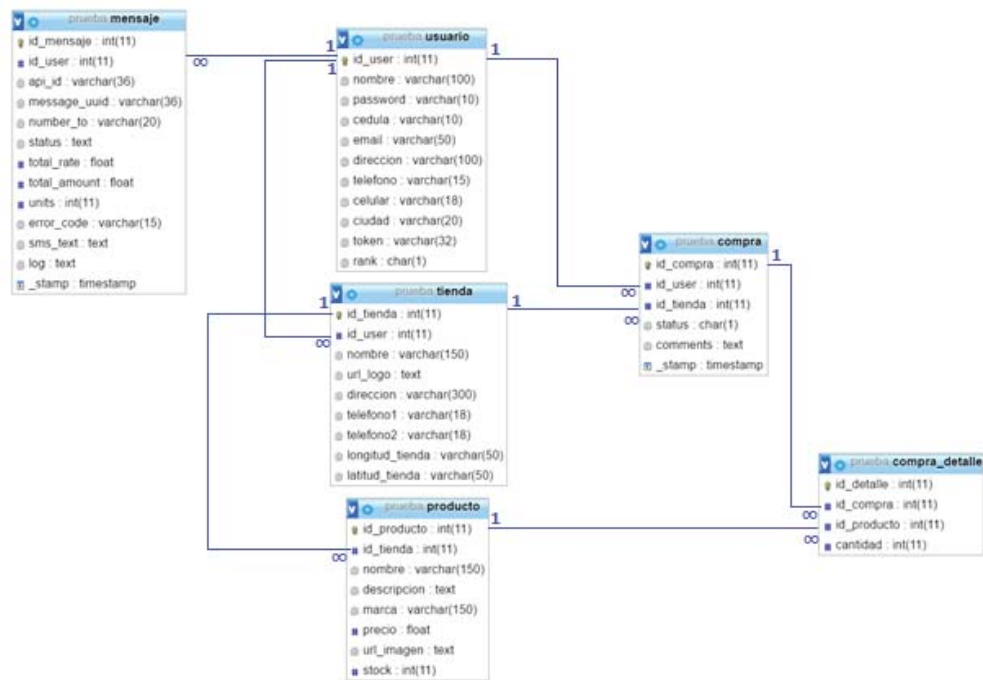


Figura 15. Modelo Base de Datos

5.2.2 Sprint 2

En el Sprint 2 se crean las bases de la aplicación como se puede ver en la tabla 3 a continuación.

Tabla 3. Sprint 2

Id	Descripción	Criterio de Aceptación
BNUI OS4	Modelo	Crear del modelo para trabajar con la base de datos de la aplicación
BNUI OS5	Controlador	Crear del controlador del modelo para la aplicación
BNUI OS6	Composer y Helper	Crear el composer y helper para trabar con la API del proveedor de mensajes de texto
BNUI OS7	API's	Crear las API para conectar la aplicación móvil y web

Desarrollo del Sprint

5.2.2.1 BNUIOS4 Creación de Modelos

Se crearon los modelos lenguaje de programación Php haciendo las consultas necesarias a la base de datos; dichos modelos se los puede ver en la Tabla 2, con la descripción de su uso:

Tabla 4. Modelos del Sistema

Modelo	Descripción
BaseModel.php	Sirve para la conexión con la base de datos.
MMarket.php	Modelo con los scripts necesarios para trabajar con las tiendas.

MProduct.php	Modelo con los scripts necesarios para trabajar con los productos.
MSMS.php	Modelo con los scripts necesarios para trabajar con los mensajes de texto.
MUser.php	Modelo con los scripts necesarios para trabajar con los usuarios.
MBuy.php	Modelo con los scripts necesarios para trabajar con las compras realizadas en el sistema

5.2.2.2 BNUIOS5 Controladores

Tal cual se explicó en MVC (Modelo – Vista - Controlador), se procedió a crear los Controladores que trabajan con las API como funciones, los nombres de los controladores se los lista a continuación:

- BaseController.php. - contiene el índice de los controladores
- CMarket.php. - Tiene las funciones de la tienda
- CProduct.php. - tiene las funciones del producto
- CUser.php. - tiene las funciones del usuario
- CBuy.php. - tiene las funciones de la compra

5.2.2.3 BNUIOS6 Composer y Helper

Se crean los archivos Php: SMS.php y PlivoHelper.php para conectarse a la API del proveedor de mensajes de texto llamado “Plivo” y de la misma manera conectarse a los Controladores antes mencionados.

Se crea el archivo SMS_callback.php para establecer el estado de los mensajes con el proveedor Plivo.

5.2.2.4 BNUIOS7 API

Se crearon ciertas API que van a interactuar con el sistema web, tal como el sistema móvil. En la tabla que está incluida en los anexos, se puede ver cada API con su información respectiva. (Ver Anexo 1).

Además, dentro de la vista se encuentra la página de inicio llamada index.html.

5.2.3 Sprint 3

En este sprint se detalla todo con relación a la creación de la aplicación web como se puede observar en la tabla a continuación.

Tabla 5. Sprint 3

Id	Descripción	Criterio de Aceptación
BNUIOS8-1	Creación de java scripts	Crear los JavaScripts para la aplicación
BNUIOS8-2	Elaborar estructura HTML de inicio	Crear la pantalla de inicio para la aplicación
BNUIOS8-3	Pantalla para cambiar la contraseña	Crear una pantalla para cambiar la contraseña
BNUIOS8-4	Pantalla para el inicio de sesión	Crear pantalla para el inicio de sesión del usuario
BNUIOS8-5	Opción para el cierre de sesión	Crear opción para el cierre de sesión del usuario
BNUIOS8-6	Pantalla donde se muestra el mapa	Crear pantalla donde se muestre el mapa de todas las tiendas
BNUIOS8-7	Pantalla para crear tiendas	Crear pantalla donde se puedan crear las tiendas
BNUIOS8-8	Pantalla para administrar tiendas	Crear pantalla donde se puedan administrar las tiendas
BNUIOS8-9	Pantalla de perfil de	Crear pantalla donde se puede ver el

	usuario	perfil del usuario
BNUIOS8-10	Pantalla para crear productos	Crear pantalla para crear productos dentro de las tiendas
BNUIOS8-11	Pantalla para administrar productos	Crear pantalla para administrar los productos dentro de las tiendas
BNUIOS8-12	Pantalla para registrar usuarios	Crear pantalla para registrar nuevos usuarios
BNUIOS8-13	Pantalla para reportes de mensajes de texto	Crear pantalla donde se reporten los mensajes de texto enviados con sus respectivos atributos
BNUIOS8-14	Pantalla de reportes de solicitudes de compras	Crear pantalla donde se muestra el reporte de solicitudes de compras
BNUIOS8-15	Pantalla de detalle de la compra	Crear pantalla donde se muestra el detalle de la compra.
BNUIOS8-16	Pantalla de reporte de ventas	Crear pantalla donde se muestra el reporte de ventas
BNUIOS8-17	Pantalla de reporte de compras realizadas	Crear pantalla donde se muestra el reporte de compras realizadas
BNUIOS8-18	Pantalla para crear usuarios.	Crear pantalla donde se pueda crear usuarios
BNUIOS8-19	Pantalla para editar usuarios.	Crear pantalla donde se pueda editar usuarios
BNUIOS8-20	Pantallas usadas cuando se ha olvidado la contraseña.	Crear pantalla de ayuda cuando el usuario ha perdido la contraseña

Desarrollo del Sprint

5.2.3.1 BNUIOS 8-1 Creación de JavaScripts

Dentro de la vista se encuentran los respectivos Java Scripts que trabajan con los controladores; los mismos que son listados a continuación:

Tabla 6. Java Scripts del Sistema

JavaScript	Descripción
Web-change-password.js	Sirve para cambiar de contraseña
Web-login.js	Sirve para iniciar sesión
Web-market-create.js	Sirve para crear tiendas
Web-market-update.js	Sirve para editar tiendas
Web-my-profile.js	Sirve para ver el perfil del usuario
Web-product-create.js	Sirve para crear productos
Web-product-update.js	Sirve para editar productos
Web-signup.js	Sirve para registrar usuarios
Web-user-create.js	Sirve para crear usuarios

5.2.3.2 BUIOS 8-2 Elaboración estructura HTML de inicio

Se crea la pantalla principal utilizando bootstrap con sus respectivas clases y javascripts como se muestra en la figura a continuación.



Figura 16. Pantalla de inicio de la aplicación web

5.2.3.3 BNUIOS 8-3 Pantalla para cambiar la contraseña

Dentro de esta pantalla se crea el archivo php: Change-password.php, donde se puede visualizar la pantalla mostrada en la figura 17.

