

UACM

Universidad Autónoma
de la Ciudad de México

NADA HUMANO ME ES AJENO

Diseño de Software

Punto de venta de una librería

Romero Cervantes Fátima Daniela

21-011-0755

Ciclo Escolar: 2025-I

Índice

Introducción.....	2
Objetivo General.....	2
Alcance en términos de calidad del sistema.....	3
Misión de Calidad.....	3
SRS.....	4
Requisitos funcionales.....	4
Requisitos no funcionales.....	5
Diagrama de casos de uso.....	6
Diagrama de clases.....	7
Métricas de calidad de software.....	8
Procesos de SQA.....	9
Estrategia de SQA.....	9
Actividades de SQA.....	9
SQAP.....	10
Tabla C.3.....	10
Tabla C.4.....	11
Riesgos.....	12
Checklist.....	12
Probabilidad del riesgo e impacto.....	19
Control.....	21
Explicación del control.....	21
Estrategias de Riesgos.....	22
Estrategias de Riesgos (explicación).....	23
Pruebas.....	25
Alcance de pruebas.....	25
Historial de versiones.....	26
Glosario.....	26

Introducción

Estamos comprometidos a ofrecer un software de alta calidad que satisfaga las expectativas y necesidades de nuestros clientes. Para asegurar su correcto funcionamiento, realizaremos diversas pruebas exhaustivas, garantizando que la aplicación esté siempre lista para usarse cuando sea necesario.

El punto de venta para librerías está diseñado para optimizar la gestión de ventas e inventario, especialmente en el contexto de ferias del libro. Con un enfoque en la calidad, sometemos nuestro software a rigurosas pruebas y auditorías, siguiendo las normas ISO y estándares de la industria para garantizar su eficiencia y fiabilidad.

El sistema ofrece una gama completa de funcionalidades, incluyendo:

- Control de ventas e inventario en tiempo real.
- Generación de reportes financieros.
- Gestión de precios y descuentos.
- Adaptación a diferentes ferias del libro.
- Múltiples opciones de pago.

Con este punto de venta, los gerentes de librerías podrán disfrutar de una mejor organización, una atención al cliente optimizada y un control total sobre sus operaciones en las ferias.

Objetivo General

Implementar un sistema de punto de venta de una librería que permita gestionar ventas, inventarios, reportes financieros, conocer precios y descuentos que tiene cada título. Dicho punto de venta será utilizado para llevar el control en las ferias del libro a las que asista la editorial, por lo que los títulos podrían variar según la feria, de igual manera se podrá elegir el nombre o lugar de la feria y llevar el control de las ventas en esa feria en particular. De esta manera se podrá garantizar una buena organización para el gerente de la librería.

Alcance en términos de calidad del sistema

Para garantizar la calidad tanto en su diseño como en su uso en el entorno real realizaremos las siguientes actividades:

Validación de requisitos: Asegurarnos de que la aplicación cumpla con las necesidades del cliente, y en caso de tener alguna duda con respecto a sus requerimientos nos pondremos en contacto con el cliente para resolverlas.

Pruebas de usabilidad: Se realizarán pruebas de usabilidad para cerciorarnos de que el software funcione de la manera que se espera, y para garantizar que el producto tenga calidad se harán diferentes pruebas y en caso de hallar errores estos se deberán corregir y documentar.

Aseguramiento de la calidad: Implementar auditorías de software para garantizar la estabilidad del sistema. El sistema deberá funcionar sin ningún problema especialmente en horarios de trabajo, el mantenimiento se le dará a la aplicación ya sea de noche o cuando no esté en uso.

Pruebas de aceptación: Definir criterios de aceptación claros para la entrega final (por ejemplo, un tiempo de respuesta máximo y un porcentaje mínimo de disponibilidad).

Misión de Calidad

La misión del proyecto es llevar una buena organización en cuanto al inventario de los títulos, también tiene la intención de brindar una atención de calidad al cliente manteniéndose óptimo el sistema. El principal objetivo es gestionar las ventas, saber la cantidad que se ha vendido, los títulos que se han vendido o los que ya están agotados en tiempo real, y brindar diferentes formas de pago. Sin embargo, también podrán consultar los precios de cada título, así como el número de existencias que se tienen en ese momento.

