

# **Aplicación móvil “TecnoWilly”**

**Juan Sebastian Marin Rios  
Noviembre de 2019.**

**Corporación Universitaria Minuto de Dios Uniminuto Sede Bello.  
Tecnología en Informática  
Proyecto de Grado**

## Resumen

La empresa TecnoWilly realiza la presentación de sus productos por medio de documentos en Word y pdf, durante el proceso cada usuario interesado en un producto debe solicitar la cotización por medio telefónico y esperar al envío de la información; de acuerdo con los pasos agigantados que da la comunidad digital actualmente este proceso manual ocasiona una serie de reprocesos y reduce la satisfacción del cliente por los tiempos de espera. Teniendo como base lo antes descrito el desarrollo del proyecto móvil para tecnowilly brinda un acercamiento más ágil y con una experiencia de usuario mucho más amena al momento de presentar los productos y servicios, adicionalmente el catálogo de productos esta actualizado constantemente y las personas que realicen uso de la aplicación móvil pueden solicitar información del producto y/o servicio a través de la plataforma móvil y ser contactados por medio telefónico por el dueño de la compañía para llegar un acuerdo en la venta del bien y/o servicio.

Con la implementación de esta mejora en la experiencia de usuario Tecnowilly espera lograr la fidelización de muchos más clientes en la ciudad de Medellín y a futuro lograr la expansión de su catálogo de productos y servicios para brindar mayor satisfacción al consumidor final.

**Palabras clave:** Transformación Digital, innovación, Agilísimo, Desarrollo de software, Aplicación móvil, Experiencia de Usuario, TecnoWilly.

## **Abstract**

The TecnoWilly company makes the presentation of its products through documents in Word and pdf, during the process each user interested in a product must request a quote by telephone and wait for the information to be sent; According to the leaps and bounds that the digital community is currently taking, this manual process causes a series of reprocesses and reduces customer satisfaction due to waiting times. Based on the aforementioned, the development of the mobile project for tecnowilly provides a more agile approach and with a much more enjoyable user experience when presenting the products and services, additionally the product catalog is constantly updated and the people who make use the mobile application can request product and / or service information through the mobile platform and be contacted by telephone by the owner of the company to reach an agreement on the sale of the good and / or service.

With the implementation of this improvement in the user experience Tecnowilly hopes to achieve the loyalty of many more customers in the city of Medellin and in the future achieve the expansion of its catalog of products and services to provide greater satisfaction to the final consumer.

**Keywords:** Digital Transformation, innovation, Very Fast, Software Development, Mobile Application, User Experience. TecnoWilly.

## **Introducción**

El presente proyecto tiene como finalidad la realización de una aplicación móvil para la comercialización de un catálogo de productos de la compañía TecnoWilly. La idea surge como un punto de mejora en la experiencia de usuario para presentar productos y servicios actualizados los cuales podrán ser consultados desde sus dispositivos móviles.

Según un estudio de la Asociación de la Industria Móvil de Colombia (Asomóvil) realizado en el año 2017 revela que los colombianos utilizan en promedio cuatro horas al día el celular y unas 3,4 horas de su tiempo diario para navegar.

Teniendo como base la investigación realizada por Asomóvil antes descrita, se puede tomar como un antecedente positivo para el cambio o transformación que la empresa desea brindar ya que los usuarios podrán consultar los productos y servicios en cualquier lugar y en cualquier momento de su vida cotidiana sin complicaciones y TecnoWilly tiene la oportunidad de ser parte de los teléfonos inteligentes de los ciudadanos como estrategia de negocio.

Este documento está estructurado en los siguientes capítulos, el primero: contiene la descripción problema, antecedentes, justificación, objetivos y alcance del proyecto. En el segundo se consignan los referentes teóricos que dan sustento al proyecto. El tercero abarca el desarrollo del proyecto, en su parte metodológica. En el cuarto se describen los resultados obtenidos. Finalmente se registran las conclusiones y recomendaciones del trabajo de grado.

## Tabla de contenido

Capítulo 1. Descripción del proyecto.....	8
Planteamiento el problema.....	8
Antecedentes .....	9
Justificación .....	12
Objetivos .....	13
Objetivo general.....	13
Alcance .....	13
Capítulo 2. Marco Teórico .....	14
Capítulo 3. Desarrollo de la propuesta .....	19
Metodología.....	19
Cronograma .....	21
Presupuesto .....	23
Capítulo 4. Resultados .....	25
Capítulo 5. Conclusiones y Recomendaciones .....	49
Lista de referencias.....	51

## Lista de tablas

Tabla 1 Cronograma.....	23
Tabla 2 Presupuesto .....	24
Tabla 3 Historia de Usuario Administración Web.....	25
Tabla 4 Historia de Usuario Gestión de Usuarios .....	26
Tabla 5 Historia de Usuario Maestro Categorías .....	27
Tabla 6 historia de usuario Catálogo de Productos .....	28

## Lista de figuras

Ilustración 1 Comparativo Alibaba y AliExpress (Ramirez, 2018).....	10
Ilustración 2 Uso de Aplicaciones (Iqbal, 2019).....	11
Ilustración 3: Diagrama MVC. tomado de (Alvarez, 2014) .....	18
Ilustración 4 Ilustración Controllers .....	28
Ilustración 5 Models .....	29
Ilustración 6 Ejemplo relación modelos.....	29
Ilustración 7 resources .....	30
Ilustración 8 migrations .....	30
Ilustración 9 ejemplo tabla desde migrate .....	31
Ilustración 10 archivo conf. Global.....	31
Ilustración 11 - login .....	32
Ilustración 12 Menú de navegación .....	32
Ilustración 13 datatable catálogo de productos.....	33
Ilustración 14 Creación de Producto .....	33
Ilustración 15 Notificaciones SweetAlert.....	34
Ilustración 16 Creación Categoría .....	34
Ilustración 17 creación usuario .....	35
Ilustración 18 Funcion en ProductController – API.....	36
Ilustración 19 relaciones necesarias modelo de productos.....	36
Ilustración 20 ruta API. ....	36
Ilustración 21 ruta API postman .....	37
Ilustración 22 array información producto .....	37
Ilustración 23 array imagen de producto.....	37

## **Capítulo 1. Descripción del proyecto**

### **Planteamiento el problema**

Teniendo en cuenta que las empresas cada vez generan pasos más agigantados para los procesos de transformación digital y mejora de sus modelos de negocios con el propósito de generar un cambio de estrategia, el acercamiento con los clientes se hace cada vez más necesario.

Según el estudio *¿Is Your Company Ready for a Digital Future?* publicado por el Instituto de Tecnología de Massachusetts en la revista “Winter” (2017), los líderes de las compañías deben elegir el camino correcto para llevar a sus compañías a otro nivel y prepararse para el futuro.

Becoming future ready requires changing the enterprise on two dimensions: customer experience and operational efficiency. (Woerner, 2017).

Para TecnoWilly, un negocio que presta servicios de Tecnología a domicilio a los clientes se identificó un punto de mejora el cual consiste en cambiar la experiencia de usuario y el acercamiento con los clientes por medio de una aplicación móvil la cual tendrá en sus inicios cabida solo en la ciudad de Medellín. La idea de realizar una mejora en la experiencia de usuario se presenta con el fin de mejorar la interacción de los usuarios con el catálogo de productos de TecnoWilly el cual es ofrecido actualmente en formato pdf. Un cambio en la usabilidad de los recursos podrá generar una experiencia positiva y una atracción de nuevos clientes hacia los productos y/o servicios que esta ofrece.



## **Antecedentes**

En los últimos años, la tecnología de los dispositivos móviles ha ido mejorando de forma muy acelerada, las empresas lanzan al mercado nuevos modelos, con tecnologías cada vez más novedosas y a la par el amplio mundo de las aplicaciones acompañan este crecimiento pudiendo realizar cada vez más acciones de la vida diaria en los dispositivos, esto abre de forma considerable las posibilidades de su desarrollo para aprovechar al máximo las características de los dispositivos. (Díaz, 2015).

A nivel Internacional podemos tomar como ejemplo el caso de éxito del grupo de empresas de origen chino Alibaba. En sus inicios Alibaba estaba enfocada solamente al sector B2B. Su propósito era conectar las empresas chinas con empresas del exterior en un marketplace ágil. Poco a poco se posicionó como un espacio seguro para hacer transacciones gracias a su sistema de puntos y evaluaciones. Atrajo la atención de muchas empresas preocupadas por el largo historial de irregularidades en el ámbito empresarial chino.

Su plataforma de comercio B2B para pequeñas empresas ha ayudado a conectar a los compradores de América del Norte y Europa con proveedores y fabricantes de China. La compañía ha puesto en marcha una serie de otros negocios conexos, como portales web en línea de transacciones B2B, ventas minoristas y servicios de pago, un motor de búsqueda de compras y servicios de computación en nube centrada en datos que, en conjunto, impulsan el 80% de todo el comercio en línea en China. (Gómez)

Otro aporte internacional en empresas B2C es Aliexpress, la cual es una compañía fundada en el año 2009 con el objetivo de abrir un mercado que ofrezca productos a precios de fábrica directos de China a los consumidores.

Las órdenes al comprar en AliExpress pueden ser tan pequeñas como un solo producto o tan grandes como cientos del mismo producto. (pymex.com)

El siguiente es un cuadro comparativo con las principales diferencias entre los portales de venta online Alibaba.com y AliExpress en el cual se podrán identificar algunas similitudes y diferencias.

