

Merkit

“Análisis y diseño”

Tamara Badilla, Álvaro Valverde
Y Sebastián Pacheco

16 de Agosto del 2020

<i>Contenidos</i>	<i>Merkit</i>
→ Sobre el proyecto	2
→ UML	3
→ Caso de uso	4
→ Diagrama conceptual	6
→ Diagrama de colaboración	6

Sobre el proyecto

“*Merkit*” es un proyecto que gira en torno a los mini-mercados. El propósito de este es facilitar y hacer más eficiente el trabajo de los administradores al momento de realizar una transacción o venta, esto se logra mediante la posibilidad de almacenar los precios de cada uno de sus productos, manteniendo un stock que permite acceder a ellos en cualquier momento de una forma rápida y sencilla.

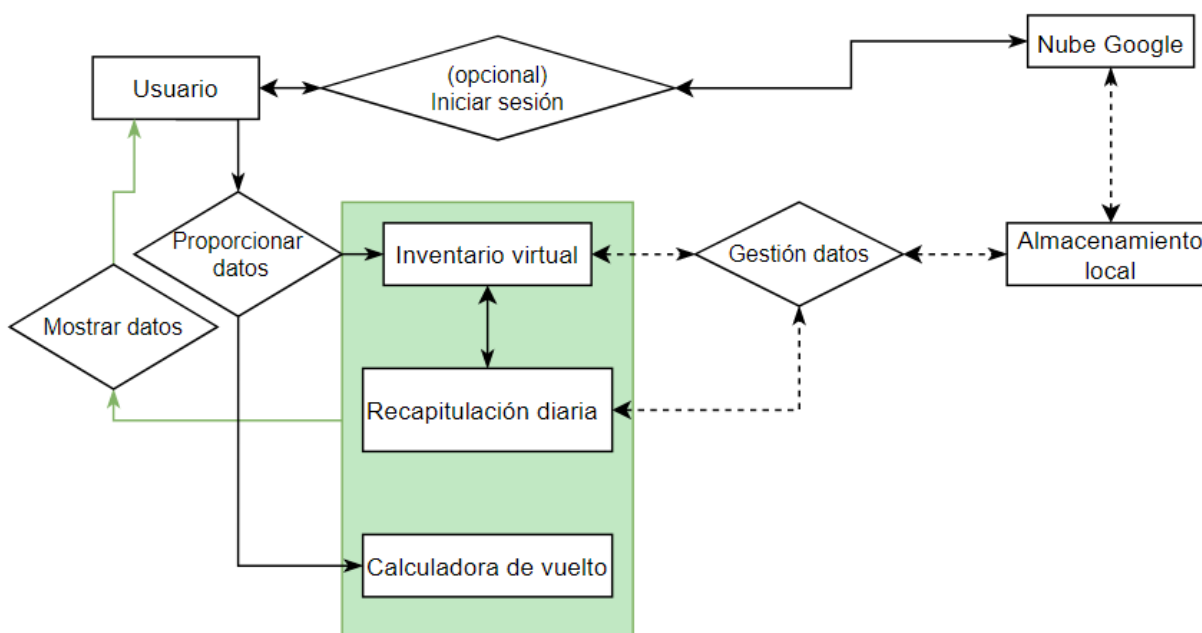
Otras funciones que posee el sistema, es el resumen de ventas diarias, una calculadora que permite al usuario calcular el vuelto que debe de entregar a su cliente, y tener un respaldo de los datos del mercado en la nube. Con toda esta información siendo almacenada y gestionada por “*Merkit*”, se logra obtener una mejora en el rendimiento, la velocidad, y la calidad de atención a los clientes del mini-mercado.

La aplicación “*Merkit*” estará disponible de forma gratuita y en un principio sólo para el sistema operativo Windows 10, con la posibilidad de expandirse en un futuro a la gestión de datos a través del celular y/o tablet.

Nos referiremos a la persona que interactúa con el Software como Administrador, Cajero o Usuario. Para la persona que compra, nos referiremos como Comprador o Cliente.