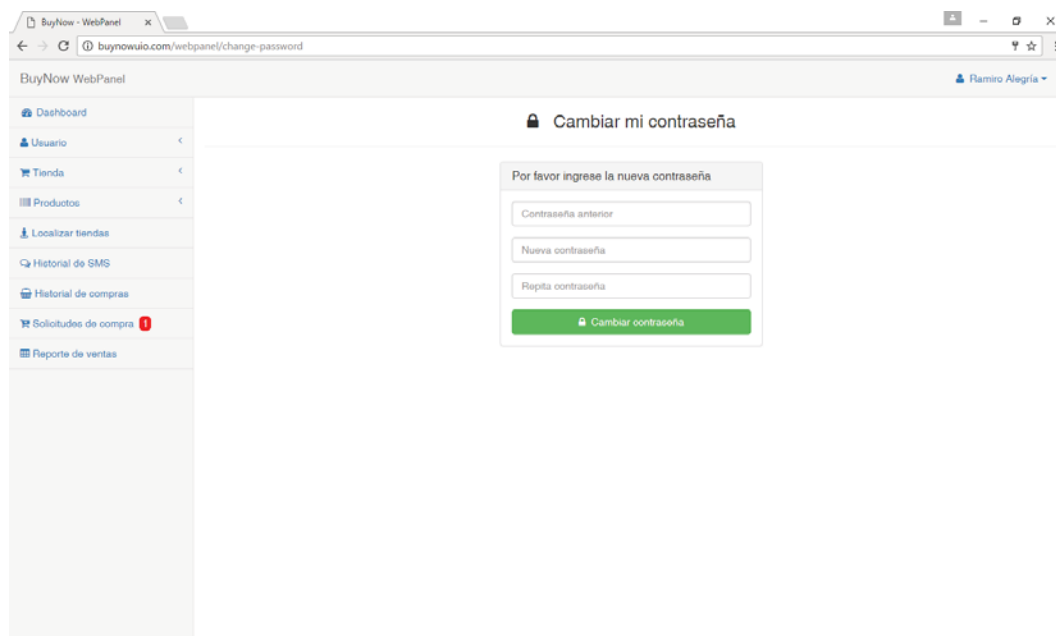


Figura 17. Pantalla web para cambiar la contraseña

5.2.3.4 BNUIOS 8-4 Pantalla para el inicio de sesión

Dentro de esta pantalla se procede a crear el archivo phpIndex.php con su respectiva pantalla mostrada en la figura a continuación.

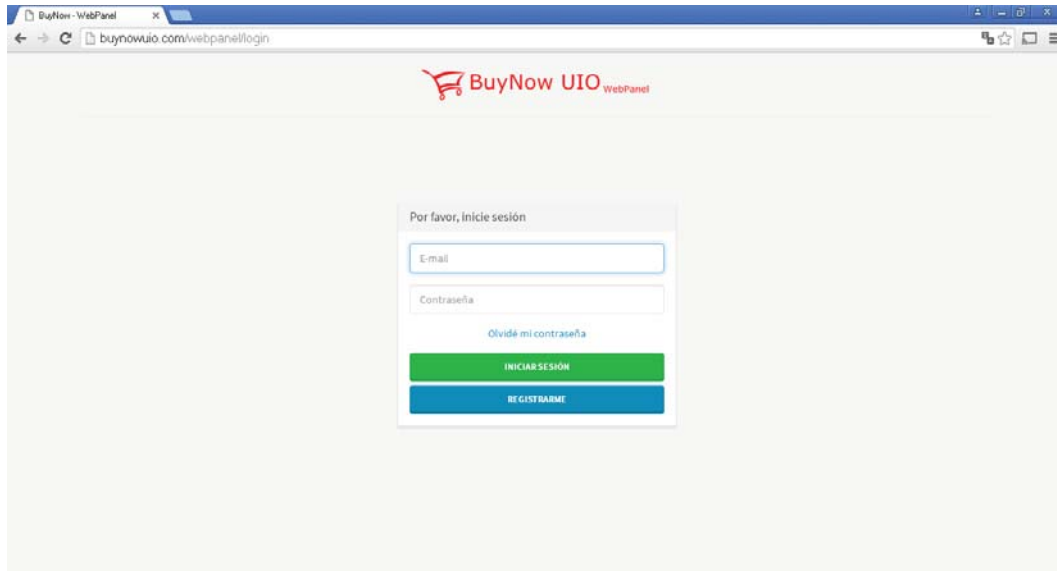


Figura 18. Pantalla web para el inicio de sesión

5.2.3.5 BNUIOS 8-5 Opción para cerrar sesión

Para esta opción se crea el archivo php con el nombre Logout.php

5.2.3.6 BNUIOS 8-6 Pantalla donde se muestra el mapa

En la figura 19 se puede observar el mapa donde se encuentran las tiendas que cuentan con la aplicación. Para esto se crea el archivo php de nombre Map-view.php

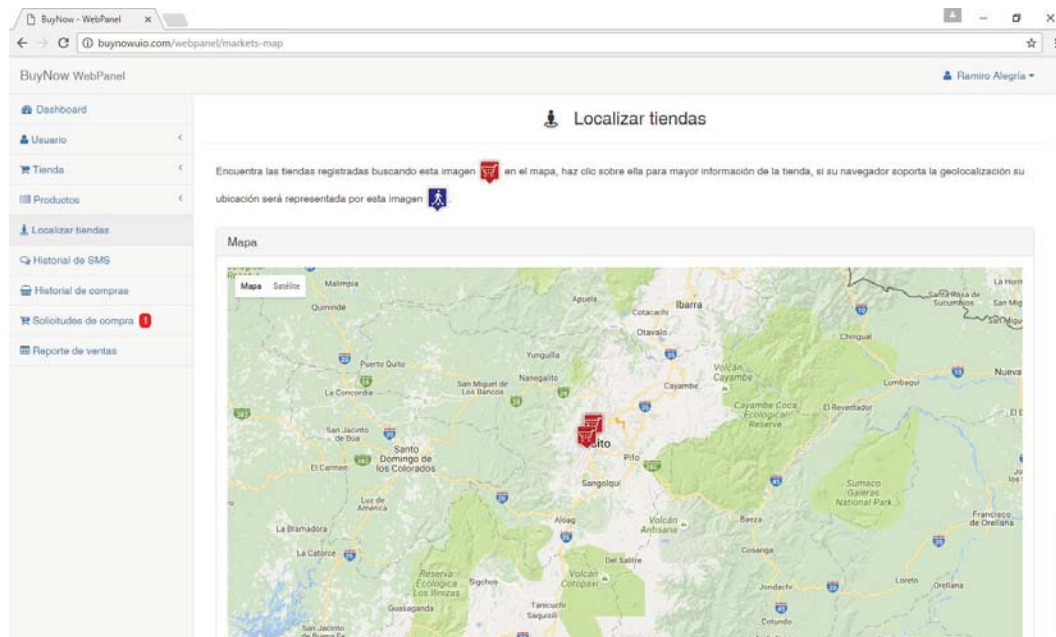
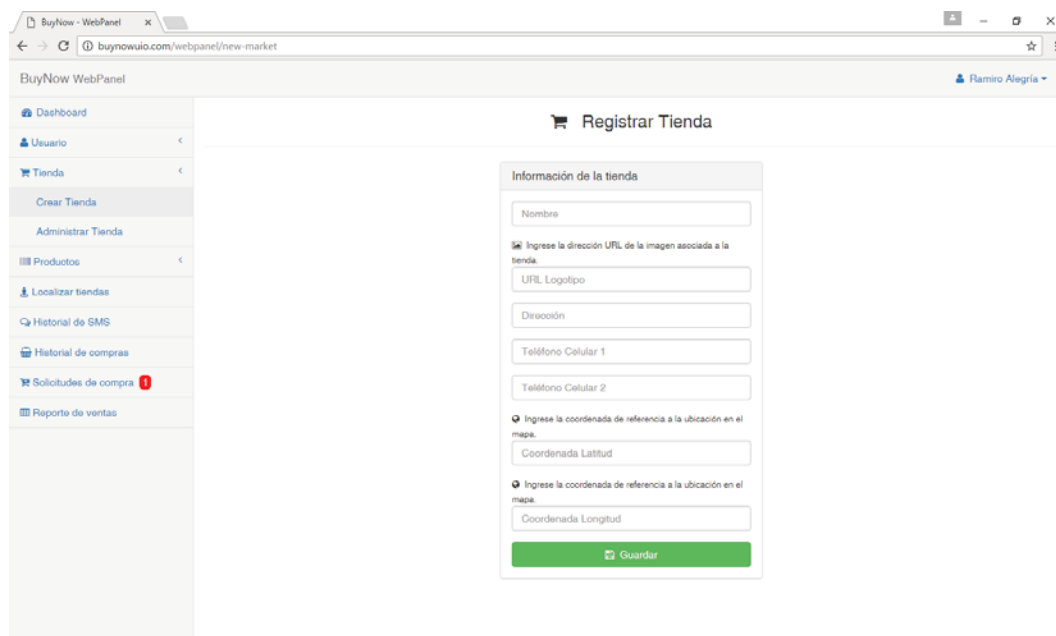


Figura 19. Pantalla web donde muestra el localizador de tiendas

5.2.3.7 BNUIOS 8-7 Pantalla para crear tiendas

Para la creación de las tiendas se crea el archivo Market-create.php; el mismo que se puede visualizar en la figura a continuación.



The screenshot displays a web browser window with the address bar showing 'buynowulo.com/webpanel/new-market'. The page title is 'BuyNow WebPanel' and the user is logged in as 'Ramiro Alegria'. A sidebar on the left contains navigation links: Dashboard, Usuario, Tienda, Crear Tienda (highlighted), Administrar Tienda, Productos, Localizar tiendas, Historial de SMS, Historial de compras, Solicitudes de compra (with a red notification icon), and Reporte de ventas. The main content area is titled 'Registrar Tienda' and contains a form titled 'Información de la tienda'. The form includes the following fields: 'Nombre', 'Ingresar la dirección URL de la imagen asociada a la tienda.' (with a sub-field 'URL Logotipo'), 'Dirección', 'Teléfono Celular 1', 'Teléfono Celular 2', 'Ingresar la coordenada de referencia a la ubicación en el mapa.' (with a sub-field 'Coordenada Latitud'), and another 'Ingresar la coordenada de referencia a la ubicación en el mapa.' (with a sub-field 'Coordenada Longitud'). A green 'Guardar' button is located at the bottom of the form.

Figura 20. Pantalla web para creación de tiendas

5.2.3.8 BNUIOS 8-8 Pantalla para administrar tiendas

Así mismo, se crea un archivo de nombre Market-update.php para la administración de tiendas, con su respectiva pantalla, que se puede observar en la figura 21.