	ALIBABA.COM	ALIEXPRESS
FUNDADO EN	1999	2010
PARTE DE	Grupo Alibaba	Grupo Alibaba
PRODUCTOS A LA VENTA	De todo tipo	De todo tipo
TIPO DE VENTA	Al por mayor	Al consumidor final
PROCESO DE COMPRA	Primero contacto, después compra	Compra directa
VENDEDORES	Empresas	Empresas y particulares

*Ilustración 1 Comparativo Alibaba y AliExpress (Ramirez, 2018)*

Según el sitio web *BusinesofApps*, Colombia se encuentra como el tercer país de Latinoamérica que más uso realiza de aplicaciones móviles especialmente las redes sociales. Este estudio revela que los niveles más altos de uso diario se informan en los mercados de Asia, América del Sur y África. Para Colombia la cifra promedio de uso diario es de 3 horas 31 minutos.

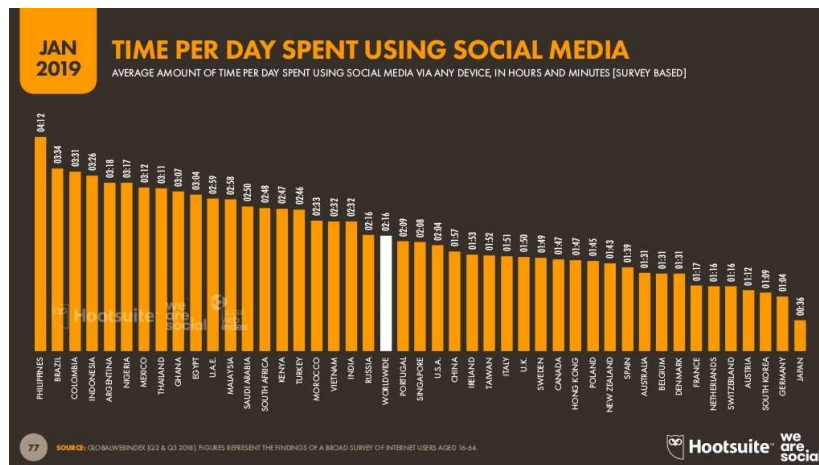


Ilustración 2 Uso de Aplicaciones (Iqbal, 2019)

Un ejemplo importante de aportes con crecimiento en Colombia por medio de sitios de comercio electrónico lo tiene Mercado Libre, el cual es un Marketplace dedicado a las compras, ventas y pagos por internet. Según el sitio web dataiFX.com, los ingresos de mercado libre en Colombia aumentaron un 58% en el año 2018.

En Colombia la compañía tuvo un crecimiento exponencial durante 2018 que se refleja en el aumento de las cifras en todo su ecosistema. Se reportaron 22 millones de artículos publicados lo que representó un aumento del 300%, así mismo los ingresos de la compañía aumentaron en un 58%, superando las expectativas de la industria. La cantidad usuarios registrados en la plataforma creció en un 30% respecto al año anterior, los envíos a través de la unidad Mercado Envíos crecieron en un 90% gracias a la alianza que tiene con los operadores logísticos más importantes del país. Las tiendas oficiales crecieron en un 100% dentro del marketplace lo que apalanca la estrategia de democratización del comercio electrónico en Colombia. (dataiFX, 2019)

Según el sitio web pulsosocial, la aplicación móvil de mercado libre es actualmente una de las apps más utilizadas en Latinoamérica.

Actualmente, en Colombia el tráfico en la plataforma vía smartphones, tablets y otros dispositivos móviles representan el 70%, siendo este el medio a través del cual se realiza la mayoría de las compras de Mercado Libre. Los países líderes en esta materia son Argentina con un 15% del total de descargas, Brasil y México. (pulsosocial, 2018)

*“Para Colombia este hito representa avances significativos en materia de comercio electrónico y nuevos hábitos de consumo que se reflejan en coyunturas específicas en las que el protagonista es este nuevo canal de compras y ventas, y en donde priman los beneficios de las herramientas móviles que ponemos a disposición de los usuarios para favorecer sus necesidades”,* afirma Andrés Robotél, Gerente de Market Place de Mercado Libre Colombia.

## **Justificación**

Con el desarrollo de una aplicación móvil se espera impactar positivamente en los usuarios, ofreciendo un catálogo de productos y asequible en el momento que se desee. Es por esto, que la empresa TecnoWilly desea innovar y transformar la forma en la que ofrece sus servicios, optimizar los tiempos y recursos para dar solución a la necesidad planteada aprovechando que “La gente tiene una gran cercanía emocional con las telecomunicaciones, específicamente con el celular” tal como lo manifestó Nancy Patricia Gutiérrez, durante su periodo en la presidencia de Asomóvil (TECNÓSFERA, 2017)

Al ser la aplicación compatible con la mayoría de versiones Android del mercado se podrá llegar a muchos más usuarios en el mercado de la ciudad de Medellín como punto inicial con una interfaz de usuario sencilla, atractiva y con productos de calidad.

## **Objetivos**

### **Objetivo general**

Desarrollar aplicación móvil Android para la publicación de productos y servicios de la Empresa TecnoWilly de manera que los clientes puedan solicitarlos de forma simple y eficiente contando con información actualizada.

### **Objetivos específicos**

1. Diseñar las interfaces de la aplicación móvil con una experiencia de usuario agradable y de fácil uso.
2. Crear un servicio API REST con laravel para la actualización continua de la información en la interfaz de catálogo de productos.
3. Subir la Aplicación a la Play Store donde podrá ser descargada sin costo alguno.

## **Alcance**

El alcance de la propuesta se ciñe a los requerimientos planteados por el cliente los cuales estarán consagrados en las historias de usuario con los requisitos de información.

El desarrollo comprende desde la construcción de un sitio web para la creación de categorías y de productos, hasta la implementación de una aplicación móvil Android con servicios Api en laravel para la integración de la página web con el catálogo de productos que será ofertado a los clientes por medio de la app.

La aplicación móvil no contará con integración con medios de pago, pero tendrá la opción de solicitar información por medio de correo electrónico integrado con las apps del dispositivo móvil.

Por último, la aplicación móvil completamente funcional será desplegada en la Play Store.

## **Capítulo 2. Marco Teórico**

Para llevar a cabo el proyecto es necesario tener claridad de algunos conceptos como lo son el desarrollo en arquitectura MVC la cual será llevada a cabo con el framework Laravel, programación para móviles con lenguaje Kotlin, estructura de bases de datos relacional y sincronización por medio de APIRest entre bases de datos Mysql y aplicaciones Android.

### **¿Qué es una App?**

Una Aplicación móvil o App es un programa que se instala en el Smartphone o Tablet del cliente. De esta manera la App pasa a formar parte del celular, un lugar privilegiado en la que toda marca querría estar. El ícono que la identifica, con la imagen de su empresa, estará visible en todo momento, recordando al usuario nuestra existencia, tanto cuando la vaya a usar como cuando esté realizando otra tarea.

Recuerde que cuando el usuario está trabajando, mira el celular, cuando está aburrido, mira el celular, cuando está viajando, mira el celular, cuando está comunicándose, mira el celular, cuando se está divirtiendo, mira el celular, cuando lee el correo, consulta la agenda, etc., mira el celular. (Gretter, s.f.)

### **¿Qué es un Call to Action (CTA) o llamada a la acción?**

Según Viñaras (s.f), lo que se denomina CTA, call to action o llamada a la acción no es más que un reclamo para que los visitantes a nuestro sitio realicen una acción en concreto. Su principal característica es que tiene forma de botón o de enlace y un texto que anima al usuario a hacer clic. Por lo general, este texto contiene términos en modo infinitivo o imperativo ("enviar",

"apúntate ya", "descarga el ebook"), aunque no siempre es así, ya que a veces se recurre a soluciones creativas como "Yo me apunto".

Otra característica que suele identificar a los call to action es que presentan un diseño que les hace destacar dentro de la página, por ejemplo, un botón en relieve con un color que genere contraste.

Según Cortés (s.f), Para implementar un buen CTA se debe realiza un estudio previo del público al que estará dirigida la aplicación, saber quiénes serán los posibles clientes, la percepción que se quiere brindar y las formas más efectivas para dirigirse a dicho público. Es importante tener en cuenta que un *call to action* está desarrollado con el propósito que sea el cliente quien busque esa ayuda, información u oferta que se desea presentar, incitarlos a dar clic y saber más acerca de los servicios que son ofrecidos.

### **¿En qué consiste la transformación digital?**

La transformación digital se puede definir como la integración de las nuevas tecnologías en todas las áreas de una empresa para cambiar su forma de funcionar. El objetivo es optimizar los procesos, mejorar su competitividad y ofrecer un nuevo valor añadido a sus clientes. (Arana, s.f.). Según Rouse (2015), la transformación digital es la reinención de una organización a través de la utilización de la tecnología digital para mejorar la forma en que la organización se desempeña y sirve a quienes la constituyen. Digital se refiere al uso de la tecnología que genera, almacena y procesa los datos.

