

**Universidad de Ingeniería y Tecnología**  
**Departamento de Ciencias de la Computación**

**Ingeniería de Software I**

**Lab II: Requirements**

**2024 - II**

<b>TEAM</b>	<b>Los del Fondo</b>
<b>INTEGRANTES</b>	<ul style="list-style-type: none"><li>- Alonso Rodríguez ~ 202220266</li><li>- Yesenia Valero ~ 202120584</li><li>- Jose Bojorquez - 201910004</li><li>- Gonzalo Suárez ~ 201910306</li><li>- Piero Guerrero ~ 202210505</li><li>- Aarón Timaná ~ 202220350</li></ul>

**Background:**

Necesidad vs Requerimiento  
Use Case / Edge Case  
Requerimientos Funcionales y No Funcionales  
Usuario Modelo  
Requerimientos Priorizados

## Problema

Juan creó un negocio hace 2 años para venta de abarrotes y él se encargaba de todas las áreas desde inventario hasta contabilidad. Y está llevando todo el registro de ventas en un archivo excel que le ayuda su sobrino Pedro que está en quinto ciclo de Ciencias de la Computación en UTEC.

Pedro está llevando su curso de Ingeniería de Software. Y reúne a sus amigos y deciden ayudar a su tío Juan implementando un sistema para su tienda de abarrotes.

## Preguntas

### **1. Detalle cuál es la necesidad que tiene Juan en el caso de estudio ?**

Juan necesita un sistema para llevar un registro de ventas de sus negocio que contenga una base de datos que le permita gestionar su inventario de manera eficiente. Debe poder registrar cada producto con sus atributos correspondientes, además de actualizar automáticamente el stock con cada venta o cuando se añade un producto, para así poder calcular el total de cada venta de manera precisa.

### **2. ¿Qué use cases podría tener este sistema ? Escriba al menos 10 use cases ?.**

1. Registrar cada producto
2. Actualizar el stock del inventario
3. Eliminar un producto
4. Calcular el total de las ventas
5. Realizar transacciones sin ningún conflicto
6. Administrar usuarios
7. Controlar finanzas de la tienda
8. Gestionar ofertas en tiempos limitados.
9. Gestionar clientes
10. Registrar proveedores de productos

### **3. ¿Existe algún Edge Case ? Escriba al menos 3 edge cases ?.**

1. Que se duplique un producto ya existente.
2. Que se genere un cálculo incorrecto.
3. Que se actualice el stock incorrectamente
4. Que se elimine un producto al mismo tiempo que se realiza una venta
5. Que se interrumpa la conexión al actualizar datos.

**4. Detalle los requerimientos funcionales. Al menos 12 requerimientos.**

1. Registro detallado de ingresos y egresos en la tienda.
  2. El sistema debe permitir registrar productos en el inventariados (solo a usuarios, no a clientes)
  3. El usuario debe poder editar el contenido de estos.
  4. Debe haber un reporte de ventas según periodos predefinidos.
  5. Se debe actualizar la gestión y almacén de los productos ingresados después de la adquisición del proveedor.
  6. Debe alertar si hay poco stock en un producto.
  7. Que haya copias de seguridad recurrentes de la base de datos. También sería bueno redundancia
  8. Interfaz amigable para usuarios y clientes
  9. Debes poder crear y editar ofertas
  10. Se debe registrar detalladamente la compra de productos realizados con el proveedor
  11. Debes poder registrar ventas realizadas en la tienda
  12. Se deben gestionar roles para habilitar permisos en la página

**5. Detalle los requerimientos NO funcionales. Al menos 5 requerimientos.**

1. El programa debe poder funcionar en cualquier dispositivo con internet.
  2. Debe ser escalable según el número de usuarios y productos agregados
  3. El tiempo de respuesta debe ser menor a 2 segundos (latencia, carga de multimedia, apis, etc)
  4. Debe tener un sistema de autenticación confiable para que no haya problemas con la integridad de las cuentas y datos.
  5. Debe evitar errores con la base de datos y que sea seguro

**6. Existen algunas metricas que justifiquen la solucion que esta planteando a Juan. De ser asi sustentelas.**

1. Las operaciones realizadas por los usuarios deben demorar poco. El control de inventario, gestión de reportes, etc. Es decir, debe demorar menos que como se hacía previamente.
2. Reducir errores. Como se hacían manualmente, esto debería ser capaz de reducir el error al ser digitales y con parámetros establecidos.
3. Mejorar el manejo del stock. Al poder tener un stock cuantificado actualizado constantemente, evitará errores como el no recordar el stock en cierto tiempo o confusiones de cual era el stock actualizado y cuando fue realizado el cambio.

**7. Quien es el usuario modelo ? Como haria la solución mas generica y no solo para Juan ?**

Al ser un producto realizado para Juan, él es el usuario modelo. Es una persona con conocimientos básicos en tecnología y necesita un sistema para gestionar su tienda y productos. Para que esto se convierta en una solución más genérica, se necesita personalizar más la interfaz y reportes para que puedan ser utilizado por personas con pocos o nulos conocimientos en tecnologías emergentes y demás. Además, podrías orientar la página a negocios diferentes como panaderías, restaurantes, negocios de entretenimiento, etc.