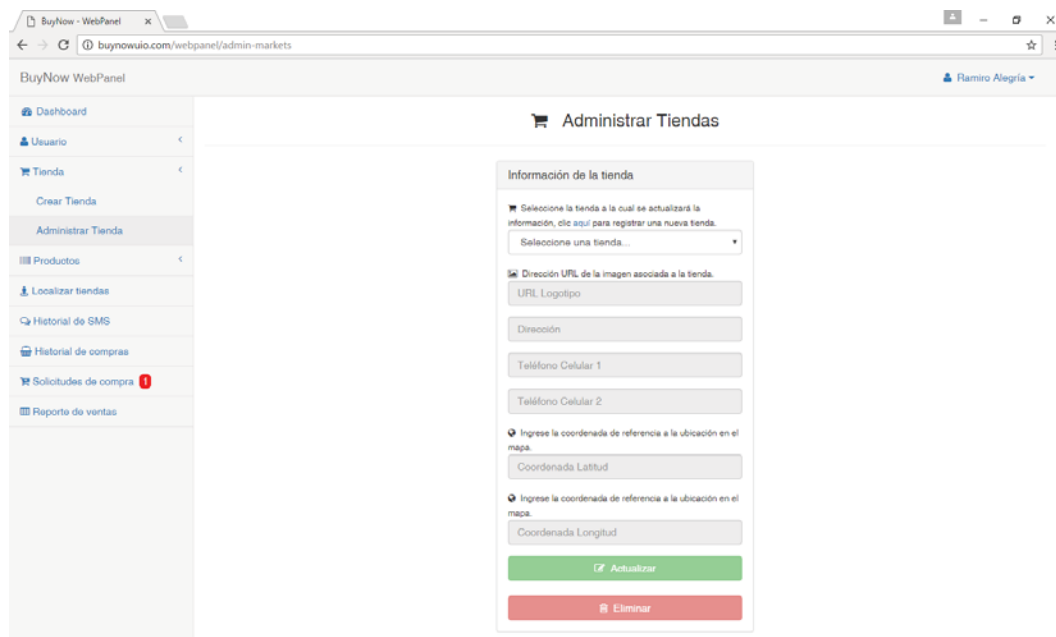


Figura 21. Pantalla web para administrar tiendas

5.2.3.9 BUIOS 8-9 Pantalla de perfil de usuario

Existe una pantalla donde se puede ver el perfil del usuario donde su archivo creado tiene el nombre de My-profile.php, y se lo puede ver en la figura a continuación.

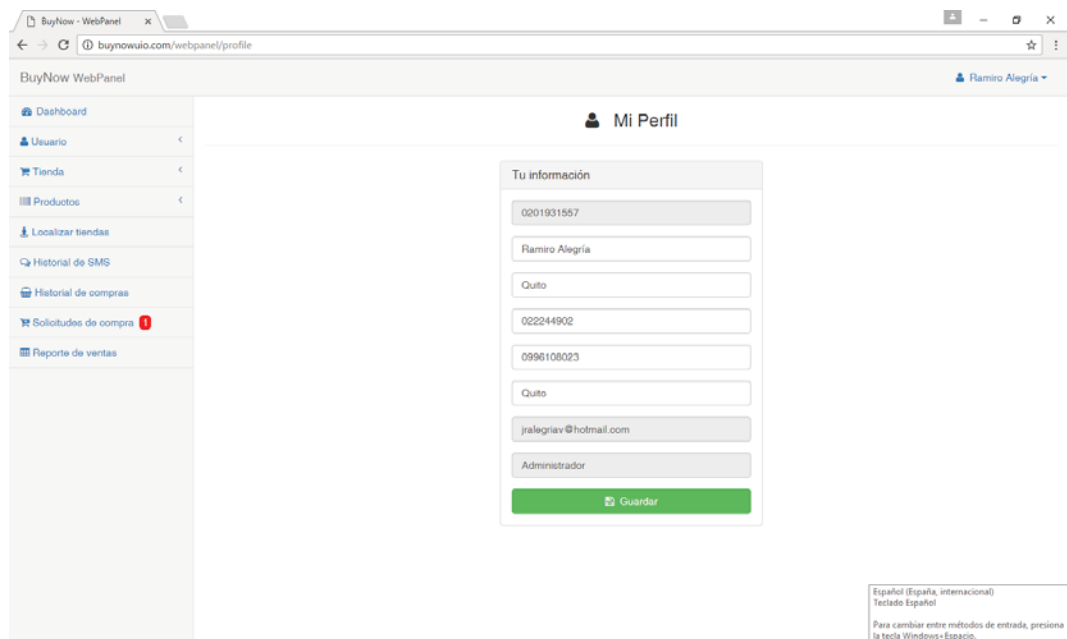
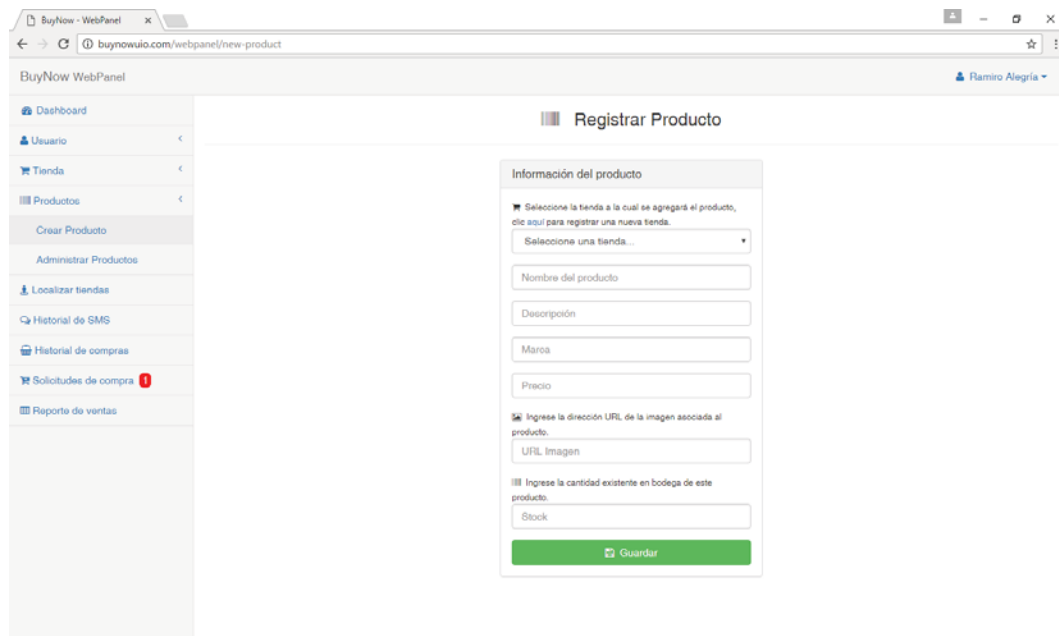


Figura 22. Pantalla web para ver el perfil del usuario

5.2.3.10 BNUIOS 8-10 Pantalla para crear productos

Existe una pantalla para crear productos que se la puede visualizar en la figura 23, donde su archivo lleva el nombre de Product-create.php.



The image shows a web browser window displaying the 'Registrar Producto' (Register Product) page in the BuyNow WebPanel. The browser's address bar shows the URL 'buynowulo.com/webpanel/new-product'. The page title is 'Registrar Producto'. On the left, there is a sidebar menu with options: Dashboard, Usuario, Tienda, Productos, Crear Producto (highlighted), Administrar Productos, Localizar tiendas, Historial de SMS, Historial de compras, Solicitudes de compra (with a red notification icon), and Reporte de ventas. The main content area contains a form titled 'Información del producto' with the following fields and instructions:

- Selección de tienda:** 'Seleccione la tienda a la cual se agregará el producto, clic aquí para registrar una nueva tienda.' Below this is a dropdown menu labeled 'Seleccione una tienda...'.
- Nombre del producto:**
- Descripción:**
- Marca:**
- Precio:**
- URL de imagen:** 'Ingrese la dirección URL de la imagen asociada al producto.' Below this is a text input field labeled 'URL Imagen'.
- Stock:** 'Ingrese la cantidad existente en bodega de este producto.' Below this is a text input field labeled 'Stock'.

At the bottom of the form is a green button labeled 'Guardar' with a save icon.

Figura 23. Pantalla web para crear productos

5.2.3.11 BNUIOS 8-11 Pantalla para administrar productos

Tal como existe una pantalla para crear productos, existe una pantalla para administrarlos con su archivo de nombre Product-update.php que se lo puede observar en la figura a continuación.