Según Salesforce “La transformación digital es el uso de la tecnología para mejorar radicalmente el desempeño y el alcance de las empresas”. (coresystems, 2018)

## **El mundo de las Aplicaciones Móviles en la Transformación Digital**

La mayoría de los usuarios ya han realizado su transformación digital de forma personal y para ellos es habitual moverse en el Internet y dentro de las Apps, están acostumbrados a utilizar Aplicaciones Móviles para sus tareas relacionadas con los negocios: banca, salud, información, compras, etc. Es por eso que la empresa debe estar allí para ofrecer una experiencia digital, a la que están acostumbrados los clientes hoy día. (coresystems, 2018)

### **¿Por qué toda empresa debería tener una app?**

De acuerdo con el sitio web Emprende hoy (2017) “Los propietarios de pequeñas empresas a menudo creen (erróneamente) que una app móvil sólo vale la pena para las grandes empresas o para ciertos rubros. Esto es falso. Casi cualquier empresa, independientemente de la industria o el tamaño, puede beneficiarse de una aplicación móvil como complemento a la estrategia digital que desarrolla en la web y redes sociales.

Según Campos (2016), las razones para crear una aplicación móvil para las empresas son las siguientes:

- 1- Comprometerse con sus clientes: Las Apps eficaces, significan clientes contentos y comprometidos, que por lo general se traducen en clientes que compran.
- 2- Promover productos y servicios: La investigación de mercados ha demostrado que los clientes perciben a las empresas que ofrecen aplicaciones móviles de forma más positiva que los que no lo hacen. Una App puede hacer que el negocio se destaque, y ayude a construir relaciones con los clientes.
- 3- Maximizar el retorno de inversión (ROI)
- 4- Ampliar la base de clientes.

Toda empresa puede beneficiarse de una aplicación móvil como complemento a la estrategia digital.



## ¿Qué es Android Studio?

Android Studio es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de aplicaciones para Android y se basa en IntelliJ IDEA . Además del potente editor de códigos y las herramientas para desarrolladores de IntelliJ, Android Studio ofrece aún más funciones que aumentan la productividad durante la compilación de apps para Android. (Developers, s.f.)

## ¿Qué es Kotlin?

Kotlin es un lenguaje de programación estáticamente tipado, es decir, es la máquina virtual la que infiere el tipo a las variables y, por tanto, no hay que especificarlas. Es un lenguaje que corre bajo la Máquina Virtual de Java, por lo que tiene el mismo rendimiento que Java, sin penalizar en ningún momento en este aspecto.

Está diseñado para interoperar con Java, por lo tanto, se pueden tener módulos programados en Java y otros módulos desarrollados en Kotlin. Estos módulos se podrían comunicar perfectamente sin problemas, simplemente tendrían que salvar el escalón que existe entre el desarrollo en Java y las premisas que tiene el desarrollo en Kotlin. (Jurado, 2018).

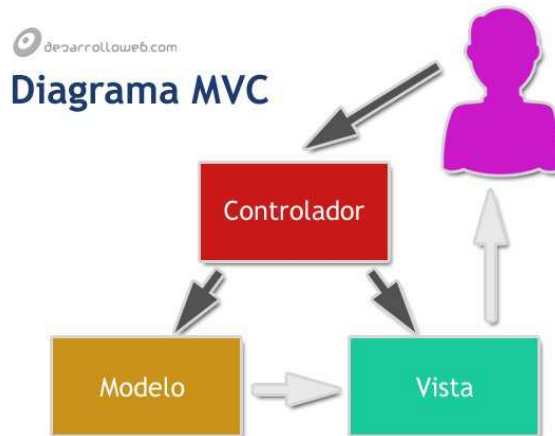
## Patrón de Arquitectura MVC

MVC es una propuesta de diseño de software utilizada para implementar sistemas donde se requiere el uso de interfaces de usuario. Surge de la necesidad de crear software más robusto con un ciclo de vida más adecuado, donde se potencie la facilidad de mantenimiento, reutilización del código y la separación de *concerns*. Su fundamento es la separación del código en tres capas diferentes, acotadas por su responsabilidad, en lo que se llaman Modelos, Vistas y Controladores.

MVC es un "invento" que ya tiene varias décadas y fue presentado incluso antes de la aparición de la Web. No obstante, en los últimos años ha ganado mucha fuerza y seguidores

gracias a la aparición de numerosos frameworks de desarrollo web que utilizan el patrón MVC como modelo para la arquitectura de las aplicaciones web. (Alvarez, 2014)

Ilustración 3: Diagrama MVC. tomado de (Alvarez, 2014)



## Elementos del patrón de arquitectura MVC:

### Modelos

Es la capa donde se trabaja con los datos, por tanto contendrá mecanismos para acceder a la información y también para actualizar su estado. Los datos se tienen habitualmente en una base de datos, por lo que en los modelos se tendrán todas las funciones que accederán a las tablas y harán los correspondientes selects, updates, inserts, etc. (Alvarez, 2014)

### Vistas

Las vistas, contienen el código de la aplicación que va a producir la visualización de las interfaces de usuario, o sea, el código que permitirá *renderizar* los estados de la aplicación en HTML. En las vistas solo están códigos que permiten mostrar la salida. (Alvarez, 2014)

## **Controladores**

Contiene el código necesario para responder a las acciones que se solicitan en la aplicación, como visualizar un elemento, realizar una compra, una búsqueda de información, etc.

Es una capa que sirve de enlace entre las vistas y los modelos, respondiendo a los mecanismos que puedan requerirse para implementar las necesidades de la aplicación. Sin embargo, su responsabilidad no es manipular directamente datos, ni mostrar ningún tipo de salida, sino servir de enlace entre los modelos y las vistas para implementar las diversas necesidades del desarrollo. (Alvarez, 2014)

## **Capítulo 3. Desarrollo de la propuesta**

### **Metodología**

Para el desarrollo de la propuesta se utilizará la metodología ágil ScrumBan., la cual unifica la gestión ágil de Scrum con la facilidad de comprensión de los flujos de trabajo de Kanban. El proyecto está compuesto por 11 iteraciones y una duración total de 16 semanas discriminados por cada uno de los procesos de la aplicación.

Así pues, Scrumban permite planificar los proyectos según se van ejecutando. Se marcan unos hitos o momentos clave, al final de los denominados sprints y cuando se llega a esa fecha inamovible, se revisa el trabajo hecho y se planifica el siguiente sprint o la denominada también iteración, según los nuevos requerimientos o cambios producidos. (Sinnaps, s.f.)

Las metodologías ágiles cuentan con un manifiesto ágil compuesto por 4 valores y 12 principios, así:

## **Valores**

- 1- Valoramos más a los individuos y su interacción que a los procesos y las herramientas.
- 2- Valoramos más el software que funciona que la documentación exhaustiva.
- 3- Valoramos más la colaboración con el cliente que la negociación contractual.
- 4- Valoramos más la respuesta al cambio que el seguimiento de un plan

## **12 Principios**

- 1- Nuestra principal prioridad es satisfacer al cliente a través de la entrega temprana y continua de software de valor.
- 2- Son bienvenidos los requisitos cambiantes, incluso si llegan tarde al desarrollo. Los procesos ágiles se dobligan al cambio como ventaja competitiva para el cliente.
- 3- Entregar con frecuencia software que funcione, en periodos de un par de semanas hasta un par de meses, con preferencia en los periodos breves.
- 4- Las personas del negocio y los desarrolladores deben trabajar juntos de forma cotidiana a través del proyecto.
- 5- Construcción de proyectos en torno a individuos motivados, dándoles la oportunidad y el respaldo que necesitan y procurándoles confianza para que realicen la tarea.
- 6- La forma más eficiente y efectiva de comunicar información de ida y vuelta dentro de un equipo de desarrollo es mediante la conversación cara a cara.
- 7- El software que funciona es la principal medida del progreso.
- 8- Los procesos ágiles promueven el desarrollo sostenido. Los patrocinadores, desarrolladores y usuarios deben mantener un ritmo constante de forma indefinida.
- 9- La atención continua a la excelencia técnica enaltece la agilidad.
- 10- La simplicidad como arte de maximizar la cantidad de trabajo que se hace, es esencial.

11- Las mejores arquitecturas, requisitos y diseños emergen de equipos que se auto organizan.

12- En intervalos regulares, el equipo reflexiona sobre la forma de ser más efectivo y ajusta su conducta en consecuencia. (scrummanager, 2014)

### **Cronograma**

<b>Actividad</b>	<b>Fecha</b>
<b>Iteración 1: Duración 2 semanas</b>	
1- Levantamiento de la lista de deseos con el Cliente.	05-08-2019 / 09-08-2019
2- Creación de las Historias de Usuario.	12-08-2019 / 16-08-2019
<b>Iteración 2: Duración 2 semanas</b>	
1. Desarrollo Formulario Login y Registro con Laravel	20-08-2019 / 23-08-2019
2. Desarrollo Formulario Categorías	26-08-2019 / 29-08-2019
3. Documentación Login, Registro y Categorías / Reunión con Cliente	29-08-2019 / 30-08-2019
<b>Iteración 3: Duración 2 semanas</b>	
1. Desarrollo formulario Catálogo de Productos	02-09-2019 / 11-09-2019
2. Documentación Catálogo de Productos / Reunión con Cliente	12-09-2019 / 13-09-2019
<b>Iteración 4: Duración 1 semana</b>	
1. Desarrollo Servicio API Rest con Laravel.	16-09-2019 / 20-09-2019

