

Universidad Autónoma De Ciudad Juárez
Instituto de Ingeniería y tecnología
Departamento de tecnologías

Caso de estudio: Mandado a domicilio

Hernández Rivadeneyra Ulises 159965
Ruiz Sánchez Brian Baltazar 163930



Desarrollo de software

Proyecto

Docente: Pérez Martínez Jorge Antonio

Abril 2018

Introducción

El documento muestra el análisis de un proyecto propuesto para crear una aplicación de compras de mandado, las descripciones del mismo, características, tablas de requisitos de la aplicación, las diferentes problemáticas que este puede resolver así como sus soluciones, casos de uso del software, y anexos del mismo.

Tabla de contenido

Introducción

Justificacion.....	1
Descripcion.....	2
Requisitos del software.....	3
UdeD.....	4
Representación grafica del UdeD.....	6
Casos de uso.....	7

Justificación

La tecnología empieza a satisfacer las necesidades que nos quitan tiempo en la vida diaria, quizá no muy difíciles de hacer, pero por comodidad o en ciertos casos por falta de tiempo y un itinerario muy rígido hay cosas que pasan a segundo término y se vuelve complicado realizar. Ese es el mercado al que queremos llegar, quitarle una carga o ayudar a la gente en algunos casos. Como ya se mencionó anteriormente, existen diferentes razones, desde la comodidad, hasta la necesidad. Existen zonas de la ciudad que están alejados de supermercados. También se piensa en la gente que no tiene un medio de transporte para poder regresar con las compras a su hogar y con los gastos ya sea de un taxi, o ruta pagar él envió al domicilio. El motivo del guardado de las compras está pensado porque muchas de las compras en el hogar son las mismas semanalmente, la visualización del stock está enfocado al cambio de algún producto que no exista en ese momento, para que la lista no quede incompleta, y a su vez para poder evitarlo, se quiere implementar la reserva de productos. Los niveles de supermercados de baratos a más caros, para poder cumplir con cualquier presupuesto puesto que no todos tienen los mismo ingresos y el de calidad por si se requieren ciertos productos para ocasiones especiales.

Descripción

La aplicación, permite comprar productos de supermercados (mandado), esta tiene niveles de precios-calidad de supermercados y de productos de los mismos. En la aplicación el usuario podrá ingresar por medio de su cuenta personal, guardar compras anteriores para poder replicarla las veces que desee, visualizar el stock de productos para poder hacer cambios, elegir si se desea pasar a recoger los productos o bien, que sean enviados a su casa y poder apartar productos para fechas específicas. Podrá reservar su compra para evitar que el día que se realice, algún producto este agotado. La aplicación contiene principios de usabilidad así como personalización de perfil, pago a través de tarjeta o efectivo. La aplicación será gratuita y los usuarios gastarán solo en el precio de su compra, y el envío si así lo desean.

Tabla de requisitos

- 1- Se muestra los supermercados disponibles para la compra de productos en base a la ubicación del usuario en forma de croquis/mapa.
- 2- Se debe ver los productos disponibles del supermercado seleccionado con su información (precio, marca, cantidad).
- 3- Tiene una lista de los supermercados ordenados por precios de menor a mayor en base a la canasta básica.
- 4- Se muestra el perfil de cada usuario con su respectiva interface de inicio de sesión, la cual es rápida e intuitiva, así como la edición de los datos del mismo (Nombre, Tarjetas, Dirección etc.).
- 5- Se permite ver una lista de los productos ordenados por departamentos simulando la tienda física.
- 6- Tiene protección de la cuenta del usuario con contraseña y pregunta de seguridad.
- 7- Se muestra el historial de las compras realizadas con los datos de las mismas (fecha, costos, productos, etc.)
- 8- Cuando se elija envió del mandado a casa, se debe ver una interfaz para seleccionar, fecha, hora preferida de entrega, el costo y los datos de la compra.
- 9- En compras fallidas, se muestra una pantalla de error en la cual se puede decidir si volver a intentar la compra, o salir y volver a la pantalla del supermercado.
- 10- Se muestra la pantalla para ver el transcurso del envió del mandado a la dirección destino en caso de que se seleccione dicha opción.
- 11- Se puede elegir entre pagar con efectivo o con tarjeta.
- 12- Se puede ver la interfaz al momento de apartar la compra, donde se ve los artículos seleccionados.

Universo del discurso

En la aplicación de mandado se necesita mantener información y estarse actualizando cada día con nuevos productos en venta para ello se utiliza una base de datos, otra base de datos para almacenar la información de las cuentas de los clientes(nombre, tarjetas, compras etc.). Utiliza el sistema gestor de información del mapa que se utiliza para la localización de los supermercados. El sistema se puede instalar en teléfonos Android y iOS, para lo cual se utilizan diversos programas para la edición de diseño, gráficos, optimización etc. El sistema cuenta con conexión a los bancos para los pagos de tarjetas y por ultimo cuenta con conexión a la base de datos de las empresas repartidoras par el envío de las compras.

Características de los usuarios: Los usuarios se componen en clientes, administradores de la base de datos, administradores de la aplicación y administradores de la base de datos de cada supermercado.

