

DETALLE TÉCNICO DESARROLLO

ITM VENDORS

Anaya Ruiz Yair Alejandro
Jimenez Ruiz Gustavo Alfredo
Santander Martínez Ángel Antonio



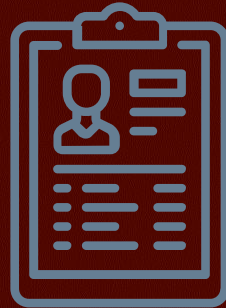


01 | **Reglas de negocio**

-
1. El usuario deberá crear una cuenta en la aplicación.
 2. Es indispensable agregar una dirección de hogar/oficina para entrega de artículos.
 3. Si las compras serán con tarjeta, esta no importa si es crédito o débito, solo que sea válida.
 4. Todo artículo para vender deberá tener al menos 2 fotografías en sus características, así como un modelo 3D o artículo entregado a la empresa para su herramienta en realidad aumentada: Esta característica genera un precio extra al vendedor.
 5. Si el método de pago es efectivo, el pago debe hacerse dentro de las próximas 48 hrs. después de su confirmación.
 6. Todo artículo debe contar con descripción y características específicas del mismo.
 7. Toda publicación será revisada y aprobada con el fin de brindar seguridad y respeto entre usuarios.
 8. Todo vendedor deberá asegurar seguimiento en garantías, dudas y devoluciones.
 9. Mantener actualizado cualquier cambio en el producto.

10. El cliente tiene 15 días para realizar reclamación sobre un producto después de su entrega.

11. Para una mejor experiencia, se recomienda al usuario contar con una cámara para la herramienta de realidad aumentada, aunque no es obligatorio el uso de la misma.





02

Requerimientos funcionales y no funcionales

Funcionales

- La aplicación debe permitir el registro de nuevos usuarios, como vendedores o clientes, además la capturas de sus datos.
- La aplicación debe permitir que los usuarios registrados puedan iniciar sesión.
- La aplicación debe permitir que los usuarios recuperen sus cuentas si olvidaron su contraseña.
- La aplicación debe permitir a los usuarios vendedores la publicación de nuevos artículos.
- La aplicación debe permitir a los usuarios vendedores la modificación de artículos ya publicados.

- La aplicación debe permitir a los usuarios vendedores la eliminación de artículos publicados.
- La aplicación debe permitir registrar todas las características posibles de un producto.
- La aplicación debe mostrar todos los artículos publicados al usuario cliente.
- La aplicación debe mostrar los artículos relacionados a la búsqueda del usuario cliente tomando en cuenta el filtro que aplique.
- La aplicación debe mostrar los artículos seleccionados para la compra por el usuario cliente.

Funcionales

- La aplicación debe mostrar los artículos seleccionados para como favoritos por el usuario cliente.
- La aplicación debe mostrar el estado de envío de un producto.
- La aplicación debe ofrecer el pago en efectivo o tarjeta.
- 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.
- La aplicación debe permitir la modificación o adición de la dirección de envío del usuario cliente.

- La aplicación debe implementar la funcionalidad de realidad aumentada para los artículos en los que el usuario vendedor desee.
- 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.
- La aplicación debe solicitar los permisos necesarios para el uso de la cámara y el carrete de fotos del usuario cliente.
- La aplicación debe ser intuitiva para el usuario.

No Funcionales

- La aplicación debe soportar una gran cantidad de usuarios, es decir, debe ser diseñada lo suficientemente flexible para ser escalable.
- La aplicación debe desarrollarse con tecnologías que se encuentren en constante actualización.
- 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.
- La aplicación debe ser mantenible a lo largo del tiempo.
- La aplicación debe cifrar y descifrar la información de los usuarios para brindar confidencialidad en sus datos.
- La aplicación debe comprobar el envío y recepción de datos con tal de brindar integridad de los datos.
- La aplicación debe ofrecer alto rendimiento en la aplicación en general pero sobre todo para la función de realidad aumentada.
- La aplicación debe estar bien documentada.
- La aplicación debe asegurar el resguardo de los datos de las tarjetas bancarias de los usuarios.
- La aplicación debe asegurar una alta disponibilidad que permita a los usuarios ingresar las 24 hrs al día.



Viabilidad

Realidad aumentada:

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.



Alcance mínimo

Versión beta con todas funciones principales que cumplen con lo establecido, haciendo posteriores actualizaciones y mejoras.

Se pensaría en un lanzamiento en una región limitada como prueba inicial.



03 | **Funcionalidad 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. 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.

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

ios

Arkit: ARKit: 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.

Realitykit: Implementa simulación y renderizado 3D de alto rendimiento.

Model I/O: El marco Model I/O proporciona una comprensión a nivel de sistema de los activos del modelo 3D y recursos relacionados.



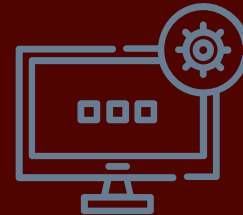


04 | **Modelo vista controlador**

Vista

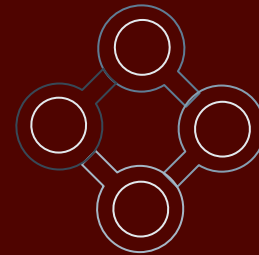
- **Registro del usuario:** Se solicitan al usuario sus datos.
- **Inicio de sesión:** Se solicitan los datos de ingreso del usuario previamente creados.
- **Contraseña olvidada:** Se solicita el correo electrónico con el que se registró el usuario.
- **Menú de opciones:** Se muestran opciones como publicar 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.