<b>Iteración 5: Duración 1 semana</b>	
1. Desarrollo Activity Perfil Profesional	23-09-2019 / 26-09-2019
2. Documentación Activity Perfil Profesional	26-09-2019 / 27-09-2019
3. Pruebas de diseño y usabilidad con usuarios finales / Reunión con Cliente	27-09-2019
<b>Iteración 6: Duración 1 semana</b>	
1. Desarrollo Activity Acerca de	30-09-2019 / 03-10-2019
2. Documentación Activity Acerca de	03-10-2019 / 04-10-2019
3. Pruebas de diseño y usabilidad con usuarios finales / Reunión con Cliente	04-10-2019
<b>Iteración 7: Duración 1 semana</b>	
1. Desarrollo prototipo Activity Catálogo de Productos	07-10-2019 / 11-10-2019
<b>Iteración 8: Duración 3 semanas</b>	
1. Desarrollo Constructor de CardView con kotlin (se usará para integración automática con APIRest del catálogo de productos)	15-10-2019 / 29-10-2019
2. Documentación Constructor / Reunión con Cliente	30-10-2019 / 01-11-2019
<b>Iteración 9: Duración 1 semana</b>	
1. Pruebas de Integración Constructor de CardView con Activity Catálogo de Productos	05-11-2019 / 07-11-2019
2. Documentar Activity Catálogo de Productos / Reunión con Cliente	08-11-2019
<b>Iteración 10: Duración 1 semana</b>	

1. Desarrollo formulario Contáctenos. (se integrará con los clientes de correo electrónico instalados en el dispositivo mediante Call to Action)	12-11-2019 / 14-11-2019
2. Documentar Formulario Contáctenos / Reunión con Cliente	15-11-2019
<b>Iteración 11: Duración 1 semana</b>	
1. Pruebas Funcionales y de integridad de datos con el Cliente.	18-11-2019 / 21-11-2019
2. Publicación Sitio Web y APP en los repositorios definidos	22-11-2019

*Tabla 1 Cronograma*

## Presupuesto

<b>Recurso Humano</b>				
Encargado	Rol	Valor hora	# Horas proyecto	Valor total
yo	Desarrollador	\$20.0000	128	2.560.000

<b>Recursos físicos</b>			
Tipo	# Meses	Valor mensual	Valor total
Internet	4	30.000	120.000
Servicios públicos	4	50.000	200.000
Transporte	4	20.000	80.000
<b>Total</b>		<b>400.000</b>	

<b>Recursos software y hardware</b>			
Tipo	Cantidad	Valor unitario	Valor total
Equipo de cómputo	1	1.500.000	1.500.000
host	1	246.600	246.600
Key playstore	1	86.000	86.000
Total		1.832.600	

*Tabla 2 Presupuesto*

Valor total del proyecto: \$ 4.393.000



## Capítulo 4. Resultados

Para llevar a cabalidad la puesta en marcha del proyecto TecnoWilly, fue necesario distribuir el desarrollo de la aplicación en 3 etapas de la siguiente manera:

- Sistema de información web para la creación de productos.
- Desarrollo de rutas API para la sincronización del catálogo con la APP.
- Desarrollo de APP móvil para la presentación de productos y servicios.

### Aplicación WEB

Para el desarrollo de la aplicación web fueron levantadas 4 historias de usuario con el cliente final las cuales se detallan a continuación.

HU_001	Como Administrador quiero poder autenticarme para poder administrar el sistema web	
<b>CRITERIOS DE ACEPTACIÓN</b>		
1	<b>Ingreso exitoso</b>	Al ingresar el usuario y contraseña, el sistema validara con la base de datos de usuarios activos y permitirá el login exitoso a la aplicación
2	<b>Ingreso exitoso</b>	si se confirma el acceso correcto al sistema, se mostrara el menú de la aplicación con permisos para las funcionalidades "Registro de Usuario, Maestro de Categorías y Maestro de Productos"
3	<b>Ingreso fallido</b>	En caso que el usuario y contraseña ingresado sean incorrectos, se notificara sobre el intento de login fallido y no se permitirá el acceso a ninguna de las funcionalidades del sistema
4	<b>Ingreso fallido</b>	Si el ingreso es fallido, el administrador del sistema debe generar usuario y contraseña por el modulo de Gestión de Usuario de la HU_002
5	<b>Campos Obligatorios</b>	Al iniciar sesión se debe validar que el correo y contraseña sean obligatorios, de no ingresar alguno de los dos registros el sistema notificara "Usuario y/o contraseña incorrectos"
6	<b>Tiempo de Sesión</b>	la sesión del sistema debe terminarse cada 30 minutos de no identificar interacción por parte del usuario
7	<b>Cerrar Sesión</b>	en el menú del sistema debe existir un botón para terminar la sesión correctamente por parte del usuario

Tabla 3 Historia de Usuario Administración Web

<b>HU_002</b>	<b>Como Administrador quiero acceder al modulo gestión de usuarios para poder administrar el acceso al sistema</b>	
<b>CRITERIOS DE ACEPTACIÓN</b>		
<b>1</b>	<b>Al dar clic en el menú "Registrar Usuario"</b>	el sistema cargara la pantalla para la administración de los usuario de acceso al sistema. En esta pantalla se permitirá crear, actualizar y eliminar usuario, adicionalmente debe constar de un filtro de búsqueda general y una tabla en la cual visualizar todos los usuarios del sistema
<b>2</b>	<b>Al dar Clic en el Botón "Crear Usuario"</b>	EL sistema debe permitir crear un nuevo usuario con las siguientes condiciones: 1- Nombre: Nombre del usuario 2- Correo: Correo electrónico que a su vez será utilizado como el usuario de acceso al sistema, el correo debe ser único y esto debe validarse en pantalla 3-Contraseña: se debe crear una contraseña de mínimo 8 caracteres y se debe confirmar para asegurar que el usuario esta agregando los datos correctamente
<b>3</b>	<b>Modal Crear Usuario - Clic en el botón "Registrar"</b>	al dar clic en el botón registrar se debe validar que todos los datos sean obligatorios y que cumpla con las contusiones descritas para cada campo en el criterio 2. de no cumplirse cualquiera de ellos se deberán generar las siguientes validaciones: 1- Correo: el correo debe ser un correo valido con @, debe ser único en el sistema, si el correo ya existe se informara en pantalla que dicho usuario ya esta creado. 2- Confirmar Contraseña: se validara que los dos campos sean iguales, si alguno de los dos campos no coincide se informara en pantalla "contraseñas ingresadas no coinciden" 3- si alguno de los campos obligatorios no fue completado, se notificara al Admón. en pantalla el campo que debe completar
<b>4</b>	<b>Modal Crear Usuario - Clic en el botón "Registrar"</b>	Si las validaciones para la creación de usuario se cumplieron satisfactoriamente, se procede con la creación del usuario para acceso al sistema
<b>5</b>	<b>Modal Crear Usuario - Clic en el botón "Cerrar"</b>	Al presionar el botón "cerrar" se saldrá del modal de registro y regresara a la pantalla principal de registro de Usuarios
<b>6</b>	<b>Filtro de Búsqueda</b>	la pantalla de registro de usuarios debe permitir buscar un usuario por nombre o correo electrónico
<b>7</b>	<b>Filtro de Búsqueda</b>	La tabla de resultados debe permitir cambiar la cantidad de registros mostrados en pantalla por medio de un filtro. Por defecto la tabla debe pintar información de 5 registrar y generar paginación de 5 en 5
<b>8</b>	<b>Editar Usuario</b>	El sistema permitirá actualizar un usuario registrado en la base de datos, al momento de guardar los cambios realizados se deben generar las mismas validaciones descritas en el criterio 4
<b>9</b>	<b>Eliminar Usuario</b>	El sistema debe permitir eliminar usuario existentes en el sistema, se realizara mediante un botón y se confirmara sobre la eliminación exitosa del registro

*Tabla 4 Historia de Usuario Gestión de Usuarios*

<b>HU_003</b>	<b>Como administrador quiero crear categorías para poder utilizarlas en el catalogo de productos</b>	
<b>CRITERIOS DE ACEPTACIÓN</b>		
<b>1</b>	Al dar clic en el menú "categorías"	el sistema carga la pagina principal de categorías, en la cual se podrá Crear, Actualizar y Eliminar, debe existir una tabla con la información de todas las categorías ingresadas al sistema y un filtro de búsqueda para revisar la información
<b>2</b>	En el menú categorías, al presionar el botón "Añadir Categorías"	el sistema cargara el formulario para el registro de una nueva categorías en el sistema. En esta pantalla se ingresara el nombre de la categoría y una descripción
<b>3</b>	en el menú Añadir Categorías, clic en Registrar	el sistema validara los campos de la siguiente manera: 1- Nombre: campo texto, obligatorio y único. Si se intenta registrar una categoría con el mismo nombre, el sistema debe generar alerta y no continuar le proceso. 2- Campo Descripción: no obligatorio.
<b>4</b>	en el menú Añadir Categorías, clic en Cerrar	El sistema cancelara la creación de categoría y re direccionara automáticamente a la pagina principal
<b>5</b>	al presionar el botón Actualizar	si selecciono una categoría existente, el sistema debe permitir editar la información teniendo en cuenta las validaciones del criterio 3
<b>6</b>	al presionar el botón Eliminar	El sistema eliminara la categoría y notificara al usuario sobre el proceso exitoso
<b>7</b>	Al realizar filtros de búsqueda	el sistema debe permitir filtrar por los campos de Nombre y Descripción, debe paginar de 5 en 5 y permitir cambiar los filtros de búsqueda para el paginado