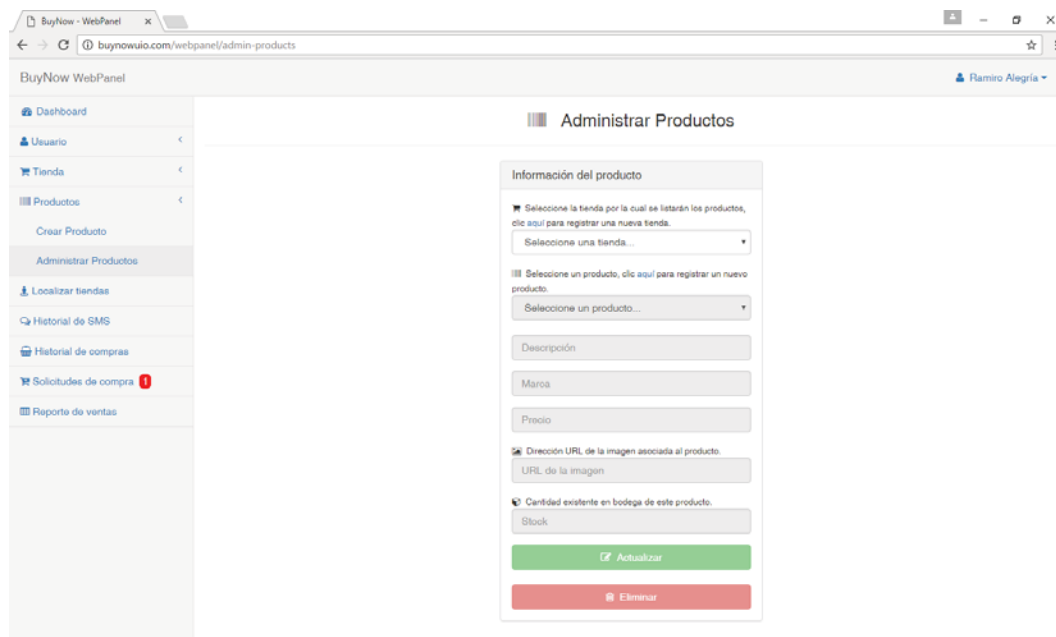
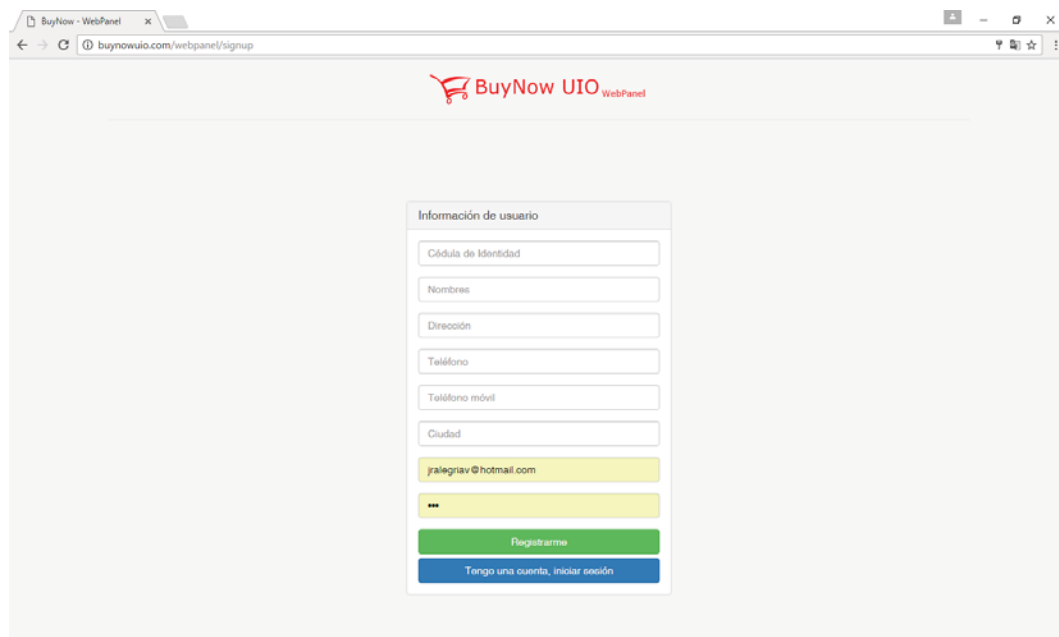


Figura 24. Pantalla web para administrar productos

5.2.3.12 BNUIOS 8-12 Pantalla para registrar usuarios

Dentro de la pantalla para registrar usuarios se crea el archivo Signup.php donde se pueden crear nuevos usuarios y su respectiva pantalla se visualiza en la figura a continuación.

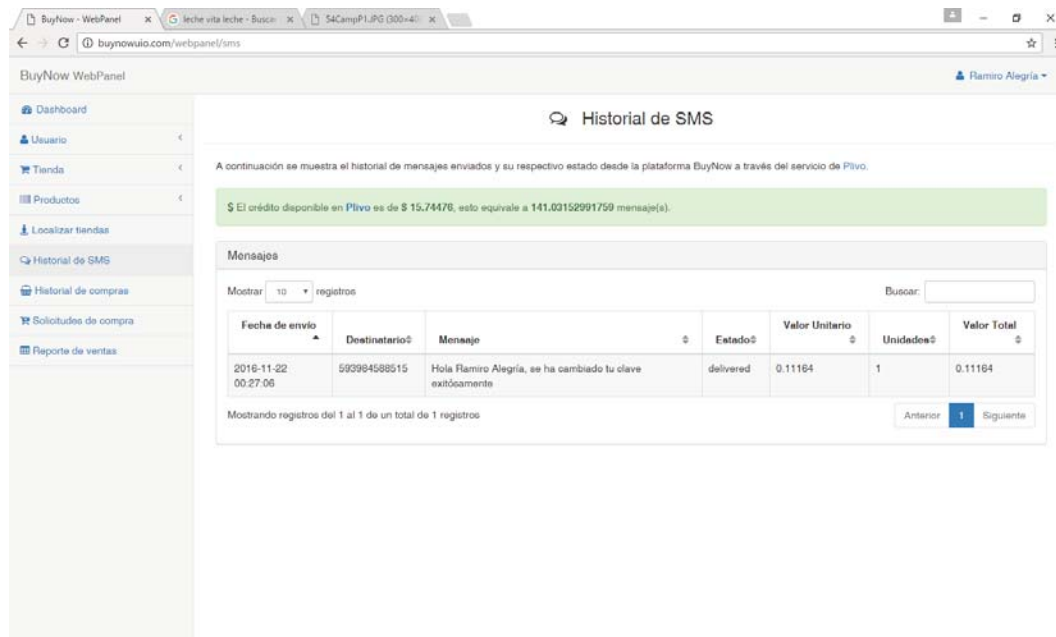


The image shows a web browser window displaying the registration page for BuyNow UIO. The browser's address bar shows the URL "buynowuio.com/webpanel/signup". The page features the "BuyNow UIO WebPanel" logo at the top. Below the logo is a form titled "Información de usuario" (User Information). The form contains several input fields: "Cédula de Identidad", "Nombre", "Dirección", "Teléfono", "Teléfono móvil", and "Ciudad". Below these fields are two yellow input fields for email, with the first one containing "jalegravi@hotmail.com" and the second one containing three asterisks. At the bottom of the form are two buttons: a green "Registrarme" button and a blue "Tengo una cuenta, iniciar sesión" button.

Figura 25. Pantalla web para registro de usuarios

5.2.3.13 BNUIOS 8-13 Pantalla para reporte de mensajes de texto

Existe una pantalla donde se muestran los mensajes de texto enviados desde la aplicación, su nombre es Sms-view.php y se lo puede observar en la figura 26.



The screenshot shows a web browser window with the URL `buynowuio.com/webpanel/sms`. The page title is "Historial de SMS". A green notification bar states: "El crédito disponible en Plivo es de \$ 15.74476, esto equivale a 141.03152991759 mensaje(s)". Below this is a table titled "Mensajes" with the following data:

Fecha de envío	Destinatario	Mensaje	Estado	Valor Unitario	Unidades	Valor Total
2016-11-22 00:27:06	593994588515	Hola Ramiro Alegria, se ha cambiado tu clave exitosamente	delivered	0.11164	1	0.11164

At the bottom of the table, it says "Mostrando registros del 1 al 1 de un total de 1 registros". Navigation buttons for "Anterior" and "Siguiente" are visible, with "1" highlighted in the center.

Figura 26. Pantalla web para el reporte de mensajes de texto

5.2.3.14 BNUIOS 8-14 Pantalla de solicitudes de compra

Para esta pantalla se puede verificar la figura 27, donde se ha creado el archivo Purchase-view.php.

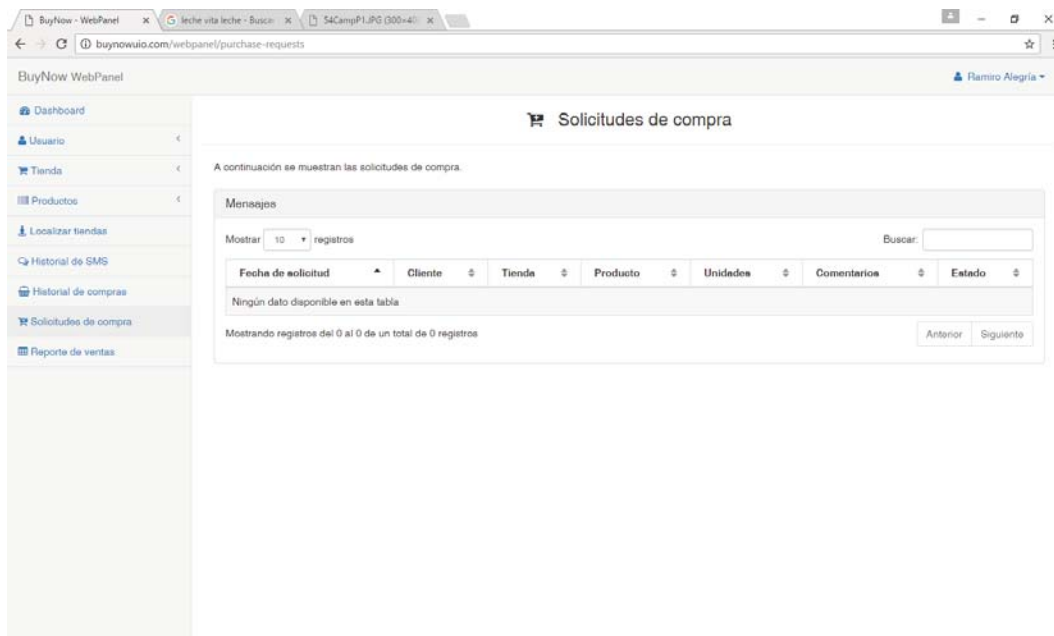
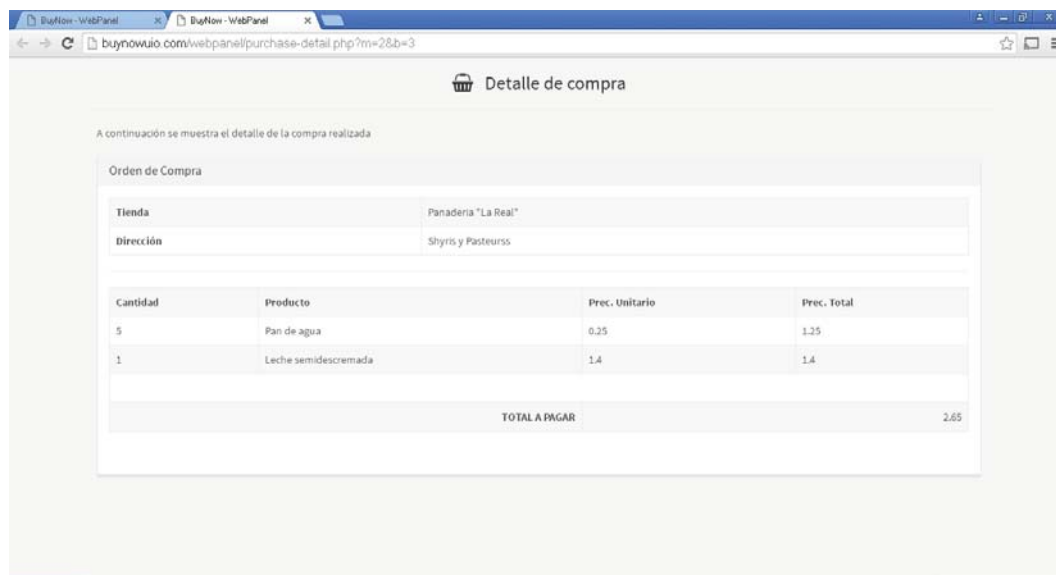


