



Universidad Nacional Autónoma de México

Facultad de Ingeniería

Cómputo móvil



Detalle técnico de desarrollo de la App

itmVendors®

Alumnos	
Anaya Ruíz Yair Alejandro	
Jimenez Ruiz Gustavo Alfredo	
Santander Martinez Ángel Antonio	
Profesor	
Ing. Marduk Pérez de Lara Dominguez	
Fecha de entrega: 18 de diciembre del 2021	Grupo: 02

Índice

Introducción	3
Desarrollo	4
Reglas de negocio y detalles	4
Requerimientos	5
Funcionales:	5
No funcionales:	6
Viabilidad	6
Alcance y producto mínimo viable de la aplicación	7
Funcionalidades y frameworks	7
Realidad aumentada	7
Android	7
iOS	8
Modelo Vista Controlador	9
Flujo de datos (API, Endpoints, autenticación, seguridad, etc)	12
Servicios	14
Comunicación de datos	15
Persistencia de los datos	15
Seguridad de los datos	15
Permisos y requisitos para publicación	15
Lenguajes de programación y servicios	16
Conclusiones	16
Referencias	17

Introducción

A lo largo del proceso de diseño de nuestra aplicación, hemos analizado a detalle las distintas fases que conlleva este proceso. Como primer trabajo entregado, se revisó la idea principal y las funcionalidades de la aplicación, así como sus características; como segundo análisis se realizó el diseño de la interfaz de la misma, verificando que su estructura fuera óptima para la navegación, funcional y amigable para los usuarios. Como último paso, nos enfocamos en la viabilidad de la implementación de nuestra idea principal al corroborar que es posible implementar las funciones y características definidas anteriormente.

El presente trabajo evalúa cada una de las funciones prometidas, analizando cómo se logra el objetivo de la aplicación con diferentes herramientas y lenguajes de programación a los que se tiene alcance. En nuestro caso particular, enfocarnos en una aplicación dedicada al comercio electrónico, nos brinda seguridad para poder llevar a cabo el desarrollo de la aplicación cumpliendo con lo establecido en un inicio, ya que hay diversas aplicaciones que están enfocadas en ecommerce y esto confirma que existen herramientas suficientes para desarrollar aplicaciones que se basen en la compra y venta de artículos en línea. El toque único será que los productos registrados en nuestra aplicación contarán con la posibilidad de ser visualizados en realidad aumentada. Nuestra visión es que la mayoría de los productos en la aplicación puedan contar con esta característica, habiendo excepciones en artículos donde el vendedor no proporcione el artículo para generar el modelo 3D.

A continuación se presentará el análisis e investigación de las distintas herramientas y lenguajes de programación que son factibles para el desarrollo técnico del producto mínimo viable, así como las especificaciones puntuales para el correcto funcionamiento de nuestra aplicación enfocada al comercio electrónico.

Desarrollo

Reglas de negocio y detalles

1. El usuario debe crear una cuenta en la aplicación: Es necesario que todo usuario que quiera hacer uso de ITM VENDORS, ingrese un correo electrónico válido, así como nombre completo y empresa a la que pertenece (en caso de aplicar), además de llenar campos básico como contraseña, nombre de usuario y teléfono, en el registro deberá proporcionar su INE para generar una cuenta única.
2. Es necesario agregar una dirección de hogar/oficina para el usuario que se da de alta como cliente, esto para que en caso de hacerse una compra, los artículos se envíen a dicho sitio.
3. Si el usuario planea hacer compras con tarjeta de crédito/débito, esta información se puede llenar al inicio de su registro en la aplicación, con el fin de no perder tiempo en el proceso de compra, claro está que los datos de dicha tarjeta deberán ser validados y aprobados por el mismo sistema al momento de su llenado.
4. Todo artículo que se registre en la aplicación para venta, deberá contener al menos 2 fotografías del mismo para que los clientes las vean, además de que si se desea mostrar en realidad aumentada, se debe enviar un artículo de prueba a la empresa para poder generar su respectivo modelo 3D, o bien enviar el modelo del artículo si es que el vendedor ya cuenta con él, esto con el fin de que se pueda consumir a través de una API de almacenamiento en la nube proporcionado por un tercero como puede ser AWS o Azure. Todo artículo que se registre con esta característica tendrá un costo extra en la renta de almacenamiento de la publicación.
5. Si el método de pago elegido por el cliente es en efectivo, este se deberá hacerse en las próximas 48 hrs. después del pedido, de no hacerlo, éste quedará con estatus de cancelado y los artículos que hayan estado en la compra actualizarán su número de stock disponible, ya que no quedan apartados.
6. Los usuarios registrados como vendedores, deberán informar acerca de las características de los productos y servicios, tales como las dimensiones, funcionalidades, color, material de fabricación, indicando de igual manera si se trata de un bien o producto nuevo, reparado, reacondicionado o usado.
7. Cada artículo que se desee vender, se mandará a revisión por moderadores, con el fin de verificar la información del producto y las promesas publicitarias de los mismos para evitar que afecte a poblaciones vulnerables o puedan ser perjudiciales para el usuario o medio ambiente.
8. Los usuarios como vendedores o comercios deben informar acerca de los medios para hacer válidas las garantías y realizar devoluciones, en el caso en el que un usuario desee realizar alguna de estas acciones.
9. Los vendedores o comercios deben comprometerse a publicar anuncios en sus publicaciones en caso de algún cambio en la mercancía que vendan, tales como agotamiento de existencias del producto o de una variante del mismo.