*Tabla 5 Historia de Usuario Maestro Categorías*

HU_004	Como administrador quiero crear productos para poder ofertarlos en la aplicación WEB	
CRITERIOS DE ACEPTACIÓN		
1	Al dar clic en el menú "Catalogo de Productos"	el sistema carga la pagina principal de productos en la cual se podrá Crear, Actualizar y desactivar debe existir una tabla con la información de todas los productos ingresados al sistema y un filtro de búsqueda para revisar la información
2	En el menú Productos al presionar el botón "Añadir Producto"	el sistema cargara el formulario para el registro de una nuevo Producto en el sistema. En esta pantalla se ingresara la Categoría, un Nombre para el producto y la descripción del mismo". Adicionalmente, se podrán cargar entre 1 y 5 imágenes
3	en el menú Productos- clic en Registrar	el sistema validara los campos de la siguiente manera: 1- Categoría: campo tipo select de la tabla de categorías y es un campo obligatorio. 2- Campo Nombre: será el campo con el cual se mostrar el producto en la aplicación móvil. Es obligatorio 3-Campo Descripción, se realiza una breve descripción del producto. 4- Imagen: debe existir minino 1 imágenes y máximo 5
4	en el menú Añadir Producto clic en Cerrar	El sistema cancelara la creación del producto y re direccionara automáticamente a la pagina principal
5	al presionar el botón Actualizar	si selecciono un producto existente, el sistema debe permitir editar la información teniendo en cuenta las validaciones del criterio 3
6	al presionar el botón Desactivar	El sistema desactivara el producto seleccionado y notificara al usuario sobre el proceso exitoso
7	Al realizar filtros de búsqueda	el sistema debe permitir filtrar por los campos de Nombre, Descripción y Categoría, debe paginar de 5 en 5 y permitir cambiar los filtros de búsqueda para el paginado
8	Sincronización APP Móvil	para la sincronización de los productos con la aplicación móvil se debe construir una APIRest el cual realizara solo peticiones de consulta a la base de datos sobre los productos activos y estos serán los que se muestran al usuario final

Tabla 6 historia de usuario Catálogo de Productos

El desarrollo de la aplicación está basado MVC distribuido de la siguiente manera:

Controladores: el proyecto está compuesto por 4 controladores principales encargados de la autenticación, seguridad, categorías, productos y usuarios. Los controladores se encuentran relacionados con la base de datos por medio de los modelos evitando de esta manera el uso redundante de sentencias sql en el código.

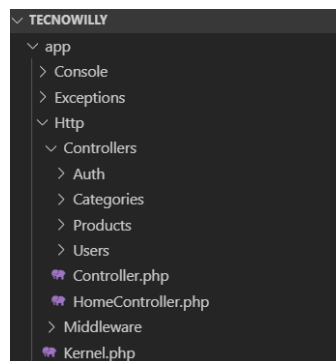


Ilustración 4 Ilustración Controllers

Modelos: el sistema cuenta principalmente con dos relaciones entre el modelo de productos y el modelo de categorías, pero adicionalmente, en el modelo de productos se encuentra una relación foreign key con el modelo de imágenes que es utilizado para almacenar las fotos asociadas a un producto.

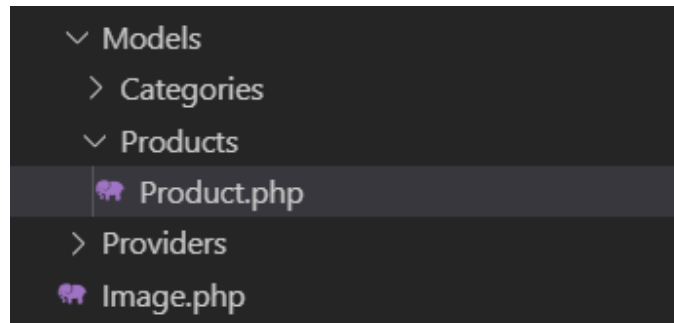


Ilustración 5 Models

```
class Product extends Model
{
    // se envian por medio de un array los datos que se quieren llenar
    protected $fillable=['category_id', 'description', 'price', 'user_id'];

    public function categoryname() // relacion 1 a 1 entre el modelo Product y el modelo Categories
    {
        // Categories es el modelo donde esta la relacion, 'id' es el campo en Categories y 'category_id'
        return $this->hasOne(Categories::class,'id','category_id');
    }

    public function image() // relacion 1 a 1 entre el modelo Product y el modelo Categories
    {
        // Categories es el modelo donde esta la relacion, 'id' es el campo en Categories y 'category_id'
        return $this->hasMany(Image::class,'product_id','id');
    }
}
```

Ilustración 6 Ejemplo relación modelos

Vistas: las vistas de la aplicación fueron diseñadas de forma dinámica independizando el menú principal, del menú de navegación y las vistas de cada uno de los módulos. En las vistas secundarias se llama una extensión y se agregan layouts.

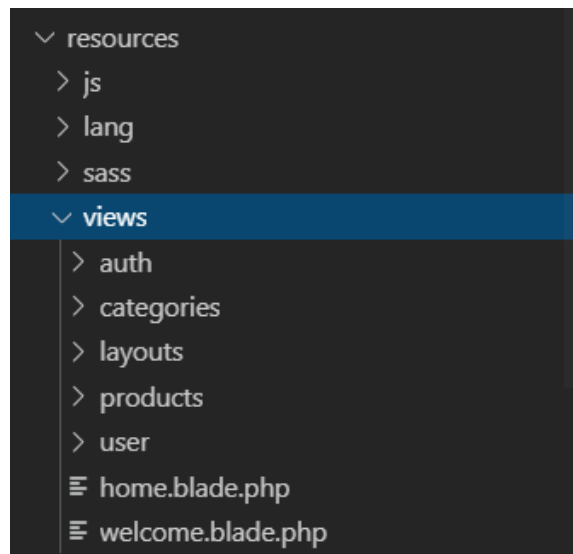


Ilustración 7 resources

El diseño web del sitio está orientado a ser responsive utilizando los atributos de filas y columnas del framework bootstrap 4. De igual forma, se utilizó una plantilla SBAdmin de bootstrap para los estilos principales de la aplicación.

Para la creación de tablas y relaciones se utilizó el recurso de migraciones que tiene incluido el framework Laravel, en el cual se crean las tablas directamente en código y para crearlas en la base de datos basta con ejecutar **php artisan migrate** en la consola de comandos y automáticamente se crearan las tablas y las relaciones en la base de datos.

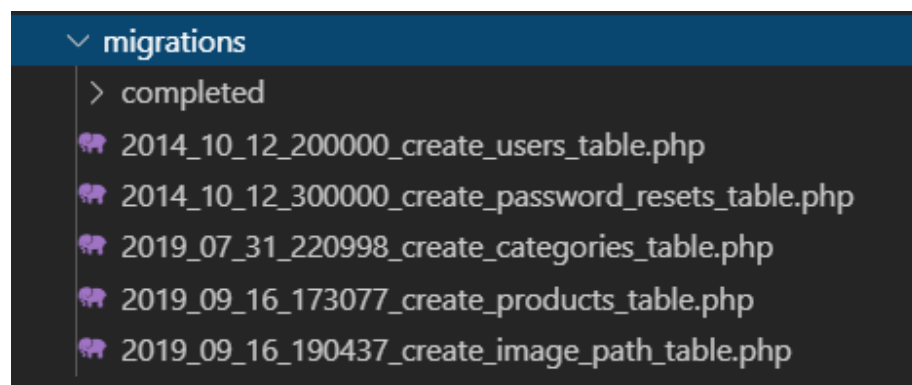


Ilustración 8 migrations

```
public function up()
{
    Schema::create('image_path', function (Blueprint $table) {
        $table->bigIncrements('id');
        $table->bigInteger('product_id')->unsigned()->index();
        $table->string('path');
        $table->timestamps();
    });

    Schema::table('image_path', function($table){
        $table->foreign('product_id')->references('id')->on('products')->onDelete('cascade');
    });
}
```

*Ilustración 9 ejemplo tabla desde migrate*

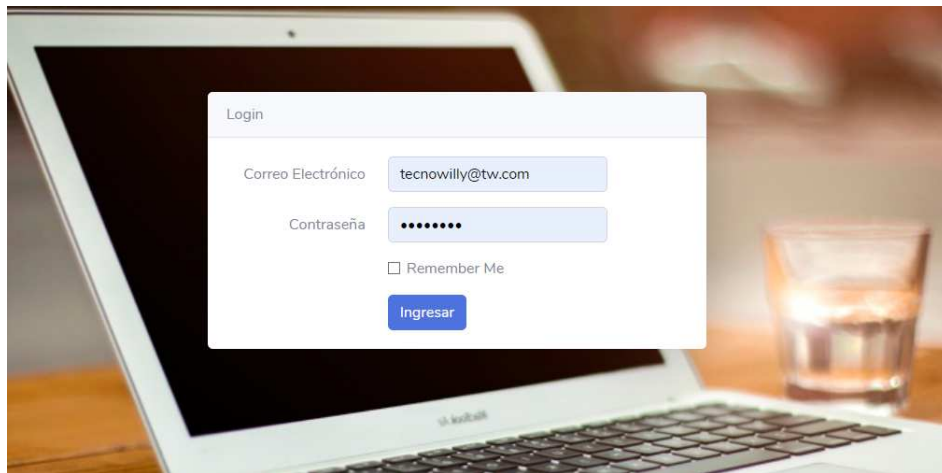
El sistema de configuración y conexión del proyecto se toma del archivo .env incluido en Laravel en el cual se establecen variables globales a utilizar en el sistema y se definen los datos de conexión a base de datos.