Figura 27. Pantalla web para las solicitudes de compra

5.2.3.15 BNUIOS 8-15 Pantalla para el detalle de compra

Dentro del historial de compra existe el detalle de la misma, el archivo para este detalle tiene el nombre de Purchase.detail.php y se lo muestra en la figura 28.



A continuación se muestra el detalle de la compra realizada

Orden de Compra

Tienda	Panadería "La Real"		
Dirección	Shyris y Pasteurss		

Cantidad	Producto	Prec. Unitario	Prec. Total
5	Pan de agua	0.25	1.25
1	Leche semidescremada	1.4	1.4
TOTAL A PAGAR			2.65

Figura 28. Pantalla web para del detalle de compra

5.2.3.16 BNUIOS 8-16 Pantalla para el reporte de ventas

Para este reporte se ha creado el archivo Sales-view.php y se lo puede ver en la figura a continuación.

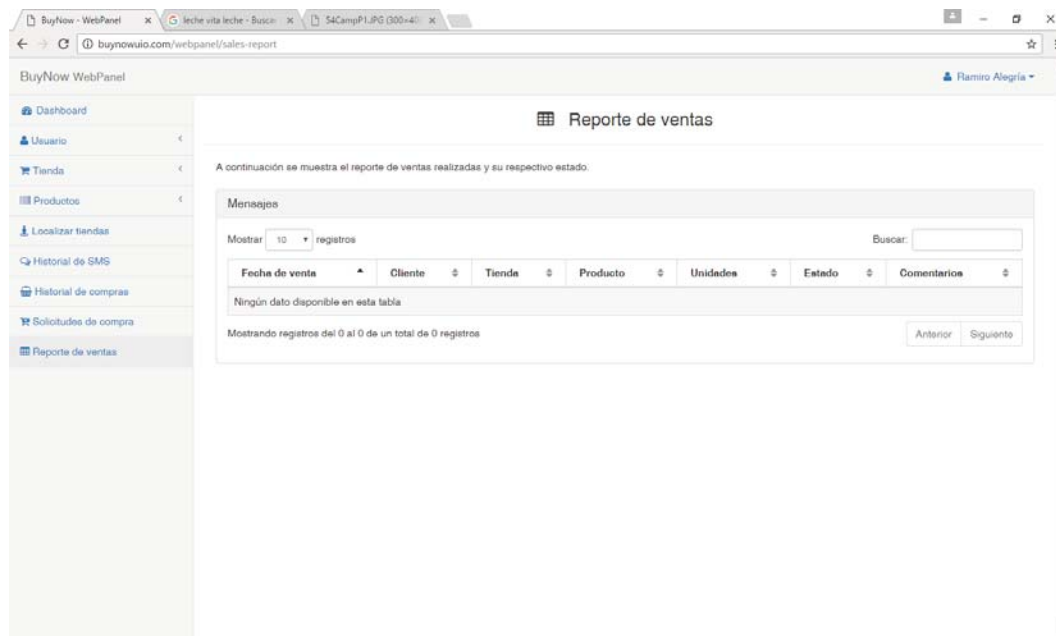


Figura 29. Pantalla web para el reporte de ventas

5.2.3.17 BNUIOS 8-17 Pantalla de compras realizadas

Esta pantalla se la puede ver en la figura 30. Para este reporte el archivo creado es Shopping-view.php.

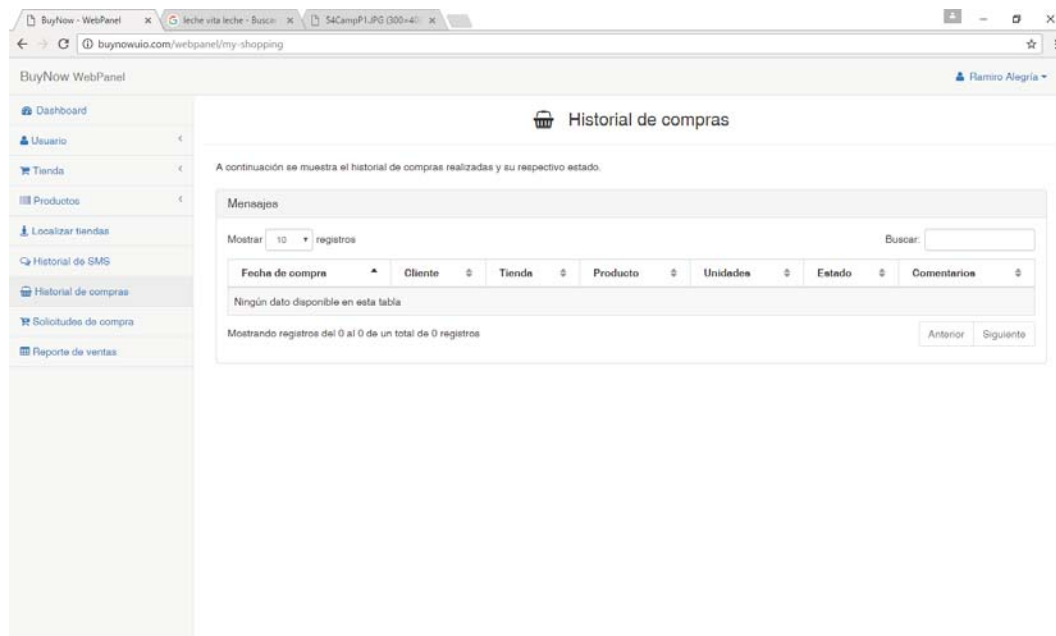
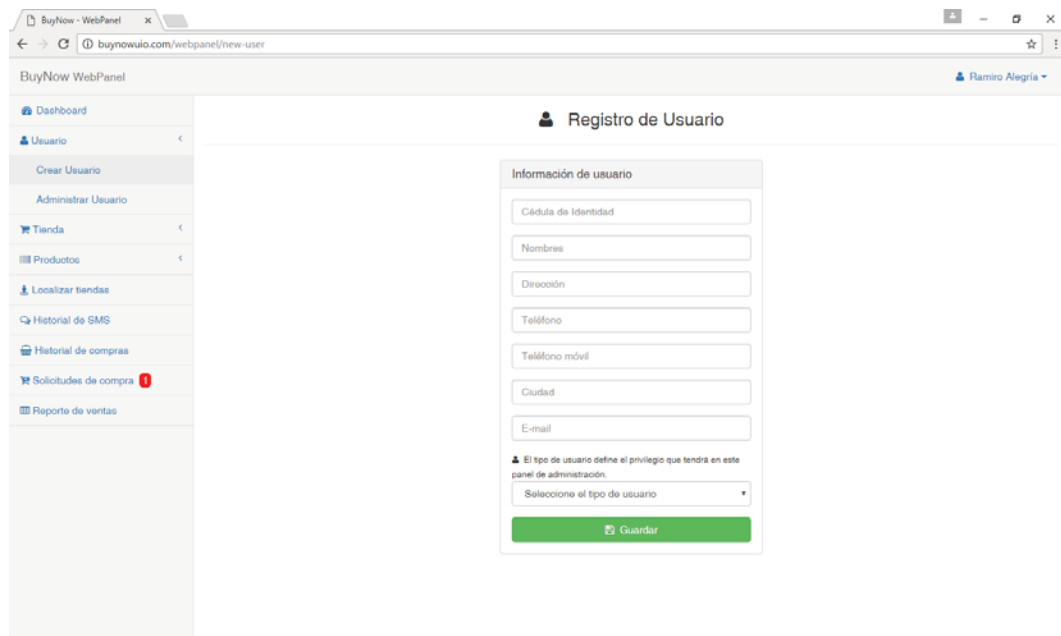


Figura 30. Pantalla web para ver el historial de compras realizadas

5.2.3.18 BNUIO 8-18 Pantalla para crear usuarios

Para crear usuarios se ha creado el archivo User-create.php; el mismo que se lo puede observar en la figura a continuación.