SRS

Requisitos funcionales

- **Verificador de precios:** La aplicación tendrá un apartado para verificar el precio de cada título, de igual manera, se mostrará si este tiene algún descuento y en caso de tenerlo podrá visualizar el precio de lista y el precio con el descuento. Esto se podrá hacer al escanear el código de barras que tiene cada libro, por título o por ISBN. Cabe mencionar que al momento de ver los precios, también le mostrará el número de existencias que se encuentran disponibles.
- **Buscador:** El usuario podrá buscar un libro ya sea por título o por autor; en caso de ser por autor entonces va aparecer una lista con todos los títulos de ese autor. Posteriormente podrá seleccionar el libro para ver el precio y si se encuentra listado en el lugar en el que se encuentra (es decir, si el título fue llevado a la feria del libro).
- **Seleccionar feria:** Debido a que el punto de venta será utilizado en las ferias del libro tendrá la opción de seleccionar una feria, dependiendo de la feria en la que se encuentre son los títulos que aparecerán en la base de datos.
- **Descargar títulos:** La base de datos tendrá una lista de libros que maneja la editorial, sin embargo, solo se descargan los títulos que se encuentran disponibles en cada feria.
- **Subir ventas a base de datos:** Las ventas serán registradas por día, aquí se podrá visualizar la cantidad de dinero que se vendió por día y serán subidos a una base de datos que registrará las ventas por día.
- **Registrar una venta:** Al momento de vender un libro se deberá registrar mediante el código de barras, ISBN, o título; entonces se va a marcar la cantidad de títulos vendidos y el total que tienen que pagar.
- **Método de pago:** Al momento de registrar una venta se podrá elegir el método de pago, este puede ser con tarjeta, en efectivo, o combinado. En caso de pagar en efectivo el cajero podrá indicar la cantidad con la que están pagando y la aplicación les dirá cuánto debe de regresar de cambio. De igual manera, si el pago es combinado, se deberá indicar cuánto va a pagar con tarjeta y la aplicación le indicará la cantidad que deberá pagar en efectivo.
- **Facturar:** Debido a que hay muchas personas que desean facturar a la hora de realizar una compra, entonces el cajero deberá pedir los datos a la persona que desee facturar y de inmediato le empezará a crear su factura.
- **Eliminar venta:** En caso de registrar una venta mal o simplemente de querer eliminar alguna venta tendrá que poner la contraseña del administrador, de lo contrario no podrá realizar tal cambio.

- **Buscar venta:** La venta podrá ser buscada mediante el ISBN de cada libro, por el título del libro o bien por el número de venta. Dentro de eso podrá ver cuántos ejemplares se vendieron, la cantidad de dinero que se vendió y el método de pago.