10. Los clientes tendrán un máximo de quince días para realizar cualquier reclamación en sus pedidos una vez que hayan recibido el producto.
11. Para una experiencia completa, los usuarios tendrán que contar con una cámara en su dispositivo móvil, con el fin de utilizar la función de realidad aumentada. Sin embargo, se podrá utilizar la aplicación sin activar la funcionalidad de realidad aumentada, ya sea porque el dispositivo no cuenta con una o porque el usuario no desea otorgar los permisos a la aplicación.

Requerimientos

Funcionales:

1. La aplicación debe permitir el registro de nuevos usuarios, como vendedores o clientes, además la capturas de sus datos.
2. La aplicación debe permitir que los usuarios registrados puedan iniciar sesión.
3. La aplicación debe permitir que los usuarios recuperen sus cuentas si olvidaron su contraseña.
4. La aplicación debe permitir a los usuarios vendedores la publicación de nuevos artículos.
5. La aplicación debe permitir a los usuarios vendedores la modificación de artículos ya publicados.
6. La aplicación debe permitir a los usuarios vendedores la eliminación de artículos publicados.
7. La aplicación debe permitir registrar todas las características posibles de un producto.
8. La aplicación debe mostrar todos los artículos publicados al usuario cliente.
9. La aplicación debe mostrar los artículos relacionados a la búsqueda del usuario cliente tomando en cuenta el filtro que aplique.
10. La aplicación debe mostrar los artículos seleccionados para la compra por el usuario cliente.
11. La aplicación debe mostrar los artículos seleccionados para como favoritos por el usuario cliente.
12. La aplicación debe mostrar el estado de envío de un producto.
13. La aplicación debe ofrecer el pago en efectivo o tarjeta.
14. La aplicación debe permitir la generación del voucher de pago o permitir la captura de la tarjeta bancaria que utilizará el usuario cliente para la compra.

15. La aplicación debe permitir la modificación o adición de la dirección de envío del usuario cliente.
16. La aplicación debe implementar la funcionalidad de realidad aumentada para los artículos en los que el usuario vendedor desee.
17. La aplicación debe permitir dentro de la funcionalidad de realidad aumentada fijar el producto en algún espacio del mundo real visto a través de la cámara.
18. La aplicación debe solicitar los permisos necesarios para el uso de la cámara y el carrete de fotos del usuario cliente.
19. La aplicación debe ser intuitiva para el usuario.

No funcionales:

1. La aplicación debe soportar una gran cantidad de usuarios, es decir, debe ser diseñada lo suficientemente flexible para ser escalable.
2. La aplicación debe desarrollarse con tecnologías que se encuentren en constante actualización.
3. La aplicación debe desarrollarse en lenguajes de programación nativos de cada plataforma como Java o Kotlin para Android y Swift u Objective-C para iOS.
4. La aplicación debe ser mantenible a lo largo del tiempo.
5. La aplicación debe cifrar y descifrar la información de los usuarios para brindar confidencialidad en sus datos.
6. La aplicación debe comprobar el envío y recepción de datos con tal de brindar integridad de los datos.
7. La aplicación debe ofrecer alto rendimiento en la aplicación en general pero sobre todo para la función de realidad aumentada.
8. La aplicación debe estar bien documentada.
9. La aplicación debe asegurar el resguardo de los datos de las tarjetas bancarias de los usuarios.
10. La aplicación debe asegurar una alta disponibilidad que permita a los usuarios ingresar las 24 hrs al día.