UML:

Para representar de una manera visual el sistema, desde el punto de vista de las interacciones de un usuario, se usará el formato UML, en el cual se mostrará qué es lo que el actor puede realizar.



(Fig.01. "Descripción gráfica del sistema")

Como se puede apreciar en la imagen anterior (Fig.01.), el usuario proporciona los datos tales como productos, stock, descripción de éstos y precio, los cuales la aplicación registra directamente en el almacenamiento local. Si es posible, luego de realizar lo anterior, se almacenan en la nube de Google (esto sólo se puede llevar a cabo si el usuario inició sesión con su cuenta de Google previamente).

Para activar la recapitulación diaria, el usuario debe indicar cuando inicia el día, y luego indicar cuándo se acaba, de esta manera la app tendrá información suficiente para entregar el resumen de ventas. La calculadora de vuelto no necesita almacenar ningún dato, por lo que no es necesario que se conecte al almacenamiento local, ni a la nube.

Casos de uso

Se evaluarán dos casos de uso para visualizarlos en diagramas de secuencia y contratos, los cuales se nombran a continuación:

Caso de uso: Compra con arrepentimiento

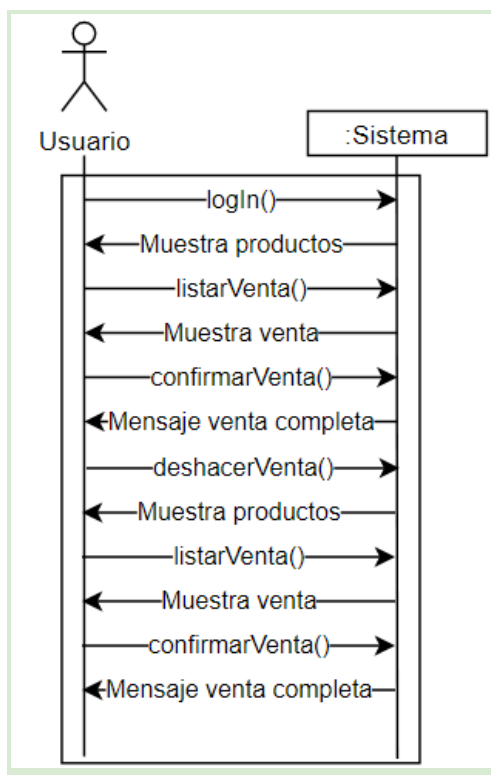
Actores: Cliente, Cajero (administrador minimarket)

Propósito: Capturar una venta y agregarla al resumen diario

Condición de Entrada: 'Merkit' está disponible en un computador

Resumen: Un cliente ingresa al minimercado en busca de unas papitas, pero luego de que el administrador completa la compra, el cliente se arrepiente, por lo que el cajero debe deshacer la compra e ingresar el nuevo pedido.

Para el este caso de uso se propone el siguiente diagrama de secuencia:



(Fig.02. "Diagrama de secuencia")

Sobre el cuál se proponen los siguientes contratos:

Nombre:	Login(<String> Usuario, <String> Contraseña)
Responsabilidades:	El sistema se encargará de crear una sesión para el usuario
Pre-condiciones:	Ninguna.
Post-condiciones:	El usuario tiene una sesión válida.

Nombre:	Listar Venta
Responsabilidades:	El sistema dispondrá al usuario de una barra de búsqueda, una zona de selección y una de listado, que le permitirá buscar, seleccionar y añadir Productos a la Lista de Compras.
Pre-condiciones:	Usuario validado, y existe un Archivo de Productos.
Post-condiciones:	El usuario ha creado una Lista de Venta válida.

Nombre:	Confirmar Venta(<Lista> Venta)
Responsabilidades:	El sistema responderá a la solicitud del usuario de confirmar la venta a través de un botón. Registrará la Venta, reajustará el inventario y registrará la transacción monetaria (dinero recibido, vuelto, etc.).
Pre-condiciones:	Usuario validado, y la Venta es válida.
Post-condiciones:	La Venta ha sido registrada con éxito.