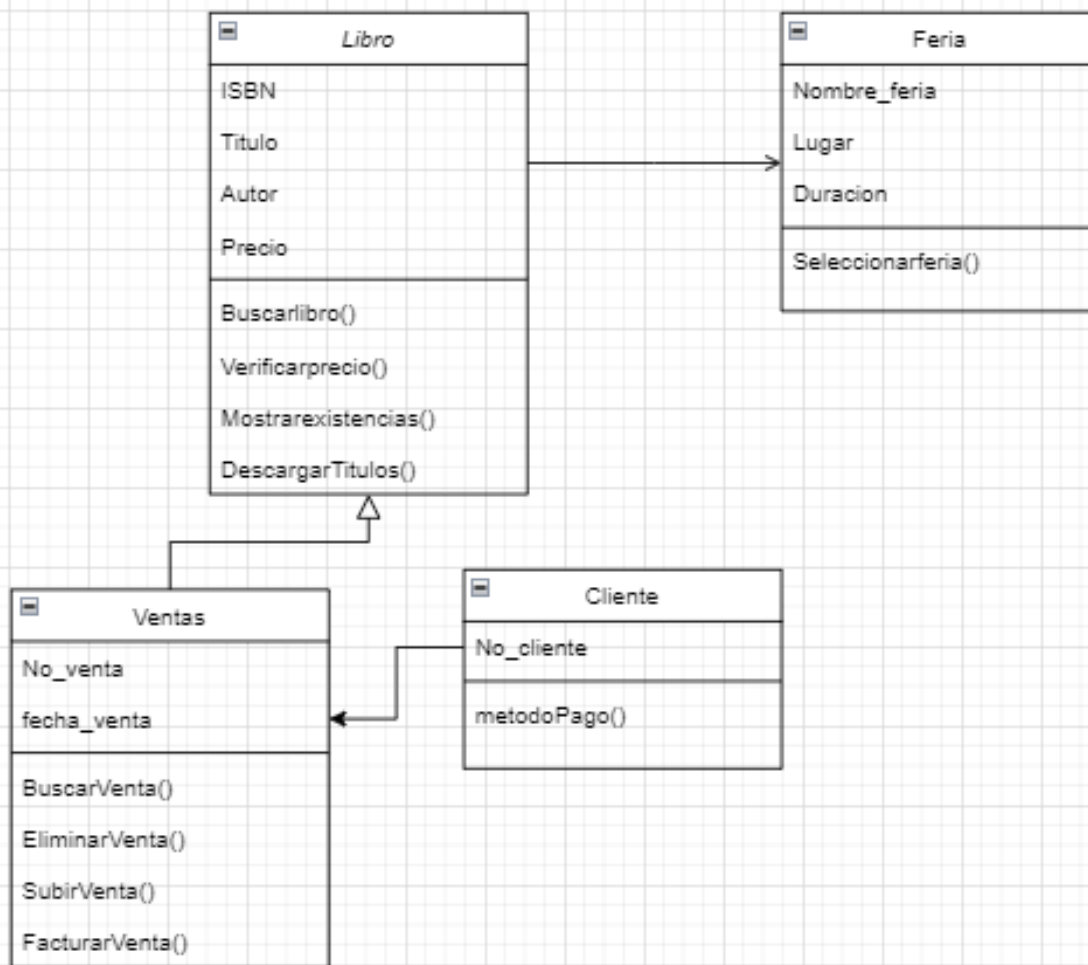
Requisitos no funcionales

- **Funcionalidad:** El sistema tendrá múltiples formas de pago, siendo estas pago con tarjeta, en efectivo y combinada. De igual manera, el inventario se tendrá que actualizar en tiempo real, con el fin de evitar problemas a la hora de hacer el corte.
- **Eficiencia:** El sistema no tiene que tardar más de 3 segundos al momento de realizar una consulta o al hacer una transacción.
- **Fiabilidad:** La aplicación estará disponible el 99.9% del tiempo, también contará con una copia de seguridad en caso de que el software llegue a presentar fallos, la información estará segura. Y en caso de que la aplicación falle aparecerá un mensaje con relación al problema.
- **Usabilidad:** La interfaz será intuitiva para que los trabajadores puedan usar la aplicación de manera sencilla y sin la necesidad de someterse a una larga capacitación.
- **Seguridad:** Para evitar que cualquier empleado pueda modificar la información del sistema será necesaria la autenticación de roles, y para hacer alguna modificación se requerirá uso de una contraseña. Los datos sensibles serán cifrados para garantizar su seguridad.
- **Mantenibilidad y Portabilidad:** El sistema estará disponible para windows y android. De igual manera, debe de ser capaz de integrarse junto con las bases de datos que vamos a utilizar tanto con el sistema de facturación que vamos a emplear.
- **Compatibilidad:** Debe de ser capaz de integrarse junto con las bases de datos que vamos a utilizar tanto con el sistema de facturación que vamos a emplear.

Diagrama de casos de uso



Diagrama de clases



Métricas de calidad de software

Nombre	Costos	Beneficios	Impacto	Valor objetivo	Factores de calidad	Herramientas
Disponibilidad	Costos de mantenimiento del sistema y servidores que se vayan a usar.	Reduce tiempos de inactividad y pérdidas de ventas.	Alto	99.9% de tiempo activo	Fiabilidad	Monitores de uptime
Tiempo de respuesta	Costos de optimización para el software y hardware requerido.	Al momento de cobrar optimiza el proceso de cobro.	Alto	3 segundos como máximo por cada consulta	Eficiencia	Reloj contador
Tasa de fallos	En caso de que la app falle, tenemos que considerar cuales son las bajas, es decir, cuánto se va a perder.	Evita que la app falle y que haya pérdida en las ventas.	Alto	Puede presentar 2 fallos por cada 1000 consultas	Fiabilidad	Github
Seguridad	Costos para cifrar los datos y almacenar información sensible.	Evita fraudes y pérdidas de información	Alta	0 accesos no autorizados	Seguridad	OWASP

Tabla 1. Conjunto de métricas

Solicitud	Elementos de datos	Cálculo	Interpretación	Consideraciones	Se requiere información
Punto de venta	Tiempo de operativo	$(\text{Tiempo activo} / \text{Tiempo total}) \times 100$	% del tiempo disponible	Asegurar redundancia	Registros del sistema
Transacciones	Tiempo de respuesta	Suma de tiempos de	Tiempo promedio por	Pruebas de carga	Logs de transacciones

		transacción/ Número de transacciones	operación		
Registro de errores	Tiempo de fallos	Suma de tiempos de transacción/ Número de transacciones	Detecta frecuencia de errores	Monitoreo continuo	Reportes de fallos
Auditoría	Intentos fallidos	(Intentos fallidos/ intentos totales)*100	% de los accesos indebidos que puede haber	Monitoreo continuo	Reportes de acceso denegado

Tabla 1. Continuación de Conjunto de métricas

Procesos de SQA

Estrategia de SQA

Para prevenir que el proyecto fracase durante su desarrollo es importante tener en cuenta el presupuesto que se tiene, de igual manera es de vital importancia tener los conocimientos necesarios para la creación del código, así como de sus bases de datos y que éstas puedan funcionar de manera correcta. Mientras que para garantizar su correcto funcionamiento es necesario recurrir a pruebas constantes para ver que la integración del sistema sea el correcto, también es importante estar monitoreando constantemente cada cambio ya sea en cuanto a requisitos, como en implementación del código.

Actividades de SQA

- **Revisiones de requerimientos:** Es de vital importancia que los requisitos hayan quedado en claro, también revisar si es que ya están completos y cubriendo realmente con las necesidades del cliente y sobre todo que estos se puedan implementar sin ningún problema. En caso de tener alguna duda o sugerencia con respecto a esto sería mejor reunirse con el cliente y comunicarlo.
- **Pruebas de usabilidad:** Se podrían realizar diferentes pruebas para garantizar su funcionamiento correcto, es decir, además de las pruebas de escritorio que se hacen usualmente, se podrían hacer pruebas con usuarios

reales, lo cual también nos serviría para saber si realmente la interfaz es intuitiva y podríamos tener una visión más amplia sobre su funcionamiento.