Viabilidad

En relación a la funcionalidad principal de la aplicación, que es el uso de la realidad aumentada para mostrar los productos virtuales en el mundo real, la existencia de los algoritmos en los que se basa esta tecnología y el uso de interfaces de programación para los mismos nos permite implementar la funcionalidad de manera sencilla dentro de la aplicación. La computación gráfica, la gran capacidad de la cámara de los celulares actuales, los algoritmos de visión artificial y el almacenamiento en la nube para detectar el

entorno en que se encuentra el usuario son comúnmente utilizados y soportados por la comunidad de desarrolladores en distintas plataformas.

Alcance y producto mínimo viable de la aplicación

Para lanzar esta aplicación en una versión beta, se crearía una interfaz que contendría las funciones principales para cumplir con su objetivo y en posteriores versiones se realizaría una mejora a las funciones para hacerla más atractiva a los usuarios.

No sería posible lanzar la aplicación omitiendo algunas de sus funcionalidades, ya que consideramos que constituyen el núcleo del negocio y omitir alguna de ellas provocaría que no fuera útil, como es el caso del seguimiento de los pedidos o que no fuera lo suficientemente atractivo para los usuarios como sucedería si se omitiera la funcionalidad de la realidad virtual.

Sin embargo, podría restringirse el registro de los vendedores, de manera que inicialmente sea necesario contactarnos con ellos de manera directa para registrar sus productos. Adicionalmente, la aplicación se lanzaría en una región limitada como lo puede ser un municipio de la Ciudad de México.

Funcionalidades y frameworks

Realidad aumentada

En la actualidad existen diversas bibliotecas que implementan los algoritmos de reconocimiento de superficies e iluminación así como de mapeo del entorno, necesarios para la implementación de una aplicación de realidad aumentada. En el caso específico de los dispositivos móviles existe un motor de realidad aumentada para cada una de las plataformas más populares (Android y iOS), en el caso de android el motor es conocido como ARCore mientras que por el lado de iOS existe iOS Arkit.

Android

ARCore: ARCore es el motor creado por Google para crear experiencias de realidad aumentada. ARCore permite que se pueda utilizar el teléfono para analizar y detectar el entorno. Esto quiere decir que el motor analizará el sitio en el que se está utilizando la cámara, utilizando algoritmos de visión artificial, hará un mapa interno del lugar y con esta información implementará la realidad aumentada. (Fernández, 2020)

El motor de ARCore, es capaz de detectar el movimiento en tiempo real para conocer su posición relativa dentro del entorno, de esta manera, podrá detectar cuando la cámara se mueve, de manera que siempre sabrá en qué posición del mapa se encuentra lo cual hace que los modelos se oculten cuando la cámara apunte a otro sitio.

También es capaz de detectar el tamaño y la ubicación de diferentes superficies horizontales, verticales y anguladas.

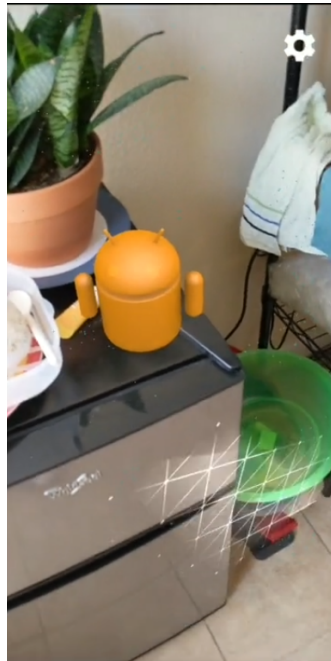


Figura 1. Uso de ARCore (Google, 2021)

Esta tecnología también es capaz de estimar las condiciones de luz del lugar en el que se encuentra el dispositivo, de manera que las imágenes de realidad aumentada que muestre se relacionen con esta luz de la manera más realista posible.

Sceneform:

Sceneform ayuda a renderizar escenas 3D realistas en aplicaciones AR y no AR, sin la necesidad de tener conocimiento OpenGL. Sceneform incluye:

- Una API de gráficos de escena de alto nivel.
- Un renderizador realista basado en la física proporcionado por Filament.
- Un complemento de Android Studio para importar, ver y crear archivos 3D en formatos como obj y fbx.

iOS

Arkit: ARKit es una plataforma de realidad aumentada (AR) de Apple para dispositivos iOS. Permite a los desarrolladores producir aplicaciones que interactúan con el mundo que se encuentran los usuarios, utilizando las cámaras y sensores del dispositivo.