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_HOST=10.220.███
DB_PORT=3306
DB_DATABASE=db_tecnowilly
DB_USERNAME=c1s███
DB_PASSWORD=crr███
```

*Ilustración 10 archivo conf. Global*

## Resultado Final Aplicación WEB

En la siguiente imagen se puede observar el sistema de login de TecnoWilly. Los usuarios son únicos y las validaciones realizadas se utilizan por medio del framework de Laravel.



*Ilustración 11 - login*

En el menú de navegación el usuario encontrará los respectivos enlaces que redireccionarán al maestro de categorías, catálogo de productos y la gestión de usuarios del sistema.



*Ilustración 12 Menú de navegación*



Al dar clic en la pantalla de catálogo de productos, el usuario podrá revisar los elementos que ya han sido creados, así como agregar nuevos productos. Los productos se relacionan con una categoría del sistema los cuales son únicos y se crean por el maestro de categorías.

Lista de Productos

[+ Añadir Producto](#)

Mostrar:  registros

Buscar:

Categoría	Descripción	Precio	Ver	Editar	Eliminar
Portátil	Portátil Lenovo	\$1.250.000			

Mostrando registros del 1 al 1 de un total de 1 registros

[Anterior](#) [1](#) [Siguiente](#)

Ilustración 13 datatable catálogo de productos

Actualizar Producto

Categoría\*

Descripción\*

Precio\*

Imágenes\*  No se eli...ó archivo  
 No se eli...ó archivo  
 No se eli...ó archivo



Ilustración 14 Creación de Producto

Para las notificaciones presentadas al usuario se utilizó una librería llamada SweetAlert que generar alertas dinámicas al usuario final.



## Registro modificado con éxito

Success Message

*Ilustración 15 Notificaciones SweetAlert*

Si presiona en el menú de categorías, el usuario podrá observar los elementos creados o crear nuevas categorías. Los nombres de una categoría son únicos y están asociados directamente con el catálogo de productos.

Actualizar Categoría

Nombre\*

Descripcion\*

*Ilustración 16 Creación Categoría*

Si la persona presiona el menú de gestión de usuarios, podrá crear un nuevo administrador del sistema, este usuario es único y se realiza la validación por medio del correo electrónico.



Actualizar Usuario

Nombre

Correo Electrónico

Contraseña

Confirmar Contraseña

*Ilustración 17 creación usuario*

### **API método GET consulta de Productos.**

Después de creados los productos en el sistema TecnoWilly, se debe generar una sincronización en la aplicación móvil para tomar los cambios de los productos agregados o eliminados en el catálogo.

Debido a que el API es un sistema de solo consulta, no es necesario generar autenticación adicional para que la aplicación móvil se consuma los servicios y genere una respuesta.

En la construcción del API se utilizó una ruta adicional en el sistema web y se creó una función en el controlador de Productos “ProductController” encargada de enviar un arreglo con la información del producto, la categoría y las imágenes asociadas. Esta función la acción que realiza es consultar las relaciones en el modelo de productos y recorrer un foreach o un ciclo con cada uno de los productos encontrados y devolverlos al servicio GET de consulta.

```

public function indexApi(){
    $products=Product::orderBy('id', 'DESC')->get();
    // con este modelo estoy capturando los datos de las relaciones desde el modelo
    foreach ($products as $key => $value) {
        $value->categoryname;
        $value->image;
    }
    return $products;
}

```

*Ilustración 18 Funcion en ProductController – API*

```

class Product extends Model
{
    // se envian por medio de un array los datos que se quieren llenar
    protected $fillable=['category_id', 'description', 'price', 'user_id'];

    public function categoryname() // relacion 1 a 1 entre el modelo Product y el modelo Categories
    {
        // Categories es el modelo donde esta la relacion, 'id' es el campo en Categories y 'category_
        return $this->hasOne(Categories::class,'id','category_id');
    }

    public function image() // relacion 1 a 1 entre el modelo Product y el modelo Categories
    {
        // Categories es el modelo donde esta la relacion, 'id' es el campo en Categories y 'category_
        return $this->hasMany(Image::class,'product_id','id');
    }
}

```

*Ilustración 19 relaciones necesarias modelo de productos*

Para poder presentar la información de las relaciones, en Laravel se debe describir si la ruta se presenta para la web o para un API. En este caso se debe crear una ruta de respuesta en el archivo api.php y como no se requiere autenticación se excluye del middleware de autenticación.

```

Route::middleware('auth:api')->get('/user', function (Request $request) {
    return $request->user();
});

Route::get('indexApi', 'Products\ProductController@indexApi');

```

*Ilustración 20 ruta API.*

Para probar el API se puede utilizar la aplicación POSTMAN que está dedicada a la solicitud y respuesta de peticiones.

En la siguiente grafica se muestra la forma adecuada para consumir el servicio, la ruta y el arreglo que retorna con la información que será utilizada en el catálogo de productos.

Ruta:

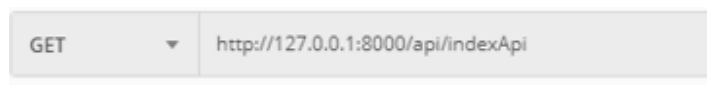


Ilustración 21 ruta API postman

Respuesta:

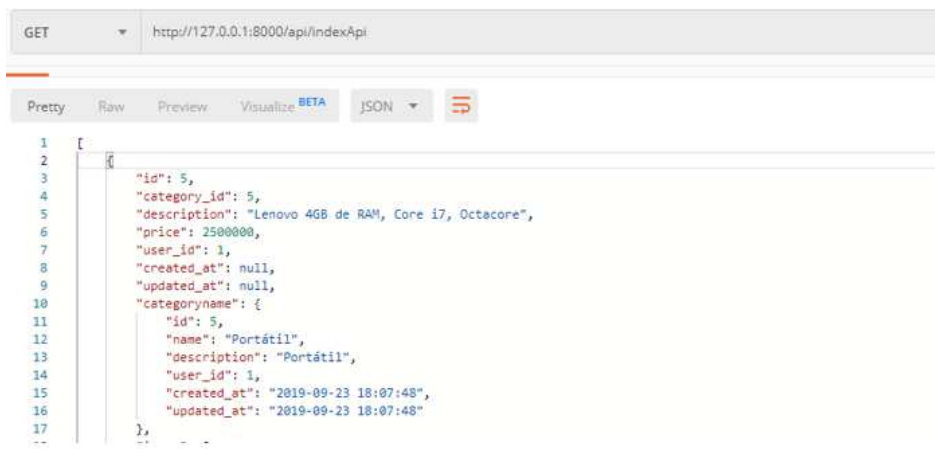


Ilustración 22 array información producto



Ilustración 23 array imagen de producto

## Aplicación Móvil

Para el desarrollo de la aplicación móvil se tuvieron en cuenta las solicitudes del cliente de contar con una vista acerca de, otra vista para el catálogo de productos dinámica de acuerdo con la información de productos activa en el sitio web y una vista para una descripción del perfil profesional.

El desarrollo de la aplicación está realizado en Android studio y como lenguaje de programación para las clases y métodos generales se utilizó Kotlin. La aplicación móvil se alimenta por medio de un servicio API Rest de Laravel que a su vez devuelve la información final de los productos en un JSON que es utilizado con la librería Retrofit en Android para consumir dicho servicio y convertir los datos en una lista válida para la aplicación móvil.

Vistas:

En el desarrollo de las vistas fue necesario utilizar activity's, fragments y layouts resource files para implementar la presentación de las diferentes vistas al usuario. La aplicación está basada en la utilización de tabs de navegación en la cual el usuario se puede desplazar de forma lateral por medio de la pantalla o presionando en cada uno de los títulos.

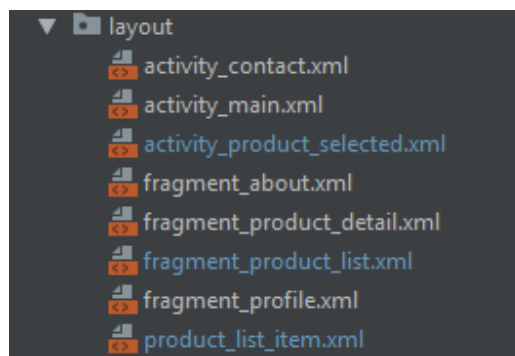


Ilustración 24 Layout aplicación movil

La aplicación móvil cuenta con una clase principal encargada de ejecutar los intentos de navegación y peticiones del usuario denominadas intent. Para navegar por los diferentes tabs es importante definir al sistema que fragmento o activity se debe mostrar al momento de cambiar entre tabs, para esto se creó una clase llamada **FragmentAdapter** en la cual se define los intentos de navegación y la acción a ejecutar sobre cada una de ellas, adicionalmente, cada posición del tab cuenta con una clase en la cual se definen las acciones por cada uno de ellos. (Activities/ “FragmentProduct, FragmentProfile, FragmentAbout”)

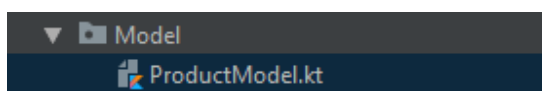
```
class FragmentAdapter (fragmentManager: FragmentManager) : FragmentPagerAdapter(fragmentManager) {  
  
    override fun getItem(position: Int): Fragment {  
        return when (position) {  
            0 -> {  
                FragmentAbout()  
            }  
            1 -> FragmentProduct()  
            else -> {  
                return FragmentProfile()  
            }  
        }  
    }  
  
    override fun getCount(): Int {  
        return 3  
    }  
  
    override fun getPageTitle(position: Int): CharSequence {  
        return when (position) {  
            0 -> "HOME"  
            1 -> "CATALOGO"  
            else -> {  
                return "PERFIL"  
            }  
        }  
    }  
}
```

*Ilustración 25 Clase PlaceholderFragment*

En la ilustración 25 se puede observar claramente que los intentos de navegación se definen por medio de un condicional when que retorna por cada tab que fragmento se debe ubicar.

## Modelos:

Para consumir y convertir los datos devueltos por el JSON es necesario crear clases para recibir cada uno de los parámetros del array. Para esto se creó una carpeta Model con las clases necesarios para la obtención de los datos.



*Ilustración 26 Carpeta Model*