- **Pruebas de software:** Lo ideal sería realizar pruebas tras implementar una nueva función para facilitar encontrar las posibles fallas que pueda tener el código; y monitoreando todas las áreas constantemente para garantizar que se esté realizando de manera correcta.
- **Auditorías de código:** Debemos de hacer revisiones constantes en el código para que sea posible detectar a tiempo problemas de vulnerabilidades de seguridad y de rendimiento. De esta manera también se podrían aclarar dudas que se tengan sobre alguna funcionalidad que tenga el sistema, ya que para programarla debe ser entendida.
- **Auditoría de seguridad:** Es de vital importancia estar monitoreando la seguridad del sistema para protegerlo contra accesos no autorizados y ataques cibernéticos. Y en caso de tener alguna vulnerabilidad esta se debe de estudiar y tomar las medidas necesarias para que el software realmente sea seguro.
- **Informes de pruebas:** Es necesario documentar los resultados de las pruebas, incluso cuando estas presenten un error, en estos casos se deberá buscar una solución para ello e incluirlo en el informe. De igual manera se debe de documentar aquellas pruebas que sean exitosas.
- **Registros de revisiones:** Registrar las revisiones de requisitos y diseño, incluyendo los participantes, los comentarios y las decisiones tomadas. Esto para llevar una buena organización con el equipo de desarrollo y a su vez saber que es lo que está haciendo cada uno.
- **Actas de reuniones:** Documentar las reuniones del equipo de SQA, incluyendo los temas discutidos, las decisiones tomadas y las acciones a seguir. Lo anterior es para estar consciente de los posibles cambios que pudiera tener el software.

SQAP

Tabla C.3

Preguntas	Entradas sugeridas
¿El alcance del proyecto está claramente definido y se entiende bien? Si está bien definido, estamos conscientes de las funciones que podrá realizar el software, así como las posibles limitaciones que puede llegar a tener. Los requerimientos son claros por lo que se entiende bien. ¿El adquirente, la organización, el equipo del proyecto y el equipo	Plan de Adquisición Contrato Concepto de operaciones

<p>de SQA comprenden el rol del SQA en este proyecto? Si, el SQA nos ayuda a definir si el proyecto que se va a entregar tiene la calidad adecuada que todo proyecto requiere.</p> <p>¿Se conocen y documentan adecuadamente los riesgos potenciales del producto? Conforme avanza el proyecto vamos conociendo los riesgos y vulnerabilidades que tiene el proyecto, sin embargo, aún no se empiezan a documentar.</p> <p>¿Se comprenden riesgos potenciales del producto de modo que las actividades de SQA se pueden planificar de manera proporcional al riesgo del producto? Sí, es posible planificar posibles soluciones a los problemas que se lleguen a presentar. El SQA nos ayuda a darnos cuenta de los errores desde antes de su implementación.</p>	
--	--

Tabla C.4

¿Qué regulaciones gubernamentales son aplicables a este proyecto?

Para garantizar al cliente que el sistema es seguro podría ser la protección de datos, y las normas de seguridad de transacciones. Este último para que esté seguro que los pagos sean almacenados correctamente y con la seguridad necesaria que requiere.

¿Qué normas específicas son aplicables a este proyecto?

La ley de protección de datos personales, nos dice que una vez que se concluya la finalidad con la que se recabaron los datos no podrán ser guardados. Esto nos podría ayudar a la hora de realizar un pago con tarjeta asegurando al comprador que sus datos no serán expuestos.

¿Qué documentos de referencia organizacionales (como procedimientos operativos estándar, normas de codificación, plantillas de documentos, etc.) son aplicables a este proyecto?

Plantillas de documentación, guías de aseguramiento de la calidad, como es el iso 730, procedimiento operativos estándar para gestionar las ventas y políticas de seguridad de la información.

¿Qué documentos de referencia específicos del proyecto son aplicables a este proyecto?

Especificación de requisitos del sistema, plan de gestión de riesgos, y plan de pruebas.

¿Se espera que SQA evalúe el cumplimiento de las regulaciones, estándares, documentos organizacionales y documentos de referencia del proyecto aplicables?

Si, el plan de SQA debe garantizar que se cumplan con todos los requisitos funcionales del sistema visto en el iso 830, cuidando que también se cumplan con las normas de calidad vistas en el iso 730.

¿Qué documentos de referencia es apropiado incluir en el SQAP?

Documentos de referencia específicos del proyecto, normas y procedimientos del SQA; como pueden ser los procedimientos de pruebas, revisiones y validaciones de calidad.