SceneKit: Ayuda con la creación de juegos en 3D y adición de contenido en 3D a las aplicaciones, utilizando descripciones de escenas de alto nivel. También permite añadir fácilmente animaciones, simulación física, efectos de partículas y renderizado físico realista.

RealityKit: Permite utilizar el marco RealityKit para implementar simulación y renderizado 3D de alto rendimiento. RealityKit aprovecha la información proporcionada por el marco ARKit para integrar objetos virtuales en el mundo real.

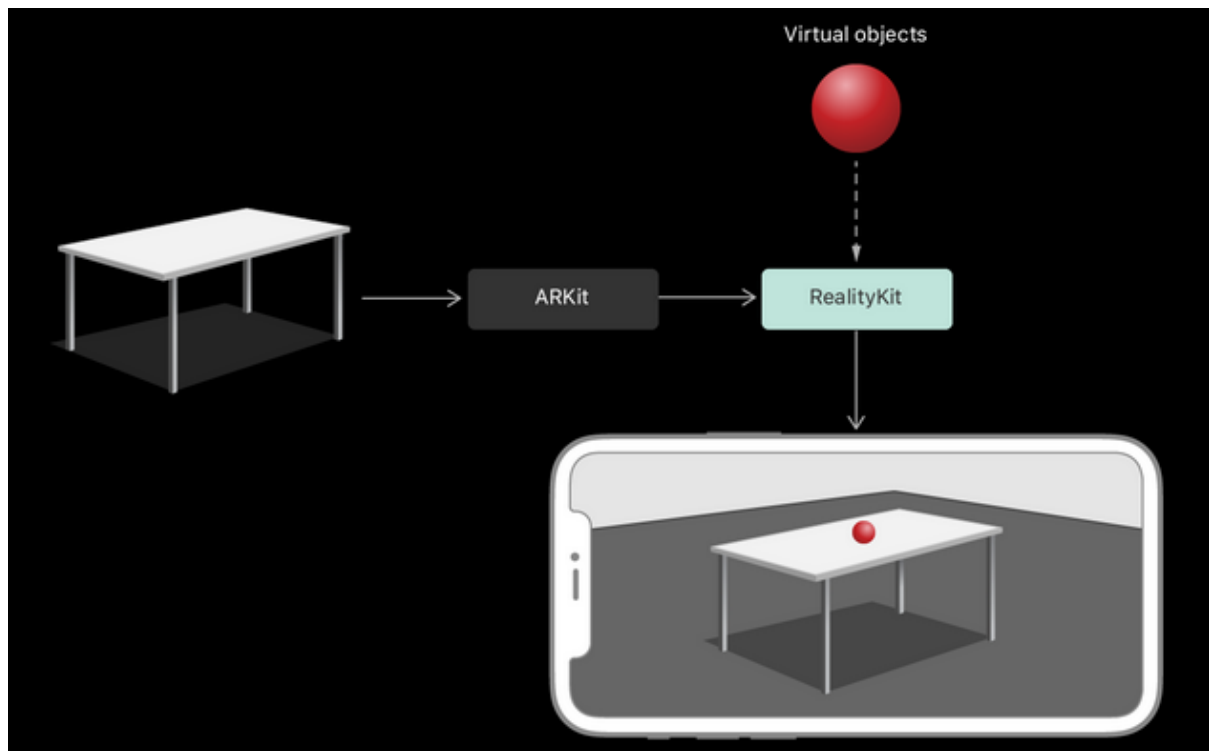


Figura 2. Uso de RealityKit y ARKit (Apple, 2021).

Model I/O: El marco Model I/O proporciona una comprensión a nivel de sistema de los activos del modelo 3D y recursos relacionados. Se puede utilizar este marco para importar y exportar activos desde y hacia una variedad de formatos de archivo estándar de la industria compatibles con herramientas de creación y motores de juegos populares. También se puede utilizar Model I/O para generar o procesar modelos y datos de textura, por ejemplo, para crear superficies de subdivisión, texturas de oclusión ambiental o generar sondas de luz. Model I/O puede compartir búferes de datos con los marcos MetalKit, GLKit y SceneKit para cargar, procesar y renderizar activos 3D de manera eficiente.