```
data class Product(  
    val category_id: Int,  
    val categoryname: Categoryname,  
    val characteristic: String,  
    val description: String,  
    val host: String,  
    val id: Int,  
    val image: List<Image>,  
    val num: String,  
    val price: String  
) : Serializable
```

*Ilustración 27 Modelo Principal Productos*

```
data class Image(  
    val id: Int,  
    val path: String,  
    val product_id: Int  
) : Serializable
```

*Ilustración 28 Modelo para las Imágenes*

```
data class Categoryname(  
    val created_at: String,  
    val description: String,  
    val id: Int,  
    val name: String,  
    val updated_at: String,  
    val user_id: Int  
) : Serializable
```

*Ilustración 29 Modelo Categorías*



## Retrofit – consumo de API:

Para el consumo del API se implementó la librería retrofit, en la cual se crea un archivo tipo interface para listar todo el array devuelto por el JSON por el método GET y otro archivo para recibir los objetos del cliente, en este caso la URL de acceso y la conversión de datos.

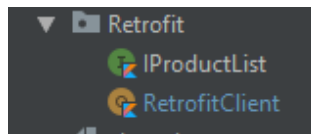


Ilustración 30 Retrofit

```
interface IProductList {  
    @GET("indexApi")  
    fun getProduct(): Observable<ProductModel>  
}
```

Ilustración 31 interface IProductList

```
Object RetrofitClient {  
    private var okHttpClient: OkHttpClient = OkHttpClient.Builder()  
        .connectTimeout( timeout: 30, TimeUnit.SECONDS)  
        .readTimeout( timeout: 30, TimeUnit.SECONDS)  
        .writeTimeout( timeout: 30, TimeUnit.SECONDS)  
        .build()  
  
    fun getRetrofitInstance(): Retrofit {  
        return Retrofit.Builder()  
            .addCallAdapterFactory(RxJava2CallAdapterFactory.create())  
            .addConverterFactory(GsonConverterFactory.create())  
            .client(okHttpClient)  
            .baseUrl( baseUrl: "http://192.168.1.2:81/tecnowilly/public/api/")  
            .build()  
    }  
}
```

Ilustración 32 Retrofit Cliente

## Adaptador Lista de Productos:

Para construir e implementar los datos en la aplicación móvil, se creó un adaptador que recibe una instancia de la lista de productos y se encarga de pintar la información dinámicamente en cada uno de los fragmentos por medio de los id asignados a los TextView, CardView, ImageView, entre otros.

```
@SuppressWarnings("SetTextI18n")
override fun onBindViewHolder(holder: ViewHolder, position: Int) {
    product = productList[position]
    Glide.with(context).load("http://"+productList[position].host + productList[position].image[0].path).into(holder.imageView)
    holder.txt_description.text=productList[position].description
    holder.txt_price.text= "$"+productList[position].price
    holder.txt_product.text = productList[position].categoryname.name
    holder.bindProduct(product)
}

inner class ViewHolder(itemView: View):RecyclerView.ViewHolder(itemView),View.OnClickListener{

    private var product: Product? = null

    internal var img_product: ImageView = itemView.findViewById(R.id.product_image) as ImageView
    internal var txt_description: TextView = itemView.findViewById(R.id.product_description) as TextView
    internal var txt_price : TextView = itemView.findViewById(R.id.product_price) as TextView
    internal var txt_product: TextView = itemView.findViewById(R.id.product_name) as TextView

    fun bindProduct(product: Product) {
        this.product = product
    }

    init {
        itemView.setOnClickListener(this)
    }
}
```

Ilustración 33 Adaptador Lista de Productos

## Constructor Vista dinámica catálogo de Productos:

Para recibir la información transformada y compilada por retrofit y la instancia de asignación por id, es necesario utilizar la librería RecyclerView, encargado de mostrar los datos de acuerdo a los parámetros enviados por el adaptador. A su vez el recyclerview envía los fragmentos de datos a un layout y se pinta de forma dinámica en las CardView definidas para este proceso.

```

<LinearLayout
    android:layout_gravity="center_horizontal"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <com.mancj.materialsearchbar.MaterialSearchBar
        style="@style/MaterialSearchBarLight"
        app:mt_speechMode="true"
        app:mt_hint="Buscar Productos"
        app:mt_maxSuggestionsCount="10"
        android:layout_marginTop="65dp"
        android:layout_marginRight="5dp"
        android:layout_marginLeft="5dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/searchBar" />

    <android.support.v7.widget.RecyclerView
        android:layout_marginTop="10dp"
        android:id="@+id/product_recycler_view"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_gravity="center_horizontal"/>

</LinearLayout>

```

*Ilustración 34 RecyclerView*

```

class FragmentProduct : Fragment(), ItemClickListener {

    private var listener: OnFragmentInteractionListener? = null
    private var productAdapter: ProductAdapter? = null
    private lateinit var recyclerViewProduct: RecyclerView

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        val viewInflater = inflater.inflate(R.layout.fragment_product_list, container, attachToRoot: false)
        recyclerViewProduct = viewInflater.findViewById(R.id.product_recycler_view)
        callServiceProduct()

        return viewInflater
    }

    override fun onAttach(context: Context) {
        super.onAttach(context)
        if (context is OnFragmentInteractionListener) {
            listener = context
        } else {
            println("Error")
        }
    }

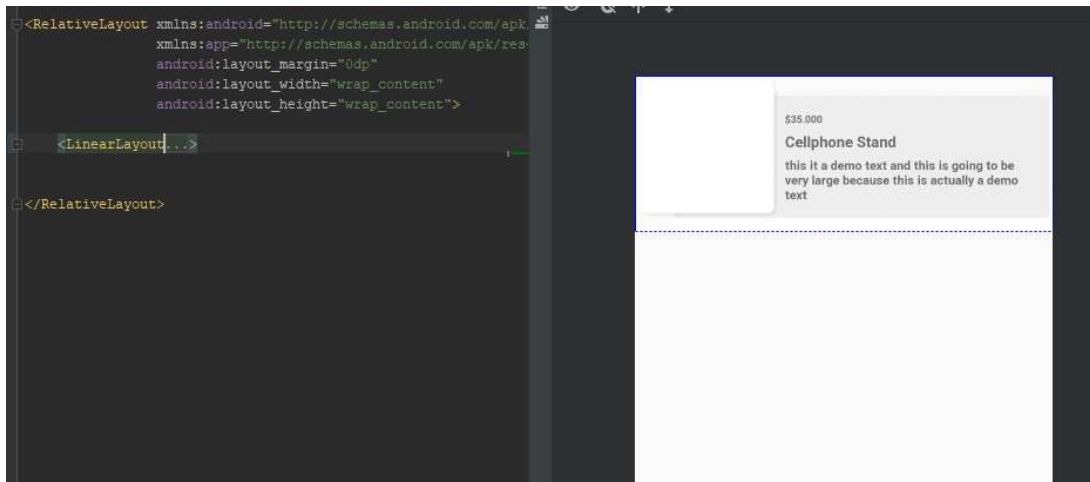
    override fun onDetach() {

```

*Ilustración 35 Carga de Información RecyclerView*

Las tarjetas o CardView que reciben la información organizada por el recyclerview constan de un Relative Layout y los respectivos linear layout para organizar la información en forma vertical. Para mostrar la información se utilizaron CardView que a su vez contienen etiquetas de

imagen y texto. Por defecto se crean las tarjetas vacías y a medida que el recyclerView cuenta y ordena datos se va reemplazando de forma dinámica para el usuario.



*Ilustración 36 Default Lista de Productos – Catalogo*

### **Ver detalle del producto:**

Para ver el detalle de un producto al dar clic sobre la tarjeta del mismo, se crearon las clases necesarias que realizan los intent sobre el producto seleccionado, para esto se utilizó una clase llamada **ProductSelectedActivity** en **ProductSelectedActivity** la cual recibe una llave enviada desde la función `itemSelected` en `FragmentProduct`, al presionar el producto se abre un activity para mostrar el resto de la información almacenada para el mismo.