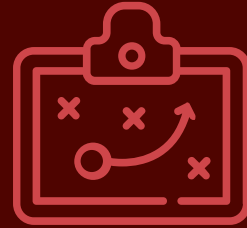
Controladores

- **Inicio de pantalla:** lee usuario y contraseña (relacionado a los datos).
- **Vista de artículos:** Consulta API que regresa información de los mismos.
- **Favoritos:** Consulta API , obtiene identificadores, nombre y descripción de artículos en esta lista.
- **Detalle de artículo:** Carga modelo a la vista de detalle de artículo, mostrando a usuario.
- **Perfil de usuario:** Modifica datos de usuario.
- **Reclamaciones:** Lee datos de formulario de reclamación, envía al endpoint de API y muestra retroalimentación.
- **Compra:** Lleva referencia de artículos que fueron seleccionados para compra, procesa pago y realiza registro de pedido.
- **Seguimiento de pedido:** Crea instancia de modelo de pedido, consultando a API para recuperar datos del modelo. Llena una tarjeta con datos y muestra a usuario.
- **Vistas:** Cambia o sirve las vistas de aplicación cuando estas no sean resultado de operación compleja.



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.





05

Flujo de datos

Registro y modificación de usuario

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 artículo

endpoint: [app.itmvendors/api/item/register](#)

- ❑ precio
- ❑ descripcion
- ❑ características
- ❑ imagenes_bytes

Detalle de 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



06

**Servicios, seguridad,
comunicación y
persistencia de datos,
permisos y requisitos**

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 datos

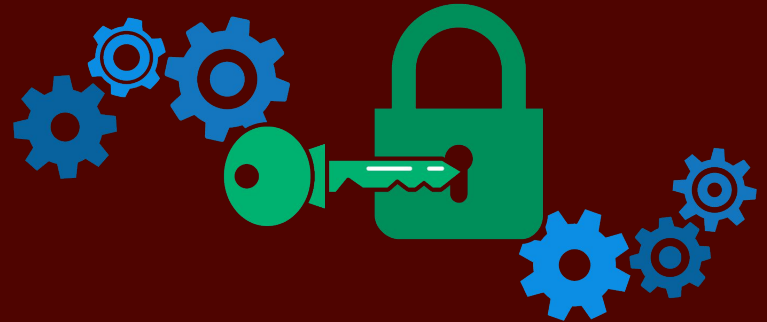
La aplicación servirá con consultas de información a servidores externos, el usuario solo podrá visualizar el flujo normal de la aplicación si tiene conexión a internet. Sin embargo, se utilizará una base de datos local para almacenar la sesión del usuario y no tener que obligar al usuario a volver a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

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.

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)

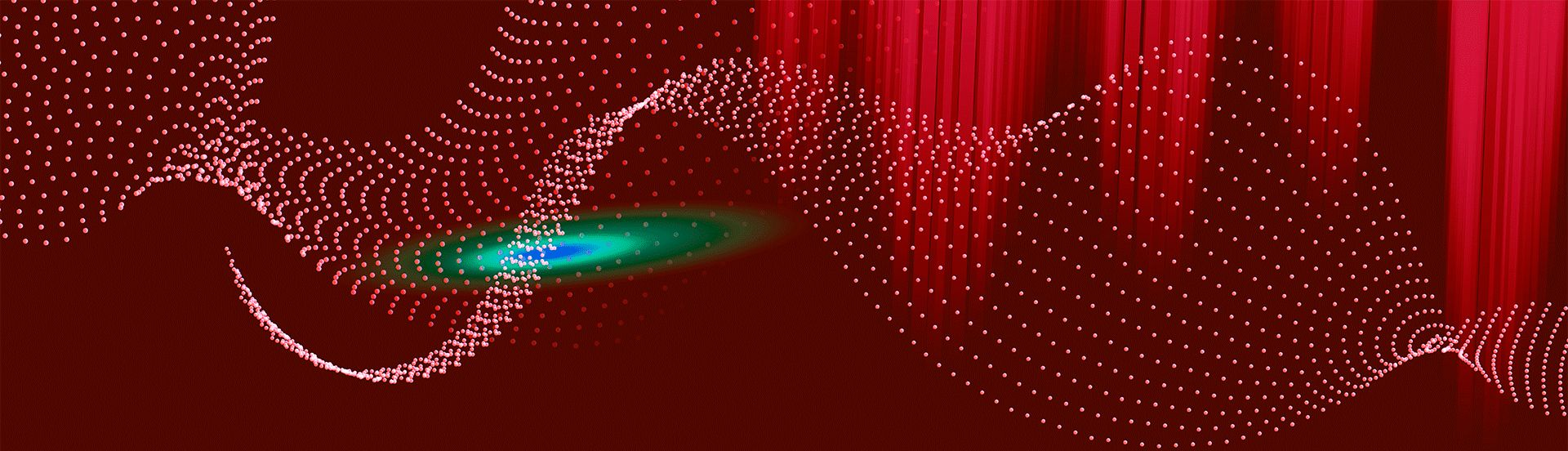
Para poder hacer el registro de la aplicación en las tiendas de aplicaciones es conveniente la creación de la empresa en sí, usando un correo corporativo.

Software actualizado: 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.





07

Conclusiones

Conclusiones

Oportunidad en e-commerce para mejorar una aplicación con una herramienta innovadora.

Hacer una aplicación no es un trabajo fácil, pues requiere de disciplina y compromiso con los clientes que apoyan la misma.

La competencia no es mala, pues nos obliga a seguir actualizando

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-vr/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>