Modelo Vista Controlador

Vistas:

- Registro del usuario: Se solicitan al usuario sus datos.

- Inicio de sesión: Se solicitan los datos de ingreso del usuario previamente creados.
- Contraseña olvidada: Se solicita el correo electrónico con el que se registró el usuario.
- Menú de opciones: Se muestran opciones como publicar un artículo, comprar un artículo, modificar datos del perfil, hacer seguimiento de un pedido o iniciar un reclamo, según sea el caso.
- Publicación de artículo: Se solicitan los datos del artículo.
- Listado de artículos: Se listan los artículos existentes o se listan los artículos que consiste con la búsqueda del usuario.
- Detalle de artículo: Se muestra la información detallada del artículo.
- Artículo con realidad aumentada: Se muestra el artículo a través de la captura de la cámara del usuario.
- Perfil de usuario: Muestra la información básica de un usuario.
- Modificación de perfil de usuario: Se solicita al usuario su nueva información.
- Formulario de reclamaciones: Se solicita al usuario el número de pedido y la causa del reclamo.
- Carrito de compras: Se listan los artículos seleccionados por el usuario cliente para la compra.
- Pago con tarjeta o generación de vouchers: Se solicita al usuario su tarjeta de crédito o débito o se le muestra el voucher de pago.
- Detalle del pedido: Se muestra la información del pedido, es decir, los artículos que se adquirieron junto con su detalle.
- Seguimiento del pedido: Se muestra la información de envío del pedido.
- Lista de favoritos: Se muestran los artículos seleccionados por el usuario cliente como favoritos.

Controladores:

- Controlador de pantalla de inicio: Se encargaría de leer el usuario y contraseña en la vista de inicio de sesión para enviárselos a la API y que esta realice la autenticación. Además se encargaría de leer los datos del usuario en la pantalla de registro de usuario y también se encargaría de enviar dichos datos a la API REST para que esta realice el registro. Finalmente se encargaría de enviar una solicitud a la API para recibir una liga para el cambio de contraseña misma que se mostrará en la vista de cambio de contraseña.
- Controlador de lista de artículos: Se encarga de realizar una consulta a la API que regrese los identificadores, nombres y descripción corta de los artículos que se

vayan a recomendar al usuario, además, realiza una consulta similar en el caso de los artículos que sean buscados por el usuario. Modifica la vista de listado de artículos agregando tarjetas con los artículos de la lista de los artículos recuperados en la consulta.

- Controlador de favoritos: Al igual que el controlador de lista de artículos este controlador se encargaría de realizar una consulta mediante la API, utilizando el identificador de usuario (obtenido de una instancia del modelo usuario), obtendría una estructura con los identificadores, nombres y descripción de los productos. Finalmente se encargaría de cargar los datos recuperados en tarjetas que se desplieguen en la vista de lista de favoritos.
- Controlador de detalle de artículo: Carga una instancia del modelo de artículo a la vista de detalle de artículo a la vez que muestra dicha vista al usuario.
- Controlador de perfil de usuario: Se encarga de realizar una petición a la api para modificar los datos del usuario, actualizar el modelo del usuario y también mostrar esos datos en la pantalla de perfil de usuario.
- Controlador de reclamaciones: Se encarga de leer los datos del formulario de reclamación, enviarlo al endpoint correspondiente de la API y mostrar retroalimentación al usuario sobre el estado del registro (exitoso o fallido) de su reclamación.
- Controlador de compra: Se encarga de llevar una referencia de los artículos que el usuario seleccionó para comprar, procesar el pago, realizar el registro del pedido mediante una solicitud a la API.
- Controlador de seguimiento de pedido: Se encarga de crear una instancia del modelo pedido, que a su vez realizará una consulta a la API para recuperar los datos de la instancia de dicho modelo. Finalmente se encargará de llenar una tarjeta con los datos y desplegarla en la vista de seguimiento de pedidos.
- Controlador de vistas: Este controlador se encargará de cambiar o servir las vistas de la aplicación cuando estas no sean el resultado de una operación más compleja (como lo puede ser la compra de una lista de productos). Un ejemplo del uso de este controlador puede ser el menú de opciones el cual puede cambiar a otras vistas como la publicar un artículo, comprar un artículo, modificar datos del perfil, hacer seguimiento de un pedido o iniciar un reclamo.