Riesgos

Checklist

Ing. de Producto	Amb. de Desarrollo	Restric. de Programa
1. Requerimientos	1. Proceso de desarrollo	1. Recursos
a. Estabilidad: Es poco probable que los requisitos cambien , ya que al ser un punto de venta el software incluye desde un inicio las funcionalidades necesarias que la empresa requiere para tener una buena organización en sus ventas.	a. Formalidad: ✓ Podríamos estar documentando mal las funcionalidades del sistema, por lo que presenta un reto hacerlo de manera correcta.	a. Calendario: Es necesario calendarizar las actividades que se tienen que hacer y también tomar en cuenta el tiempo de entrega para ajustarnos al calendario.
b. Completitud: ✓ Estaremos incluyendo todas las funcionalidades que nuestro cliente nos haya solicitado y las que se requieren en un punto de venta, por lo que el software que estaremos entregando estará bastante completo.	b. Idoneidad: ✓ Desconozco la manera en la que se podrían implementar algunas de sus funcionalidades por lo que se podría dificultar su integración en el software.	b. Personal: El personal que se tiene para el proyecto es demasiado limitado por lo que se podría retrasar el proyecto.

c. Claridad: ✓ Algunos requisitos podrían ser un poco confusos a la hora de implementarlos.	c. Control de proceso: Se tendrá una supervisión constante del proyecto y puedo ser capaz de ajustar mi tiempo a las necesidades del proyecto.	c. Presupuesto: El presupuesto que se tiene para el proyecto es bastante limitado por lo que debemos ajustarnos al mismo para cubrir lo que necesitamos.
d. Validez: Los requisitos cubren con las necesidades del usuario.	d. Familiaridad: ✓ A pesar de que se tiene conocimiento para implementar la mayoría de las funcionalidades hay algunas con las que desconozco por completo	d. Instalaciones: ✓ Las instalaciones no son del todo adecuadas debido a que no cuenta con las herramientas necesarias para desarrollar el proyecto.
e. Factibilidad: Debido a que el software contará con las funciones básicas de un punto de venta, además de algunas funciones extras, cumplirá con las expectativas del cliente.	e. Control de producto: Se van a estar haciendo pruebas continuas para verificar que el producto cumpla con los requisitos del cliente y garantizar de esta manera la entrega de un producto con calidad.	
f. Precedentes: ✓ Podría presentar algunas dificultades para cumplir con todos los requerimientos debido a la inexperiencia del desarrollador en los puntos de venta.		
g. Escala: ✓ Al tener varias funcionalidades presenta un gran reto implementarlas por completo.		

Ing. del Producto	Amb. de Desarrollo	Restric. del Programa
2. Diseño	2. Sistema de desarrollo	2. Contrato
a. Funcionalidad: Dependerá de que se conecte correctamente a la base de datos, ya que allí se almacenarán todos los datos importantes allí.	a. Capacidad: Se cuentan con los conocimientos previos que se necesitan para empezar con un proyecto, por lo que esto no sería un problema.	a. Tipo de contrato: Contrato de trabajo por obra o servicio indefinido.
b. Dificultad: ✓ Puede presentar dificultades a la hora de integrar la base de datos con la interfaz, de igual manera, podría presentar problemas a la hora de actualizar el inventario.	b. Idoneidad: Se conocen las bibliotecas correspondientes que se deben de usar para su implementación, así como los diagramas a realizar para empezar a implementar el proyecto.	b. Restricciones: ✓ Por el momento no existe ninguna restricción, por lo que podría generar un problema con el tiempo ya que este se puede alargar de más.
c. Interfaces: Las interfaces pueden presentar dificultades a la hora de conectar el back con el front, pueden presentar fallos a la hora de conectarlos, sin embargo, es posible resolver los fallos.	c. Fac. de uso: Al usar un lenguaje de programación que ya conocemos nos será más fácil a la hora de empezar a implementar el proyecto, además ya se había realizado un punto de venta con anterioridad lo que facilita el trabajo.	c. Dependencias: En caso de querer contratar subcontratistas estos deberán ser aprobados directamente por el cliente.
d. Desempeño: Se buscará mantener un sistema óptimo para que el tiempo de respuesta por consulta sea mínimo.	d. Familiaridad: Los lenguajes de programación que se van a usar los vengo manejando por lo menos desde hace 3 años.	
e. Fac. de prueba: el	e. Confiabilidad: ✓ Al	

sistema contará con un diseño claro, es decir su interfaz estará bien definida para cada función que se tenga prevista y se buscará hacer pruebas óptimas para acelerar el proyecto.	estar monitoreando constantemente las actividades que se están realizando durante su desarrollo vamos a poder identificar de manera rápida los errores, sin embargo, podría presentar ciertas dificultades en cuanto a rendimiento.	
f. Rest. de Hdw: Estará disponible únicamente para sistemas operativos como IOS y Windows.	f. Soporte al sistema: ✓ Se le estará dando mantenimiento en un rango de tiempo específico para evitar posibles problemas que pueda llegar a tener mientras se esté utilizando.	
g. Soft adquirido: El producto final será responsive para que su diseño pueda adaptarse correctamente ya sea al teléfono o bien en la computadora.	g. Disponibilidad: Se buscará que esté disponible la mayor parte del tiempo, en especial en el tiempo que requiera. Buscando los días que no se usan para hacer su mantenimiento.	