```
override fun itemSelected(product: Product) {  
    activity?.let { it: FragmentActivity  
        val intent = Intent(it, ProductSelectedActivity::class.java)  
        intent.putExtra( name: "key_product", product)  
        it.startActivity(intent)  
    }  
}
```

*Ilustración 37 key\_product fun itemSelected*

```

//Inicializacion de las variables para el adaptador de ProductDetail
private lateinit var product: Product
private lateinit var product_image_all: ImageView
private lateinit var product_name: TextView
private lateinit var product_description: TextView
private lateinit var product_description_long: TextView
private lateinit var product_price: TextView

override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_product_selected)

    if (intent.hasExtra( name: "key_product")) {
        product = intent.getSerializableExtra( name: "key_product") as Product
    }

    product_image_all = findViewById(R.id.product_image_all)
    product_name = findViewById(R.id.product_name)
    product_description = findViewById(R.id.product_description)
    product_description_long = findViewById(R.id.product_description_long)
    product_price = findViewById(R.id.product_price2)

    //se llenan los datos a cada uno de los campos
    Glide.with( activity: this).load( string: product.host + product.image[0].path).into(this.product_image_all)
    product_price.text = "$" + product.price
    product_name.text = product.categoryname.name
    product_description.text = product.description
    product_description_long.text = product.characteristic
}
}

```

*Ilustración 38 ProductSelectedActivity - Adaptador Detalle del Producto*

## Resultado Final Aplicación Móvil:

Al ingresar a la aplicación lo primero que se podrá observar es el HOME el cual cuenta con un apartado de QUIENES SOMOS, otro para los SERVICIOS y por último el número de contacto del TecnoWilly.



En el segundo tab se podrá encontrar el catálogo de productos el cual se llena de forma automática por medio del servicio construido en Laravel que recibe la información de la base de datos de los productos.



En el último tab se encontrará la vista de perfil profesional con la información detallada de la hoja de vida para ser presentadas a los posibles clientes interesados en adquirir algún servicio.



Por ultimo en todos los tab se cuenta con un botón flotante de contacto, en el cual un cliente puede ingresar a redactar una solicitud y enviarla a TecnoWilly.

A screenshot of a contact form within the TecnoWilly application. The form is set against a dark blue background. At the top, the 'TecnoWilly' logo is displayed in white. Below the logo are four input fields: 'Nombre Completo:', 'Teléfono:', 'E-mail:', and 'Comentario:'. Each of the first three fields is a white rounded rectangle. The 'Comentario:' field is a larger white rounded rectangle. At the bottom of the form is a red rounded button with the text 'Enviar' in white.



## **Capítulo 5. Conclusiones y Recomendaciones**

En el desarrollo de la aplicación web y móvil se logró llevar a cabalidad los objetivos trazados para la presentación de un catálogo de productos y servicios a los clientes de la empresa TecnoWilly logrando así generar una aplicación para los dispositivos móviles que acercará a los consumidores con la empresa.

En esta etapa se lograron aplicar conceptos de la carrera que actualmente estamos cursando y es gratificante poder llevar a cabalidad un proyecto del cual se puede decir que brindará una perspectiva totalmente diferente a la ofrecida actualmente por la empresa.

Las interfaces de la aplicación fueron diseñadas de forma tal que sean de fácil uso para el cliente, brindando una interacción sencilla y agradable a la vista. El API Rest para la sincronización de los productos fue construido con el framework Laravel el cual realiza una actualización de los productos cada vez que el usuario abre la aplicación, siempre y cuando cuente con acceso a internet.

Cada una de las fases de desarrollo brindó conocimiento y aprendizaje continuo que aporta a la formación profesional que como estudiantes queremos lograr, es satisfactorio poder aprender y reforzar dichos conocimientos con el paso del tiempo y con cada uno de los obstáculos que se presentaron.

### **Recomendaciones**

- Realizar implementación a futuro con medios de pago para fidelizar la comercialización de productos tecnológicos por medio de la aplicación de forma aún más efectiva.

- Generar estrategias de fidelización a través de redes sociales y otros medios digitales para divulgar el uso de la aplicación.

- Integrar la aplicación móvil con FireBase para la generación de estadísticas de uso y consumo por medio del servicio móvil.

- Integrar un cliente de correo automático para la vista de contáctenos, para que el cliente no tenga la necesidad de usar su aplicación de correo instalada.

## Lista de referencias

- Alvarez, M. A. (02 de 01 de 2014). *DesarrolloWeb.com*. Obtenido de <https://desarrolloweb.com/articulos/que-es-mvc.html>
- Arana, R. (s.f.). *Qué es la transformación digital y por qué es necesaria para cualquier negocio*. Obtenido de <https://www.ttandem.com/blog/que-es-la-transformacion-digital-y-por-que-es-necesaria-para-cualquier-negocio/>
- Bahit, E. (n.d.). El paradigma de la Programación Orientada a Objetos en PHP y el patrón de arquitectura de Software MVC. Retrieved May 20, 2018, from. Obtenido de <http://www.monografias.com/trabajos89/poo-y-mvc-php/poo-y-mvc-php2.shtml#introduc>
- Campos, C. (2 de 06 de 2016). *4 Razones por las que debes crear una App para tu negocio*. Obtenido de <https://gec.mx/4-razones-por-las-que-debes-crear-una-app-para-tu-negocio/>
- coresystems. (21 de 05 de 2018). *Apps Móviles y la Transformación Digital*. Obtenido de <https://coresystems.io/es/apps-moviles-la-transformacion-digital/>
- Cortés, M. F. (s.f.). *CTA o Call To Action: qué son, ejemplos y cómo usarlos*. Obtenido de <https://neoattack.com/cta-call-to-action/>
- dataiFX. (03 de 06 de 2019). *En 2018, ingresos de Mercado Libre en Colombia aumentaron 58%*. Obtenido de [dataifx.com/noticias/en-2018%20ingresos-de-mercado-libre-en-colombia-aumentaron-58](https://dataifx.com/noticias/en-2018%20ingresos-de-mercado-libre-en-colombia-aumentaron-58)
- Developers. (s.f.). *Introducción a Android Studio*. Obtenido de <https://developer.android.com/studio/intro?hl=es-419>
- Díaz, N. M. (11 de 2015). *Avance de las Aplicaciones móviles*. Obtenido de <http://jeuazarru.com/wp-content/uploads/2015/11/Apps.pdf>
- Gómez, J. C. (s.f.). *Caso Alibaba marcando las tendencias del networking y el comercio electrónico*. Obtenido de <https://www.marketinginteli.com/desarrollo-e-commerce/caso-alibaba/>
- Gretter, G. (s.f.). *¿Por qué mi empresa necesita una Aplicación Mobile, App?* Obtenido de <https://www.innovaportal.com/innovaportal/v/657/1/innova.front/por-que-mi-empresa-necesita-una-aplicacion-mobile-app>.
- hoy, E. (11 de 05 de 2017). *El uso de apps para ofrecer servicios: ¿Por qué toda empresa debería tener una?* Obtenido de <https://rpp.pe/campanas/contenido-patrocinado/el-uso-de-apps-para-ofrecer-servicios-por-que-toda-empresa-deberia-tener-una-noticia-1050014>

- Iqbal, M. (8 de 7 de 2019). *App Download and Usage Statistics (2019)*. Obtenido de <https://www.businessofapps.com/data/app-statistics/>
- Jurado, F. L. (01 de 06 de 2018). *Qué es Kotlin y sus características*. Obtenido de <https://openwebinars.net/blog/que-es-kotlin/>
- Monte, J. (2016). *Implantar scrum con éxito*. Editorial UOC. Obtenido de ProQuest Ebook Central, <http://ebookcentral.proquest.com/lib/bibliouniminutosp/detail.action?docID=4795177>.
- pulsosocial. (17 de 04 de 2018). *pulsosocial*. Obtenido de <https://pulsosocial.com/2018/04/17/empresas-mexicanas-buscan-cerrar-la-brecha-de-talento-tecnologico/>
- pymex.com. (s.f.). *¿Qué es AliExpress y cómo funciona?* Obtenido de <https://pymex.com/emprendedores/comercio-electronico/que-es-aliexpress-y-como-funciona/>
- Ramirez, I. (20 de 02 de 2018). *¿En qué se diferencian Alibaba y AliExpress?* Obtenido de <https://www.xataka.com/basics/en-que-se-diferencian-alibaba-y-aliexpress>
- Rouse, M. (12 de 2015). *Transformación digital*. Obtenido de <https://searchdatacenter.techtarget.com/es/definicion/Transformacion-digital>
- scrummanager. (27 de 04 de 2014). *El manifiesto ágil*. Obtenido de [https://www.scrummanager.net/bok/index.php?title=El\\_manifiesto\\_%C3%A1gil](https://www.scrummanager.net/bok/index.php?title=El_manifiesto_%C3%A1gil)
- Sinnaps. (s.f.). *METODOLOGÍA SCRUMBAN*. Obtenido de <https://www.sinnaps.com/blog-gestion-proyectos/metodologia-scrumban>
- TECNÓSFERA. (08 de 11 de 2017). *Vida social, en lo que más usan los colombianos el celular*. Obtenido de <https://www.eltiempo.com/tecnosfera/novedades-tecnologia/uso-del-celular-y-el-internet-en-colombia-149384>
- Trigas Gallego, M. (s.f.). *Metodología Scrum*. Obtenido de <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/17885/1/mtrigasTFC0612memoria.pdf>
- Viñaras, E. (s.f.). *Cyberclíc*. Obtenido de <https://www.cyberclick.es/numerical-blog/que-es-un-call-to-action>