Modelos:

- Usuario: Este modelo se encarga de realizar la consulta a la API para recuperar los datos de un usuario y generar una instancia u objeto con los datos de éste, que se propaga a varios de los controladores para que otros modelos puedan recuperar los datos correspondientes al usuario.

- Artículos: Este modelo se encarga de realizar la consulta a la API para recuperar los datos de un artículo y generar una instancia u objeto con los datos de éste.
- Pedido: Este modelo se encarga de realizar la consulta a la API para recuperar los datos de un pedido.

Flujo de datos (API, Endpoints, autenticación, seguridad, etc)

Las APIs que se utilizarán pertenecen a AWS o Microsoft Azure de tal manera que se puedan obtener los datos en formato de texto como json o xml o en crudo, es decir en bytes, según sea el caso, para posteriormente ser convertidos al tipo de dato que se requiera. Para los archivos en formato de bytes se serializaría el objeto y se escribiría de manera temporal en la carpeta de la aplicación.

La estructura de nuestros endpoints dependerá de la información que se desea obtener en el flujo de la aplicación, la dirección y estructura en la que serán presentados es la siguiente:

Registro de usuario o modificación del perfil:

endpoint: app.itmvendors/api/user/{user_id}

- id_usuario
- correo_electronico
- contraseña
- nombre
- direccion_de_envio
- direccion_para_envio
- INE_bytes
- telefono
- tarjeta
 - titular_de_tarjeta
 - numero_de_tarjeta
 - expiracion

Inicio de sesión:

endpoint: app.itmvendors/api/login

- id_usuario
- contrasenia

Publicación de un artículo:

endpoint: app.itmvendors/api/item/register

- precio
- descripcion
- características
- imagenes_bytes

Detalle del artículo:

endpoint: app.itmvendors/api/item/{item_id}

- vendedor
- descripcion
- características
- precio
- ubicacion_del_vendedor

Reclamaciones:

endpoint: app.itmvendors/api/order/{order_id}/complaint

- id_pedido
- reclamacion

Carrito:

endpoint: app.itmvendors/api/shopping_cart/{shopping_cart_id}

- articulos_seleccionados_carrito

Favoritos:

endpoint: app.itmvendors/api/user/{user_id}/favorites

- articulos_seleccionados_favoritos

Seguimiento de pedido:

endpoint: app.itmvendors/api/order/{order_id}/

- id_pedido
- precio_total

- estado_envio
- codigo_de_seguimiento
- reclamo

A grandes rasgos, los usuarios se comunican a través de la aplicación para vender y comprar artículos y registrar los datos de su perfil, estas peticiones son enviadas al servidor, respondidas o alojadas según sea el caso. El recibo de venta o compra es generado en el servidor de la aplicación para ser mostrado después a los usuarios involucrados, finalmente la petición de envío se manda a la compañía de paquetería y se mantiene informados del envío de los productos a los usuarios mediante un código de seguimiento.

Ya que es necesario mantener bastantes datos actualizados en todo momento en la aplicación como el precio de los productos, la cantidad en stock de cada producto, notificaciones de seguimiento de envío y la información de las características de cada producto, es indispensable la conexión a internet para utilizar la aplicación.

Esta aplicación utilizará el sensor de la cámara del dispositivo para desplegar la representación virtual de los productos en el mundo real, además para mejorar la experiencia inmersiva se utiliza el giroscopio, que permite mantener la posición del elemento virtual de forma realista en el escenario del mundo real.