**8. Priorice los requisitos detallados en la pregunta 4 y 5.**

Funcionales:

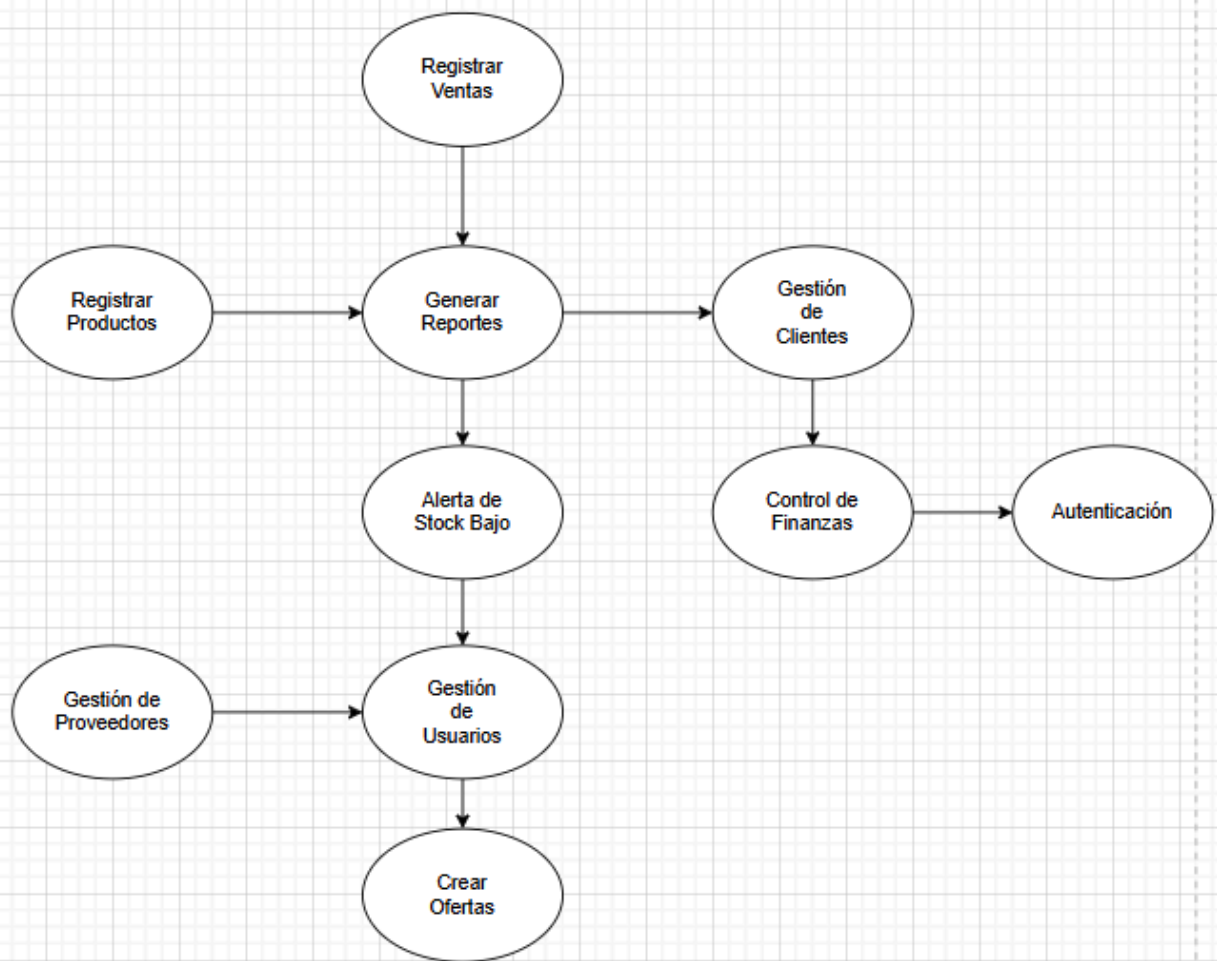
1. El sistema debe permitir registrar productos en el inventariados (solo a usuarios, no a clientes)
2. El usuario debe poder editar el contenido de estos.
3. Debe alertar si hay poco stock en un producto.
4. Que haya copias de seguridad recurrentes de la base de datos. También sería bueno redundancia
5. Interfaz amigable para usuarios y clientes
6. Que haya registro detallado de ingresos y egresos en la tienda.
7. Se deben gestionar roles para habilitar permisos en la página
8. Debes poder crear y editar ofertas

9. Debes poder registrar ventas realizadas en la tienda
10. Se debe registrar detalladamente la compra de productos realizados con el proveedor
11. Se debe actualizar la gestión y almacén de los productos ingresados después de la adquisición del proveedor.
12. Debe haber un reporte de ventas según periodos predefinidos.

No Funcionales:

1. El programa debe poder funcionar en cualquier dispositivo con internet.
2. El tiempo de respuesta debe ser menor a 2 segundos (latencia, carga de multimedia, apis, etc).
3. Debe evitar errores con la base de datos y ser seguro.
4. Debe ser escalable según el número de usuarios y productos agregados
5. Debe tener un sistema de autenticación confiable para que no haya problemas con la integridad de las cuentas y data.

**9. Elabore un diagrama donde se muestra el posible orden de implementación. Use un DAG.**



**Ejemplo Diagrama:**