Ing. de Producto	Amb. de Desarrollo	Restric. de Programa
3. Codif. y Pruebas	3. Adm. de Proceso	3. Interf. de Proyecto
a. Factibilidad: ✓ Se puede proponer un código para la implementación del proyecto, pero puede que está no sea la correcta llevando a que utilice más recursos o que el código sea muy extenso y difícil de	a. Planeación: Para llevar una buena organización y por lo tanto planear bien estaremos usando diversas herramientas, entre ellas el diagrama de gantt.	a. Cliente: ✓ El cliente tiene que estar convencido de que es el diseño que quiere y necesita para su aplicación, no obstante, puede que tenga algunas inconformidades con la interfaz lo que llevaría a

leer		cambios constantes.
b. Pruebas Unit.: Para evitar que se nos acumulen demasiados fallos estaremos haciendo pruebas de escritorio constantemente.	b. Organización Proy. Se buscará seguir con la calendarización que se vaya a realizar para implementar en tiempo y forma cada función provocando un impacto positivo en el proyecto en cuanto a la organización.	b. Socios contrac: En caso de que el cliente sea más de uno entonces debe de convencer a todos la interfaz que se vaya a manejar.
c. Codif/Implant:✓ Al momento de realizar la conexión de la base de datos con la interfaz hay probabilidades de que en un principio no funcione de manera correcta.	c. Experien. en Adm: ✓ Administrar un proyecto de la mejor manera presentará un gran desafío al ser la primera vez en jugar ese rol.	c. Subcontratistas: Se tienen que apegar al diseño que el cliente desea y en muchas ocasiones puede generar conflictos internos con estos.
	d. Interfaces proyecto: ✓ Si las interfaces del proyecto no quedan y funcionan como se tiene previsto podría retrasar todo el proyecto.	d. Contrato. principal: En un principio debería de aclararse lo del diseño de las interfaces y su arquitectura para no tener conflictos después.
		e. Dirección gen: Se puede incluir las direcciones externas de manera fácil dentro del código.
		f. Vendedores: Estos deben de aprobar el diseño, una vez que hayan decidido que es el ideal para el tipo de proyecto que se está trabajando.

		<p>g. Política: Estos deberán de tener alguna política de privacidad por la información sensible que se pueda guardar dentro de la interfaz.</p>
--	--	---

Ing. de Producto	Amb. de Desarrollo	Restric. de Programa
4. Integr. y Pruebas	4. Métodos de Administ.	
<p>a. Ambiente: Los cambios a realizar una vez realizadas las pruebas deberían ser menores.</p>	<p>a. Monitoreo: Los monitoreos se harán de manera semanal y posteriormente se podrían realizar mensualmente.</p>	
<p>b. Producto: Si se deduce que el software cumple con los requisitos funcionales que el cliente requiere y siguiendo el iso 730 de calidad se verifica que la tiene, entonces el producto final no debería presentar ningún problema.</p>	<p>b. Adm. personal: Se le asignan actividades de acuerdo a las habilidades del personal.</p>	
<p>c. Sistema: ✓ En caso de que se requieran grandes cambios generaría un gran impacto en el mismo.</p>	<p>c. Aseg. de Calidad: Para asegurarnos que cumpla con esto seguiremos el plan de SQA visto en el isp 730.</p>	
	<p>d. Adm. de Configuración: ✓ Presentará un reto mantener bajo control todas las</p>	

	áreas al mismo tiempo.	
5. Especialidades de Ing	5. Ambiente de trabajo	
a. Mantenibilidad: ✓ Dar un mantenimiento correcto podría presentarnos ciertas dificultades, ya que en muchas ocasiones no sabemos exactamente a qué le debemos dar mantenimiento.	a. Actitud a la calidad: La actitud a manejar está siendo positiva lo que ayuda a la cooperación en caso de tener que modificar algo.	
b. Confiabilidad: Se buscará que el software sea óptimo y que no presente fallas durante su uso.	b. Cooperación: Está en completa disposición de cooperar con lo que se requiera y haga falta.	
c. Seguridad de personas: Como tal no se almacena ninguna información sobre el personal o clientes por lo que no presenta ningún problema.	c. Comunicación: Se mantiene una buena comunicación con el gerente del proyecto, resolviendo de esta manera las dudas que tiene el desarrollador.	
d. Seg. de Acceso: ✓ Mediante un login y ciertas restricciones podríamos asegurar el acceso al punto de venta, pero podría haber algunas dificultades al momento de implementarlo.	d. Moral: Se refiere siempre con respeto tanto al cliente como con el gerente del proyecto.	
e. Factores humanos: Al estar		

realizando sola el proyecto no presenta ningún problema.		
f. Especificaciones: Los requisitos son claros y precisos.		