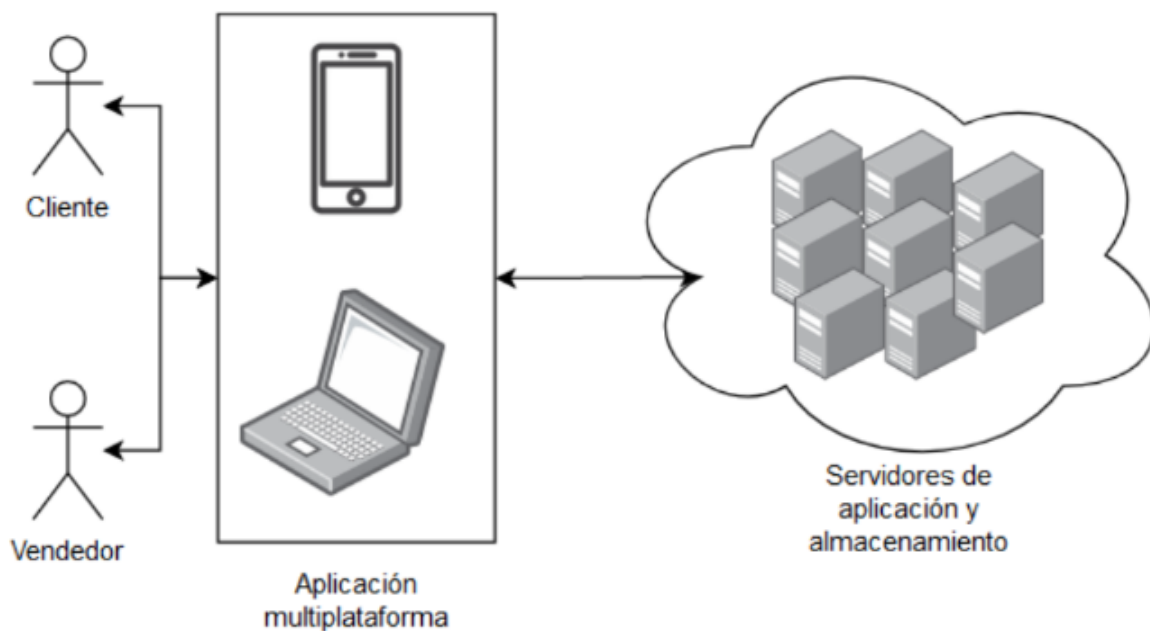


Figura 3. Funcionamiento de la aplicación.

Servicios

Los servicios que utilizará la aplicación serán AWS o Microsoft Azure para el almacenamiento de archivos multimedia, es decir, los modelos 3D así como el despliegue

de la API RESTFUL para asegurar su disponibilidad y el balanceo de carga. También se utilizarán bases de datos en la nube para el almacenamiento de los datos de los usuarios, las compras y los pedidos, a fin de asegurar la escalabilidad; estos servicios pueden ser Amazon Aurora o Azure Cosmos DB, ya que son compatibles con bases de datos relacionales como MySQL y Postgres.

En cuanto al servicio que se utilizará para crear modelos en 3D será ReconstructMe donde a través de un sensor infrarrojo de profundidad (Kinect) se hará la captura del producto en 3D, para esto será necesario contar con el producto que el vendedor ofrezca en su publicación.

Comunicación de datos

Se aprovechará la implementación de los endpoints de las APIs para poder obtener la información de cada componente de nuestro modelo.

Persistencia de los datos

La aplicación servirá con consultas de información a servidores externos, el usuario solo podrá visualizar el flujo normal de la aplicación si tiene conexión a internet. Sin embargo, se utilizará una base de datos local para almacenar la sesión del usuario y no tener que obligar al usuario a volver a ingresar sus datos de login cuando se pierde la conexión a internet o cuando cierra la aplicación.

Para la persistencia de los datos no locales se ocupará el almacenamiento ofrecido por AWS como S3 o por Azure como Azure Storage.

Cuando no hay conexión a los servicios externos que proporcionan el acceso a la base de datos para la consulta de información en texto plano o de almacenamiento, la aplicación mostrará una pantalla que indicará que la conexión ha fallado, la cual no podrá ser cerrada hasta que la conexión a los servicios sea establecida nuevamente.

Seguridad de los datos

Toda la información sensible del usuario será cifrada, únicamente podrá descifrarse una vez que el usuario se encuentre logueado a través de la llave privada almacenada en el dispositivo donde se autentifica el usuario, esta llave privada se genera solamente cuando el usuario comprueba que es quien dice ser con un código de seguridad a través de su número de celular o correo electrónico.

Permisos y requisitos para publicación

Se distribuirá a través de las tiendas de los sistemas operativos que correspondan a cada dispositivo, pero también existirá la posibilidad de descargarla directamente o utilizarla a través de una página web. Se tomarán en cuenta las políticas de cada entorno respecto a la distribución de instalables de aplicaciones, ya que sabemos que en entornos como el de Apple no es posible distribuirlas de esta manera.

Los costos de publicación en las tiendas de aplicaciones son: 25 USD para la Play Store, 99 USD en la AppStore para iPhone/iPod/iPad así como en su versión de MacOS. Este precio se mantiene también para la publicación de la aplicación en la Microsoft Store. Además, el alojamiento del sitio web tendría un costo de 10 USD (Pireddu, 2019).