The screenshot shows a web browser window with the URL `buynowuio.com/webpanel/new-user`. The page title is "Registro de Usuario". On the left, there is a sidebar menu with options: Dashboard, Usuario, Crear Usuario (highlighted), Administrar Usuario, Tienda, Productos, Localizar tiendas, Historial de SMS, Historial de compras, Solicitudes de compra (with a red notification icon), and Reporte de ventas. The main content area contains a form titled "Información de usuario" with the following fields: Cédula de Identidad, Nombres, Dirección, Teléfono, Teléfono móvil, Ciudad, and E-mail. Below these fields is a dropdown menu labeled "Seleccione el tipo de usuario" and a green "Guardar" button. A small warning icon and text are present above the dropdown: "El tipo de usuario define el privilegio que tendrá en este panel de administración."

Figura 31. Pantalla web para crear usuarios

5.2.3.19 BNUIO 8-19 Pantalla para editar usuarios

Para la administración o edición de usuarios existe el archivo User-update.php. Esta edición de usuarios se lo puede ver en la figura 32.

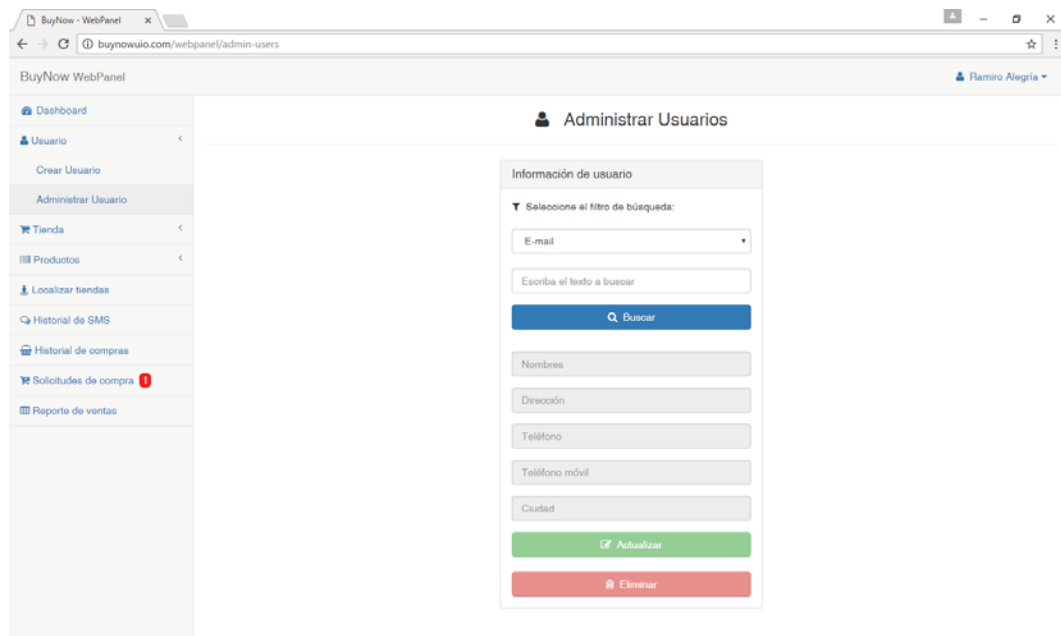


Figura 32. Pantalla web para administrar usuarios

5.2.3.20 BNUIOS 8-20 Pantalla de olvido de contraseña

Se ha creado una pantalla donde se puede recuperar la contraseña llamada Forgot-password-reset.php, la misma que se le puede ver en la figura a continuación.

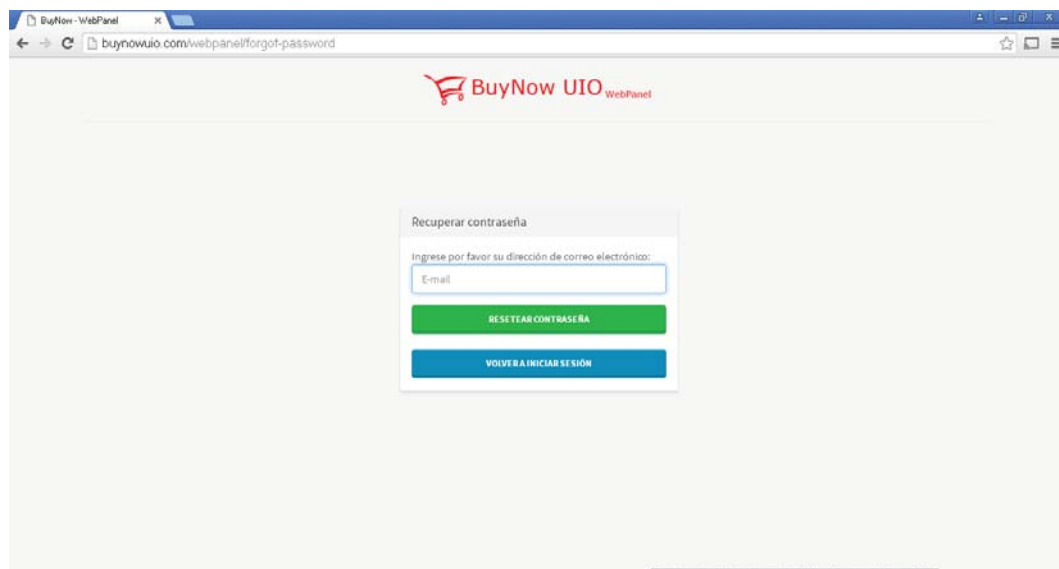


Figura 33. Pantalla para resetear contraseña

5.2.4 Sprint 5

Para el desarrollo del Sprint 4 se trabaja con todo lo relacionado con la creación de la aplicación móvil con sus respectivas clases y pantallas como se muestra en la tabla a continuación.

Tabla 7. Sprint 4

Id	Descripción	Criterio de Aceptación
BNUIOS9-1	Flujo de aplicación móvil	Crear el flujo de la aplicación móvil
BNUIOS9-2	Pantalla donde muestra la información acerca de la aplicación	Crear pantalla donde muestra la información acerca de la aplicación.
BNUIOS9-3	Pantalla donde muestra el historial de compras del usuario	Crear pantalla donde muestra el historial de compras del usuario.
BNUIOS9-4	Pantalla para el inicio de sesión del usuario	Crear pantalla para el inicio de sesión del usuario.
BNUIOS9-5	Pantalla principal	Crear pantalla principal donde muestra el mapa con sus tiendas por geo localización y el draw navigator con sus respectivas opciones por sesión de usuario.
BNUIOS9-6	Pantalla donde muestra la tienda con sus respectivos productos	Crear pantalla donde muestra la tienda con sus respectivos productos.
BNUIOS9-7	Pantalla que permite registrar usuarios	Crear pantalla que permite registrar usuarios.
BNUIOS9-8	Pantalla donde permite elegir la cantidad de productos a comprar	Crear pantalla donde permite elegir la cantidad de productos a comprar.
BNUIOS9-9	Pantalla de inicio o carga del sistema móvil	Crear pantalla de inicio o carga del sistema móvil.
BNUIOS9-10	Pantalla donde se genera la lista de los productos a elegir	Crear pantalla donde se genera la lista de los productos a elegir.
BNUIOS9-11	Pantalla utilizada en casos donde el usuario olvida su contraseña	Crear pantalla utilizada en casos donde el usuario olvida su contraseña

Desarrollo del Sprint

5.2.4.1 BNUIOS 9-1 Flujo de Aplicación Móvil

Se ha desarrollado la aplicación móvil en Android Studio utilizando la API 21 bajo el esquema de la figura a continuación.

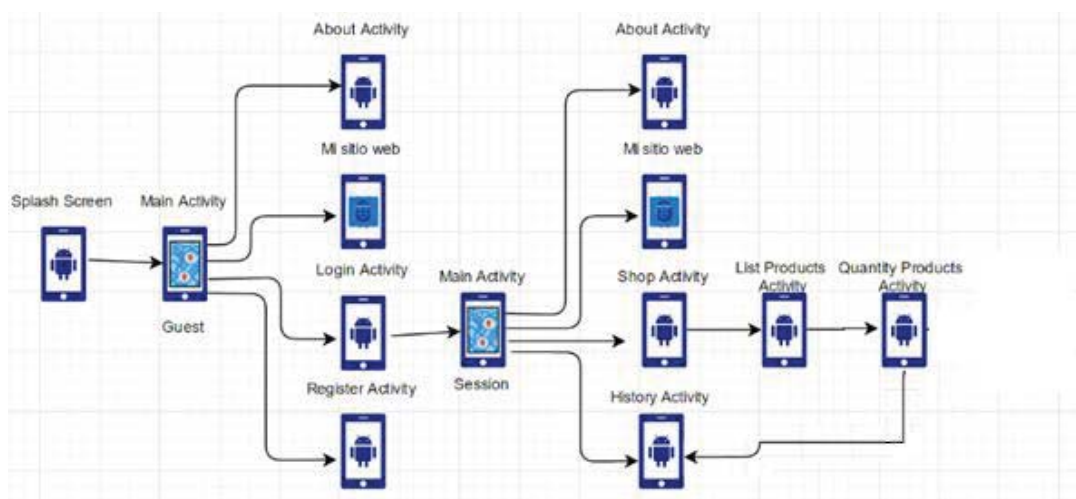


Figura 34. Diagrama de flujo Aplicación Móvil

Para el desarrollo de la aplicación previamente se generaron las Apis del lado del servidor, de las cuales se tomarán los distintos parámetros generados por las mismas. Posterior se procedió con la creación de los distintos layouts con el diseño adecuado para la aplicación; además se crearon las respectivas clases.