Probabilidad del riesgo e impacto

1. Riesgos	Probabilidad de riesgo	Probabilidad de impacto
Compleitud en requerimientos	Baja	Media
Claridad en requerimientos	Baja	Baja
Precedentes en requerimientos	Media	Media
Escala en requerimientos	Bajo	Alto
Formalidad en proceso de desarrollo	Medio	Medio
Idoneidad en Proceso de desarrollo	Medio	Alto
Familiaridad en el proceso de desarrollo	Medio	Medio
Instalaciones en recursos	Alto	Alto

2. Riesgos	Probabilidad de riesgo	Probabilidad de impacto
Dificultad en diseño	Alto	Alto
Confiabilidad en sistema de desarrollo	Bajo	Alto
Restricciones en contrato	Bajo	Medio

3. Riesgos	Probabilidad de riesgo	Probabilidad de impacto
Factibilidad en codificación y pruebas	Medio	Bajo
Codif/implementación en codificación y pruebas	Medio	Alto
Experiencia en adm en administración de proceso	Medio	Medio
Interfaz proyecto en adm de proceso	Bajo	Medio
Cliente en interfaz del proyecto	Bajo	Bajo

4. Riesgos	Probabilidad de riesgo	Probabilidad de impacto
Sistema en integración y pruebas	Media	Alto
Adm. de configuración en Métodos de administración	Baja	Baja
5. Riesgos	Probabilidad de riesgo	Probabilidad de impacto
Mantenibilidad en especialidades de ing	Bajo	Alto
Seguridad de Acceso en especialidades de ing	Media	Alto

Control

R

Alto			Instalaciones en recursos
			Dificultad en diseño
Medio		Codif/implementación en codificación y pruebas	Idoneidad en de Proceso desarrollo
			Sistema en y integración pruebas
			Seguridad de Acceso en especialidades de ing
Bajo			
	Bajo	Medio	Alto

Impacto

Explicación del control

Altos-Altos

Instalaciones en recursos(R1): Podemos trabajar con nuestro propio equipo, de esta manera resultaría más cómodo para nosotros y no correremos el riesgo de perder información. Ya que al trabajar con las instalaciones de la empresa (escuela) nos arriesgamos a que alguien más borre por accidente nuestros avances.

Dificultad en diseño(R2): Antes de colocar información en las bases de datos podemos crear únicamente las tablas y conectarlas para asegurarnos que la conexión fue exitosa. Al estar integrando el back con el front también podemos hacer pequeñas pruebas para asegurarnos de que el software realmente está realizando correctamente sus funciones.

Idoneidad en Proceso de desarrollo(R3): Podríamos tomar algunos cursos sobre el lenguaje de programación que vayamos a usar, así como asesorías para saber

más sobre los procesos de desarrollo. De igual manera para facilitar podríamos hacer algunos diagramas.

Sistema en integración y pruebas(R4): Para asegurarnos que el sistema no tenga tantas complicaciones dentro del código podemos leer la documentación propia del lenguaje para saber de qué manera es correcto hacerlo. También podríamos intentar hacer un código limpio, para que sea fácil de entender para cualquier programador.

Seguridad de Acceso en especialidades de ing(R5): Tendríamos que ver la manera de denegar accesos, y que el usuario genere una contraseña segura, es decir, debemos de poner ciertas reglas a la hora de que el usuario cree su cuenta para acceder al punto de vista.

Altos-Medios

Codif/implementación en codificación y pruebas(R6): Debemos de estar conscientes de la lógica de programación para seguir correctamente esto y que no nos genere tantos inconvenientes a la hora de correr el código. Y en cuanto a las pruebas vamos a documentar cada una, en especial cuando las pruebas salgan negativas, ya que le daremos solución a la misma y la documentamos.

Estrategias de Riesgos

Estrategia	Riesgo	Influencia
Aceptación del riesgo	r2, r4, r5	
Evitar el riesgo	r3	Media
Protección contra el riesgo	r5	Alta
Reducción del Riesgo	r1, r6	Media/Alta
Investigación del Riesgo	r2, r3	Alta
Reservas del riesgo	r4	
Transferencia del riesgo	r5	Media

Estrategias de Riesgos (explicación)

Aceptación del riesgo:

- **Dificultad en diseño(R2):** El diseño es una parte fundamental del software por lo que aceptaremos el riesgo, sin embargo, estaremos realizando pruebas constantes para evitar que a futuro cause problemas con la integración del hardware.
- **Sistemas en integración y pruebas(R4):** Integrar el software con el hardware sin duda es una característica esencial del sistema, por lo que la aceptaremos.
- **Seguridad de Acceso en especialidades de ing(R5):** La seguridad al ser lo más importante dentro de nuestro punto de venta la aceptaremos y haremos pruebas para garantizar la seguridad del usuario y sus clientes.

Evitar el riesgo:

- **Idoneidad en el proceso de desarrollo(R3):** Para evitar este riesgo capacitamos al personal con las habilidades básicas que se van a requerir para implementar el sistema. La influencia al evitar este riesgo sería alta, ya que se tendrían menos fallos optimizando el proceso de desarrollo.