Nombre:	Deshacer Venta(<Archivo> Ventas)
Responsabilidades:	El sistema permitirá al usuario revertir la Venta más reciente, desde el archivo de Ventas.
Pre-condiciones:	Existen Ventas en el archivo, o recientemente se ha realizado alguna Venta.
Post-condiciones:	Se rellena la Ventana de Venta tal cual estaría antes de confirmar la Venta, permitiendo la modificación.

Caso de uso: Llegada de cargamento

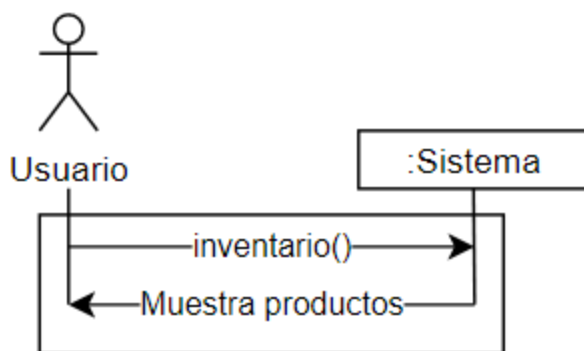
Actores: Cajero y trabajador

Propósito: Ingresar productos al inventario

Condición de Entrada: ‘Merkit’ está disponible en un computador y el usuario tiene la sesión iniciada

Resumen: Llega un cargamento de algún producto al mini mercado, por lo que el usuario ahora debe agregarlo al inventario del sistema.

Para el este caso de uso se propone el siguiente diagrama de secuencia:



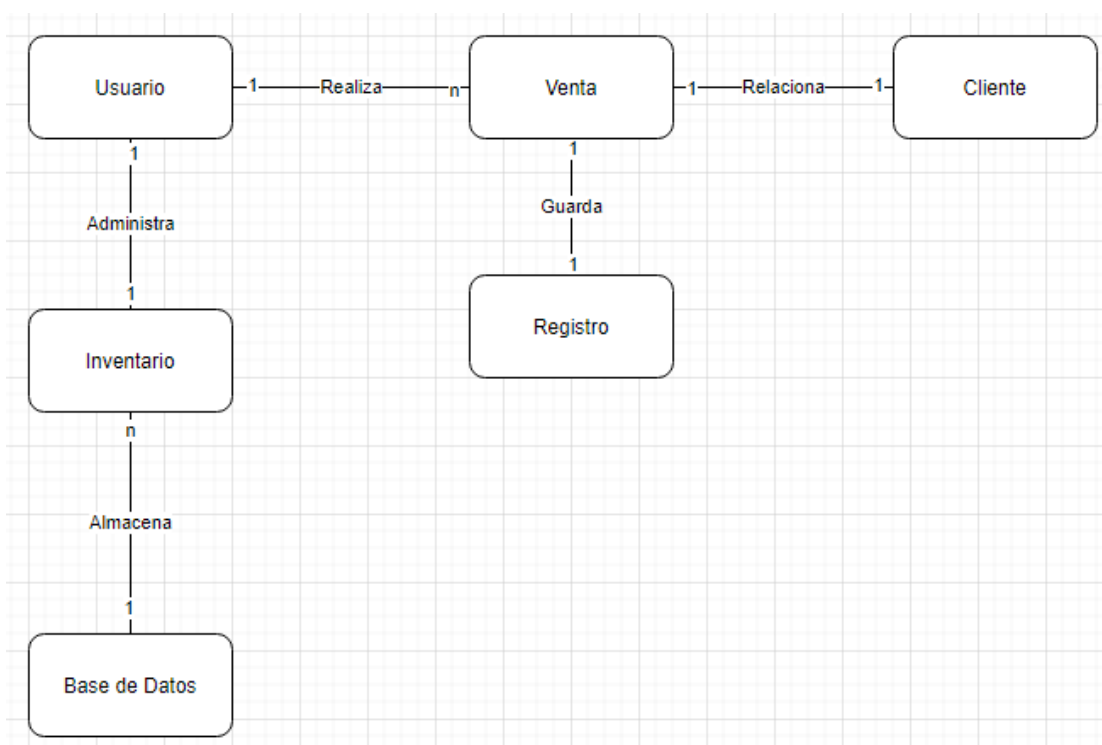
(Fig.03. “Diagrama de secuencia 2”)