Para obtener el mapa por medio de la clave de Google maps se creó la clase onMapReady en el Activity principal donde también se encuentra el Navigation Drawer Activity el cual tiene su menú superior derecho, su header y su menú lateral izquierdo. Los nombres para cada uno de las partes del layout respectivamente son:

- @menu/Activity_main_navigation.xml
- Nav_header_activity_main_navigation.xml
- @menu/layout_main_navigation_drawer.xml

5.2.4.2 BUIOS 9-2 Pantalla de la información acerca de la aplicación

Para mostrar la información acerca de la aplicación se ha creado la clase Activity_about y la pantalla con el nombre Layout_about.xml; las mismas que se muestran en la figura a continuación.



Figura 35. Pantalla móvil con información de la aplicación

5.2.4.3 BNUIO 9-3 Pantalla del historial de compras del usuario

Para mostrar el historial, se ha creado la clase Activity_bHistory y su respectivo archivo de tipo XML llamado Layout_bHistory.xml mostrado en la figura 36.

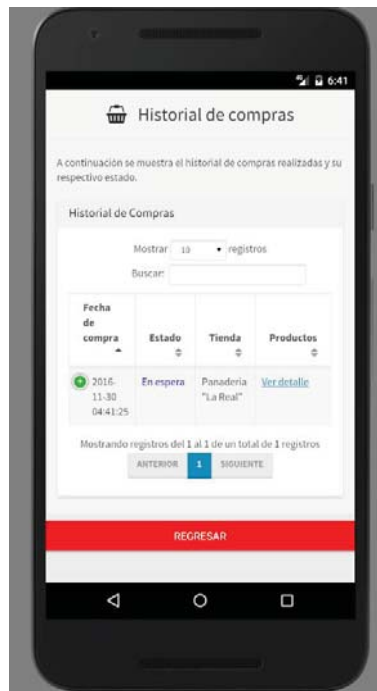


Figura 36. Pantalla móvil para el historial de compras

5.2.4.4 BNUIOS 9-4 Pantalla para el inicio de sesión del usuario.

Para el inicio de sesión se han creado la clase y layout con el nombre Activity_login y Layout_login.xml respectivamente mostrados en la figura a continuación.



Figura 37. Pantalla móvil para el inicio de sesión

5.2.4.5 BNUIOS 9-5 Pantalla principal

Dentro de la pantalla principal existe el menú de navegación y el mapa donde muestra la ubicación del cliente por geo localización. Para esta pantalla se han creado la clase `Activity_main_navigation` junto a su layout `Layout_main_navigation.xml` que se lo puede ver en la figura 38.

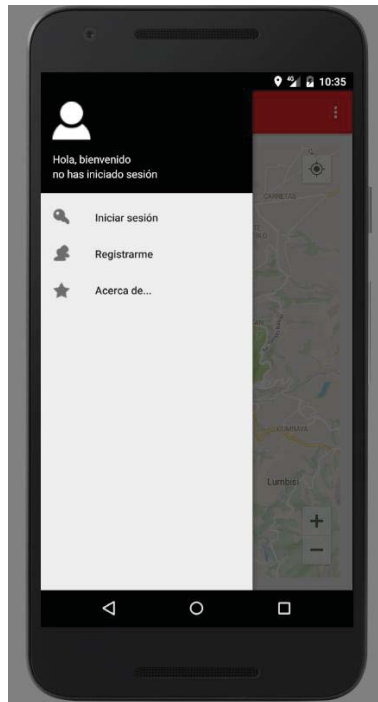


Figura 38. Pantalla móvil principal

5.2.4.6 BUIOS 9-6 Pantalla de tienda y sus productos

Para mostrar la tienda con sus respectivos productos se ha creado el layout `Layout_products.xml` junto a su clase `Activity_products`. Los mismos que se los pueden ver en la figura a continuación.

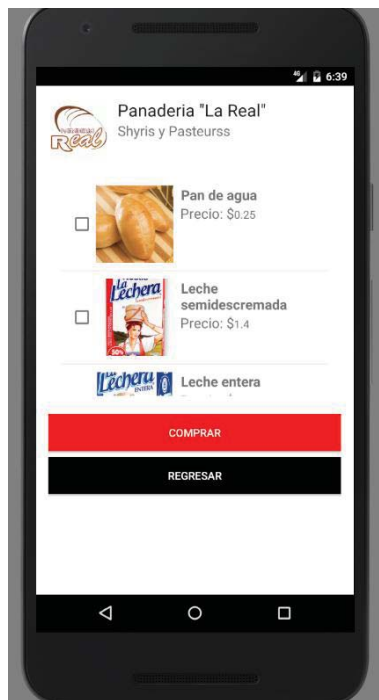


Figura 39. Pantalla móvil de la tienda con sus productos a escoger

5.2.4.7 BUIOS 9-7 Pantalla que permite registrar usuarios.

Para registrar usuarios se ha creado la clase y layout Activity_register y Layout_register.xml respectivamente. Los mismos que se reflejan en la figura a continuación.

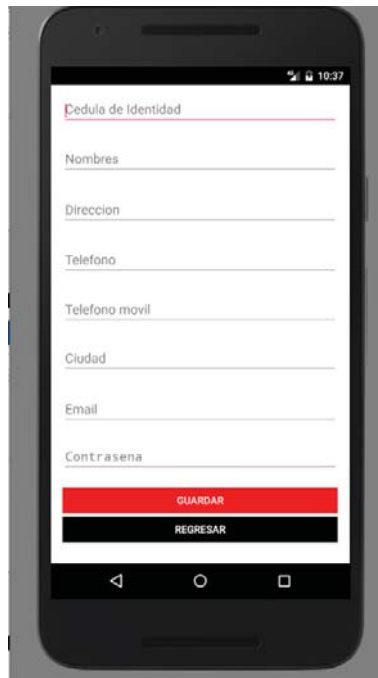


Figura 40. Pantalla móvil para registrar usuarios

5.2.4.8 BUIOS 9-8 Pantalla de elección de productos a comprar.

Para elegir los productos a comprar se han creado las clases Activity_shop y EditableProduct con sus respectivos layout Layout_shop.xml y Layout_products_to_buy.xml. Los mismos que se los pueden ver en la figura 41.

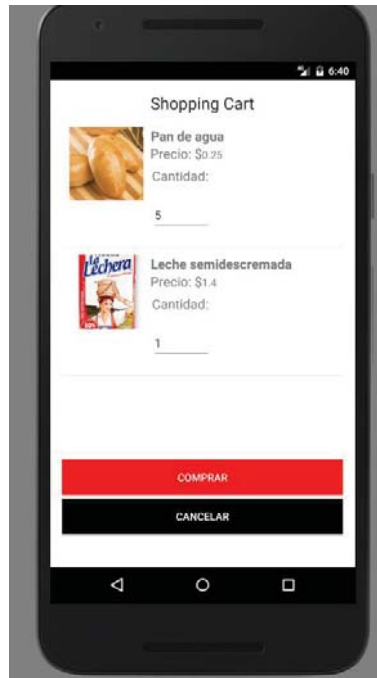


Figura 41. Pantalla móvil para seleccionar la cantidad de productos a elegir

5.2.4.9 BNUIOS 9-9 Pantalla de inicio o carga del sistema móvil

También conocida como splash screen, se ha creado la pantalla de inicio con su clase y layout `Activity_splash` y `Layout_splash.xml` respectivamente. La misma que se la puede ver en la figura a continuación.



Figura 42. Pantalla móvil de inicio

5.2.4.10 BNUIOS 9-10 Pantalla de la lista de productos a elegir

Para generar la lista de los productos a elegir desde la API se ha creado las clases:

- `Adapter_products_list`
- `SelectableProduct`
- `ProductItem`

Además se ha creado el layout: `Layout_products_listLayout_products.xml`

5.2.4.11 BNUIO 9-11 Pantalla para recordar contraseña

Para casos de emergencia donde el usuario se ha olvidado la contraseña se ha creado la clase `Activity_ForgotPassword` junto a su layout `Layout_forgotPassword.xml`. Los mismos que se pueden ver en la figura a continuación.



Figura 43. Pantalla móvil para resetear la contraseña

6. Conclusiones y Recomendaciones

6.1 Conclusiones

A continuación, se muestran las principales conclusiones que se obtuvieron a la culminación del presente Trabajo de Grado.

El uso de dispositivos móviles para la venta de productos en línea se ha vuelto muy popular, por eso existen muchas aplicaciones móviles trabajando. Esto se debe a que las personas el mayor tiempo usan estos dispositivos y al poder hacer sus quehaceres a través del mismo, resulta más sencillo su día a día.

Tanto los Smart ups como las personas buscan este tipo de aplicaciones para facilitar sus operaciones diarias, por eso es que BuyNowUIO es una alternativa.

El diseño de esta aplicación utilizando una arquitectura Cliente – Servidor, ayuda a que los datos introducidos por el cliente a través de BuyNowUIO puedan ser vistos por las tiendas y el administrador del sistema; de esta manera, se puedan ver estadísticas de venta y mejorar sus ingresos, al igual que considerar los productos que más stock deban tener.

Es factible incorporar temas de usabilidad dentro de la aplicación móvil debido a que mejora la satisfacción emitida por el usuario al usar la misma. Además, se aprovechan los recursos del dispositivo al máximo.

El uso de las API ayuda a trabajar del lado del sistema móvil al igual que web; debido a que se puede trabajar con parámetros.

Al trabajar con la API de cualquier proveedor de mensajería de texto, probar la funcionalidad dentro del país a utilizar.

6.2 Recomendaciones

Como trabajos futuros se recomienda probar con diferentes proveedores de mensajería de texto por tema de costos y funcionamiento dentro de cada país. Además, mejorar el diseño de la aplicación web. Una recomendación muy importante es realizar la aplicación móvil para los dueños de la tienda; es decir, poder vender el producto a cadenas de tiendas. Además, se recomienda la incorporación de medidas de seguridad con protocolo HTTPS con cifrado SSL.