Para poder hacer el registro de la aplicación en las tiendas de aplicaciones es conveniente la creación de la empresa en sí, además de la creación de un correo electrónico corporativo para que en el registro se coloque dicho correo y el nombre de los desarrolladores con la empresa registrada como ITM VENDORS.

Se puede considerar que al hacer el uso de la realidad aumentada, se verificará que el dispositivo cuente con las últimas actualizaciones de software para un funcionamiento óptimo. En el caso de dispositivos Apple, cuenten con iOS 15.1 o superior, mac OS 12.0 o superior, Android 4.4 o superior, Windows 8.1, 10 u 11 y distribuciones con kernel Linux. La arquitectura debe ser del procesador de 64 bits.

Lenguajes de programación y servicios

- Android:

Lenguajes de programación: Java o Kotlin.

SDK: ARcore.

- iOS:

Lenguajes de programación: Swift u Objective-C.

SDK: Arkit.

- Servicios en la nube: AWS o Microsoft Azure.

- Bases de datos: Postgres, Mysql, SQLite.

Conclusiones

El realizar este trabajo nos dió una idea muy clara de lo que conlleva elaborar una aplicación enfocada al comercio electrónico, se realizó una investigación profunda para comprender el funcionamiento de herramientas de desarrollo que nos proporcionarán la funcionalidad principal de la aplicación, es decir, la realidad aumentada. La motivación principal para el desarrollo de la aplicación de ITM VENDORS fue que notamos que a pesar de que el ecommerce se ha popularizado y que existe una cantidad considerable de empresas que se encuentran enfocadas a este sector, no existe una herramienta que permita visualizar el producto que se va adquirir, lo cual brinda un valor agregado a la aplicación, ya que el usuario se da una mejor idea de cómo es el producto a través de cómo se vería en la vida real.

Concluimos que el realizar una aplicación no es nada fácil ya que se necesita de una planeación completa de varios aspectos como por ejemplo: qué hará la aplicación, a quién será dirigida, a qué segmento pertenecerá, problemáticas regulatorias, el diseño de las pantallas, el uso de servicios externos, la creación de servicios que ofrecerá la aplicación,

número de personas para el equipo de trabajo, presupuesto, almacenamiento, escalabilidad, compatibilidad de plataformas, disponibilidad, mantenibilidad, etc, es por ello que se necesita de varias áreas para abordar todos estos aspectos y por lo tanto de diversos especialistas de cada una de ellas para tener un producto óptimo y agradable al mercado y los consumidores.

Referencias

- [1] eSemanal. (2019, 23 mayo). *Las reglas del comercio electrónico en México*. Recuperado 15 de diciembre de 2021, de <https://esemanal.mx/2019/05/las-reglas-del-comercio-electronico-en-mexico/>
- [2] Fernández, Y. (2020, 31 marzo). *ARCore: qué es y cómo saber si tu móvil es compatible*. Xataka. Recuperado 15 de diciembre de 2021, de <https://www.xataka.com/basics/arcore-que-como-saber-tu-movil-compatible>
- [3] Google. (2020, 7 abril). *Sceneform overview*. Google Developers. Recuperado 15 de diciembre de 2021, de <https://developers.google.com/sceneform/develop>
- [4] Tillman, M. (2021, 27 enero). *Apple ARKit explicó: Todo lo que necesitas saber sobre la plataforma de realidad aumentada de Apple*. Pocket-lint. Recuperado 15 de diciembre de 2021, de <https://www.pocket-lint.com/es-es/ra-y-rv/noticias/apple/141615-kit-de-apple-ar-explicado>
- [5] Apple. (2021). Apple Developer Documentation. SceneKit Framework. Recuperado 15 de diciembre de 2021, de <https://developer.apple.com/documentation/scenekit/>
- [6] Apple. (2021). *Apple Developer Documentation*. RealityKit Framework. Recuperado 15 de diciembre de 2021, de <https://developer.apple.com/documentation/realitykit/>
- [7] Apple. (2021). *Apple Developer Documentation*. Model IO Documentation. Recuperado 15 de diciembre de 2021, de <https://developer.apple.com/documentation/modelio>
- [8] Google. (2021). *Quickstart for Android | ARCore* |. Google Developers. Recuperado 15 de diciembre de 2021, de <https://developers.google.com/ar/develop/java/quickstart>