Protección contra el riesgo:

- **Seguridad de Acceso en especialidades de ing(R5):** Usaremos un sistema de rastreo para que nos notifique cada que haya alguna vulnerabilidad en el sistema, conforme vayan saliendo más vulnerabilidades se buscará solucionarlas hasta que no quede ninguna. La influencia de lo anterior es alta, ya que sería más fácil saber qué tan seguro es el sistema.

Reducción del riesgo:

- **Instalaciones en Recursos(R1):** Para reducir el presente riesgo usaremos nuestro propio equipo de trabajo, de igual manera instalaremos lo necesario para poder implementar el sistema sin ningún problema. Disminuyendo de esta manera el no poder trabajar por falta de equipo.
- **Codif/implementación en codificación y pruebas(R6):** Para reducir los errores dentro de este riesgo haremos pruebas de caja negra, identificando de esta manera su correcto funcionamiento del software en general.

Investigación del riesgo:

- **Dificultad en diseño(R2):** Se realizará una recopilación sobre las mejores maneras para trabajar con el backend y el frontend, así como los distintos hardware que se vayan a utilizar con el software y la compatibilidad de estos con el lenguaje de programación que se vaya a utilizar.
- **Idoneidad en el proceso de desarrollo(R3):** Se hará una investigación exhaustiva para determinar qué ciclos de vida es mejor usar, así como los lenguajes de programación que son mejor utilizar para aplicaciones y que tipo

de bases de datos. De igual manera, se buscarán herramientas que faciliten la creación de los distintos diagramas que tengamos que hacer.

Reservas del riesgo:

- **Sistemas en integración y pruebas(R4):** Para evitar que los costos por pruebas se excedan, las pruebas se realizan en un periodo de tiempo determinado y limitado.

Transferir el riesgo:

- **Seguridad de Acceso en especialidades de ing(R5):** Al ser la parte más importante de nuestro sistema es de vital importancia que el acceso sea seguro y confiable. Por lo que se subcontrata a alguien que esté especializado en ese tema, únicamente para la parte de la seguridad de login.

Pruebas

Alcance de pruebas

¿Qué se probará? Principalmente se va a comprobar la integración exitosa de las bases de datos con la interfaz, así como el sistema que se usará para que el cliente pueda facturar. Así mismo, se va a revisar que el backend funcione correctamente con el frontend. Por último se probará que el software se integre bien con el hardware que se vaya a utilizar, especialmente con el escáner que va a requerir el sistema.

Niveles de prueba

- **Pruebas unitarias:** Se harán pruebas para asegurarnos de que la información que estará en los inventarios es correcta y también para asegurarnos de que la información se está almacenando de manera adecuada.
- **Pruebas de integración:** Para cerciorarnos de que la base de datos se está conectando correctamente a la interfaz realizaremos algunas consultas. De igual manera, se verificará que las funcionalidades respondan como se tiene previsto.
- **Pruebas de sistema:** Aquí vamos a verificar que la aplicación funcione correctamente, es decir, que se puedan cargar de manera correcta el inventario, puedan acceder a los precios, y se puedan cargar las ventas adecuadamente.
- **Pruebas de aceptación:** Aquí deberemos de verificar que el software esté cumpliendo con todos los requisitos que nos solicitó el cliente.

Tipos de pruebas

- **Pruebas funcionales:** Después de implementar cada función haremos pruebas para verificar su correcto funcionamiento. En caso de que una función no esté respondiendo como se espera buscaremos una solución inmediata para esto y posteriormente documentaremos tanto el fracaso, como la solución que le dimos.
- **Pruebas de rendimiento:** Veremos el tiempo que tarda en realizar una consulta, o en cargar las ventas. En caso de notar que está tardando mucho en responder buscaremos la manera de optimizar la aplicación.

Historial de versiones

Fecha	Versión	Descripción	Autores
27-10-24	1.0	Requerimientos del software	Romero Cervantes Fátima Daniela
29-10-24	1.0	Procesos de SQA	Romero Cervantes Fátima Daniela
31-10-24	1.0	Riesgos (Paso 1)	Romero Cervantes Fátima Daniela
02-11-24	1.1	Probabilidad de riesgos	Romero Cervantes Fátima Daniela
05-11-24	1.2	Control de riesgos	Romero Cervantes Fátima Daniela
05-11-24	1.3	Identificación de estrategias para los riesgos	Romero Cervantes Fátima Daniela
09-11-24	1.4	Explicación del control de riesgos	Romero Cervantes Fátima Daniela
19-11-24	1.0	Pruebas	Romero Cervantes Fátima Daniela

Glosario

SQA: Aseguramiento de la calidad del software. Es un software que asegura que todos los métodos, procesos, actividades y elementos de trabajo de ingeniería de software puedan ser monitoreados y cumplan con los estándares previamente definidos.

Backend: Es aquel que ocupa tareas como almacenar y recuperar datos, procesar formularios, autenticar usuarios y gestionar la seguridad del sitio.

Frontend: Es la parte visible de un sitio web o aplicación con la que los usuarios pueden interactuar directamente.