Referencias

- Academia Android*. (2014). Recuperado el 18 de Noviembre de 2016, de <http://academiaandroid.com/android-studio-v1-caracteristicas-comparativa-eclipse/>
- Betancourt Granillo, J. O., & Sánchez Anguiano, F. (2010). PROYECTO PARA LA MEJORA DE LA COMUNICACION Y RELACIONES INTERNAS EN UNA EMPRESA DE COMERCIO ELECTRONICO. Recuperado el 15 de Noviembre de 2016, de <http://tesis.ipn.mx/handle/123456789/5816>
- González, E. (s.f.). *Aprende a Programar*. Recuperado el 18 de Noviembre de 2016, de http://aprenderaprogramar.com/index.php?option=com_content&view=article&id=590:i-que-es-y-para-que-sirve-javascript-embeber-javascript-en-html-ejercicio-ejemplo-basico-cu00731b&catid=69:tutorial-basico-programador-web-html-desde-cero&Itemid=192
- Google. (2016). *Google Developers*. Recuperado el 18 de Noviembre de 2016, de <https://developers.google.com/maps/documentation/android-api/signup>
- Hernández, E. (2002). *Departamento de Informática de Sistemas y Computadores, Universitat Politècnica de Valencia*. Recuperado el 15 de Noviembre de 2016, de <http://www.disca.upv.es/enheror/pdf/ActaUML.pdf>
- Json.org. (s.f.). *Introducción a JSON*. Recuperado el 15 de Noviembre de 2016, de <http://www.json.org/json-es.html>
- Lockhart, J., Smith, A., & Allen, R. (s.f.). *Slim*. Recuperado el 18 de Noviembre de 2016, de <http://www.slimframework.com/docs/>

- Merino, M. (2014). *¿Qué es una API y para qué sirve?: TICbeat*. Recuperado el 16 de Noviembre de 2016, de <http://www.ticbeat.com/tecnologias/que-es-una-api-para-que-sirve/>
- proyectosagiles.org. (s.f.). *Qué es Scrum*. Recuperado el 18 de Noviembre de 2016, de <https://proyectosagiles.org/que-es-scrum/>
- Reenskaug, T., & Colien, J. (2009). *The DCI Architecture: A New Vision of Object-Oriented Programming*. Recuperado el 16 de Noviembre de 2016, de http://www.artima.com/articles/dci_vision.html
- Revelo, J. (2015). *Hermos Programación*. Recuperado el 18 de Noviembre de 2016, de <http://www.hermosaprogramacion.com/2015/02/android-volley-peticiones-http/>
- Rouse, M. (s.f.). *Search Datacenter*. Recuperado el 18 de Noviembre de 2016, de <http://searchdatacenter.techtarget.com/es/definicion/MySQL>
- Solis, J. (2014). *Arweb*. Recuperado el 18 de Noviembre de 2016, de <http://www.arweb.com/chucherias/editorial/%C2%BFque-es-bootstrap-y-como-funciona-en-el-diseno-web.htm>
- Tanenbaum, A. (2008). *Sistemas Distribuidos*. Pearson Education. Recuperado el 18 de Noviembre de 2016, de http://www.academia.edu/21420808/SISTEMAS_DISTRIBUIDOS_SISTEMAS_DISTRIBUIDOS_Principios_y_Paradigmas
- The Php Group. (s.f.). *php*. Recuperado el 18 de Noviembre de 2016, de <http://php.net/manual/es/intro-what-is.php>
- Toapanta, M. (2012). MÉTODO ÁGIL SCRUM, APLICADO A LA IMPLANTACIÓN DE UN SISTEMA INFORMÁTICO PARA EL PROCESO DE RECOLECCIÓN MASIVA DE INFORMACIÓN CON TECNOLOGÍA MÓVIL. Recuperado el 15 de Noviembre de 2016, de

<http://repositorio.espe.edu.ec:8080/jspui/bitstream/21000/5893/1/T-ESPE-034427.pdf>

Varas, C. (2010). *Marketing Móvil*. Recuperado el 19 de Noviembre de 2016, de <http://imgbiblio.vaneduc.edu.ar/fulltext/files/TC098561.pdf>

Anexos

ANEXO 1: APIs

Función	URL API	Método	Parámetros
CUser_Validate	/api/user/validate/	GET	documentID. - Documento de identidad
			email. - Correo electrónico
CUser_Create	/api/user/create	POST	password. - Contraseña del usuario
			documentID. - Documento de identidad
			email. - Correo electrónico
			address. - Dirección de domicilio
			phone. - Teléfono de domicilio
			mobilephone. - Teléfono Celular
			city. - Ciudad de domicilio
CUser_ValidateLogin	/api/user/login	POST	email. - Correo electrónico
			password. - Contraseña del usuario
CUser_Update	/api/user/update	POST	user_id. - Código único de usuario
			token. - Indicador de sesión de usuario único
			name. - Nombres del Usuario
			address. - Dirección de domicilio
			phone. - Teléfono de domicilio
			mobilephone. - Teléfono Celular
			city. - Ciudad de domicilio
CUser_ChangePassword	/api/user/login/change		user_id. - Código único de usuario
			token. - Indicador de sesión de usuario único
			oldpassword. - Contraseña actual

			a cambiar
			newpassword. - Nueva contraseña
CUser_Logout	/api/user/logout	POST	user_id. - Código único de usuario
			token. - Indicador de sesión de usuario único
CUser_Register	/api/user/register	POST	user_id. - Código único de usuario
			token. - Indicador de sesión de usuario único
			name. - Nombres del Usuario
			documentID. - Documento de identidad
			email. - Correo electrónico
			address. - Dirección de domicilio
			phone. - Teléfono de domicilio
			mobilephone. - Teléfono Celular
			city. - Ciudad de domicilio
			rank. - Tipo de Usuario
CUser_Get	/api/user/get	POST	user_id. - Código único de usuario
			token. - Indicador de sesión de usuario único
			filter. - Verifica si es email ò cédula
			value. - Valor del email ò cédula
CUser_Edit	/api/user/edit	POST	user_id. - Código único de usuario
			user_edit_id. - Código del usuario a editar
			token. - Indicador de sesión de usuario único
			name. - Nombres del Usuario
			address. - Dirección de domicilio
			phone. - Teléfono de domicilio

			mobilephone. - Teléfono Celular
			city. - Ciudad de domicilio
CUser_Delete	/api/user/delete	POST	user_id. - Código único de usuario
			user_delete_id. - Código del usuario a eliminar
			token. - Indicador de sesión de usuario único
CMarket_GetList	/api/market/list	GET	Ninguno
CMarket_Create	/api/market/create	POST	user_id. - Código único de usuario
			token. - Indicador de sesión de usuario único
			name. - Nombre de la Tienda
			logo_url. - Logo de la Tienda
			address. - Dirección de la tienda
			phone. - Teléfono de la tienda
			mobilephone. - Teléfono Celular de la tienda
			geo_lng. - Longitud de la ubicación de la tienda
			geo_lat.- Latitud de la tienda
CMarket_Get	/api/market/get	POST	user_id. - Código único de usuario
			token. - Indicador de sesión de usuario único
			market_id. - Código único de la tienda
CMarket_Edit	/api/market/edit	POST	user_id. - Código único de usuario
			token. - Indicador de sesión de usuario único
			market_id. - Código único de la tienda

			logo_url. - Logo de la Tienda
			address. - Dirección de la tienda
			phone. - Teléfono de la tienda
			mobilephone. - Teléfono Celular de la tienda
			geo_lng. - Longitud de la ubicación de la tienda
			geo_lat.- Latitud de la tienda
CMarket_Delete	/api/market/delete	POST	user_id. - Código único de usuario
			market_delete_id. - Código de tienda a eliminar
			token. - Indicador de sesión de usuario único
CProduct_GetList	/api/product/list	POST	market_id. - Código único de la tienda
CProduct_Create	/api/product/create	POST	user_id. - Código único de usuario
			market_id. - Código único de la tienda
			name. - Nombre del Producto
			description. - Descripción del Producto
			mark. - Marca del Producto
			price. - Precio del Producto
			image_url. - Logo del producto
			stock. - Stock del producto
CProduct_Get	/api/product/get	POST	user_id. - Código único de usuario
			token. - Indicador de sesión de usuario único
			product_id. - Código Único de Producto
CProduct_Edit	/api/product/edit	POST	user_id. - Código único de usuario

Edit			token. - Indicador de sesión de usuario único
			product_id. - Código Único de Producto
			description. - Descripción del Producto
			mark. - Marca del Producto
			price. - Precio del Producto
			image_url. - Logo del producto
			stock. - Stock del producto
CProduct_Delete	/api/product/delete	POST	user_id. - Código único de usuario
			product_delete_id. - Código de usuario a eliminar
			token. - Indicador de sesión de usuario único
CBuy_Get List	/api/buy/list	POST	user_id. - Código único de usuario
			token. - Indicador de sesión de usuario único
CBuy_Get Requested List	/api/buy/requests	POST	user_id. - Código único de usuario
			token. - Indicador de sesión de usuario único
CBuy_Do	/api/buy/do	POST	user_id. - Código único de usuario
			token. - Indicador de sesión de usuario único
			market_id. - Código Único de Tienda
			units. - Unidades de Producto
CBuy_Deliver	/api/buy/deliver	POST	user_id. - Código único de usuario
			token. - Indicador de sesión de usuario único
			buy_id. - Código único de compra

			comments. - Comentarios de la compra
CBuy_Refuse	/api/buy/refuse	POST	user_id. - Código único de usuario
			token. - Indicador de sesión de usuario único
			buy_id. - Código único de compra
			comments. - Comentarios de la compra