Sobre el cuál a la vez también se propone el siguiente contrato:

Nombre:	Inventario(<Archivo> Productos)
Responsabilidades:	El sistema dispondrá al usuario su Inventario de Productos, y le permitirá modificar o agregar nuevos productos.
Pre-condiciones:	Usuario validado, y existe un Archivo de Productos.
Post-condiciones:	La interfaz gráfica despliega una ventana de inventario.

Diagrama conceptual

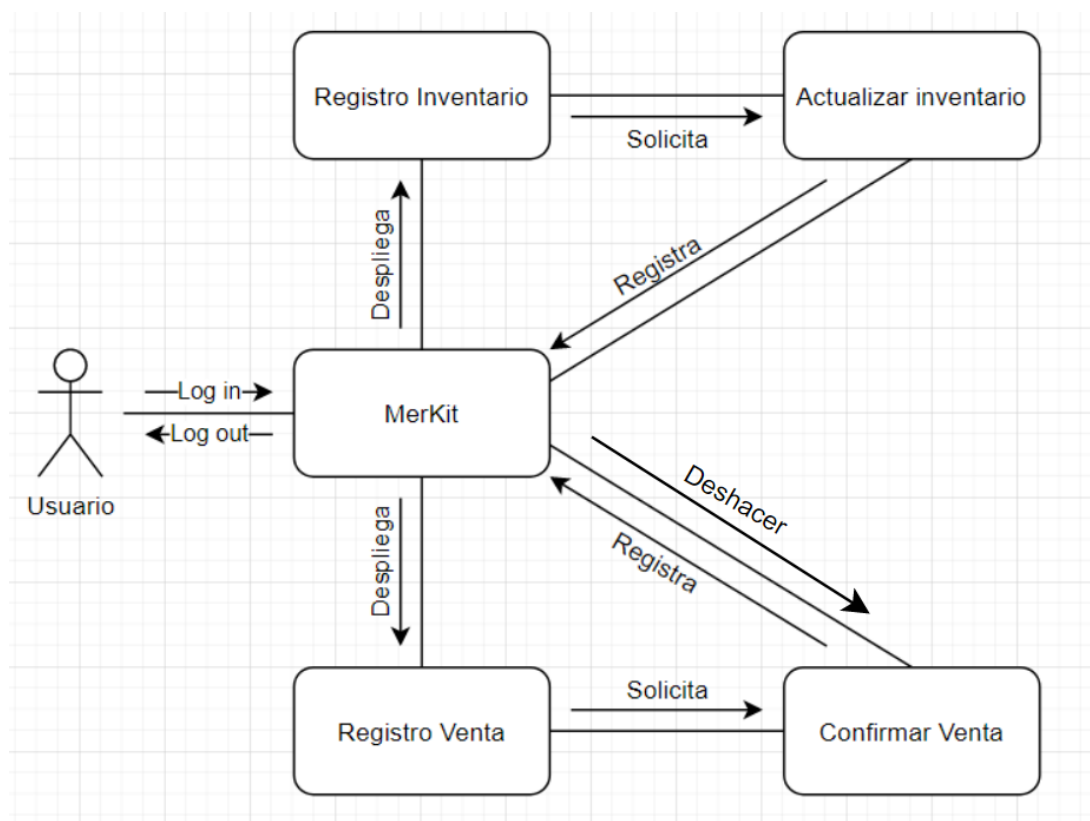
Como una manera de tener más claro el concepto del sistema a realizar, se crea un diagrama conceptual (Fig. 03) que muestra las relaciones de cada elemento observado dentro de la aplicación.



(Fig.04. "Diagrama Conceptual")

Diagrama de colaboración

Para representar las interacciones de los roles, se realiza el siguiente diagrama de colaboración:



(Fig.05. "Diagrama de Colaboración")