Limitaciones de la aplicación: La aplicación puede no ser utilizada en lugares a donde no llega señal de internet, esto genera solo un pequeño problema, ya que dichos lugares probablemente sean muy pequeños, esto de igual manera afecta puesto que las cadenas no pensarían en invertir en la aplicación, puesto que sus clientes llegan relativamente muy rápido a comprar. Así mismo existe la posibilidad de que en lugares en donde solo exista 1 cadena predominante, la aplicación no tenga éxito ya que el mercado, no necesitaría de esta opción debido a su éxito. Por ultimo tenemos el caso menor en el que la gente decida ir a comprar el mandado por ellos mismos, esto se debería a diversos factores como tener tiempo, ser adultos, tener el supermercado cerca, o no confiar en la app.

Limitaciones ajenas a la aplicación: En este caso, la única limitación es el no tener un trato con los supermercados, es mínima la posibilidad, ya que al estar en competencia por vender más, los supermercados siempre quieren estar un paso adelante. Aun así existe la posibilidad de que los supermercados no quieran trabajar con la aplicación.

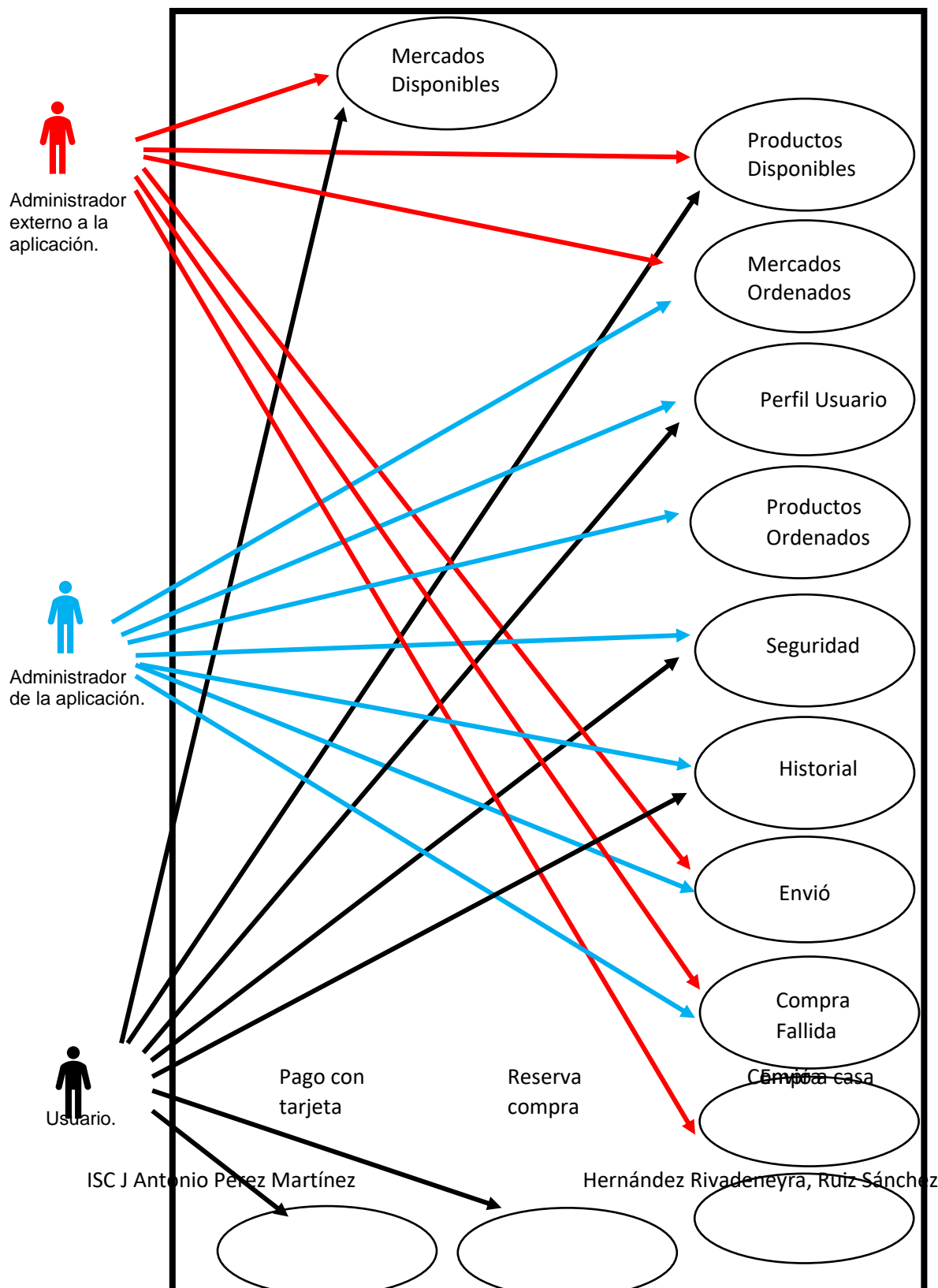
Requisitos a futuro: La aplicación necesita constantes cambios, en cuanto a seguridad, sistema de pago. Mantenimiento muy constante en la base de dato para loes errores. Posibles cambios en el método de envío y por ultimo expandir la aplicación a la mayoría de supermercados y lugares posibles.

Requisitos específicos

Requisitos no funcionales: Manuales de usos rápidos e intuitivos, interface con los principios de usabilidad, diseño de la aplicación básica pero presentable,

Requisitos funcionales: Acceso a la aplicación las 24 horas, alta velocidad en el enlace de datos(inventarios), alta velocidad en el proceso de pago, controles estrictos en la administración de las bases de datos en la aplicación, Seguridad para posibles intentos de tirar la app, cifrado de datos de los usuarios.

Representación gráfica del universo del discurso.

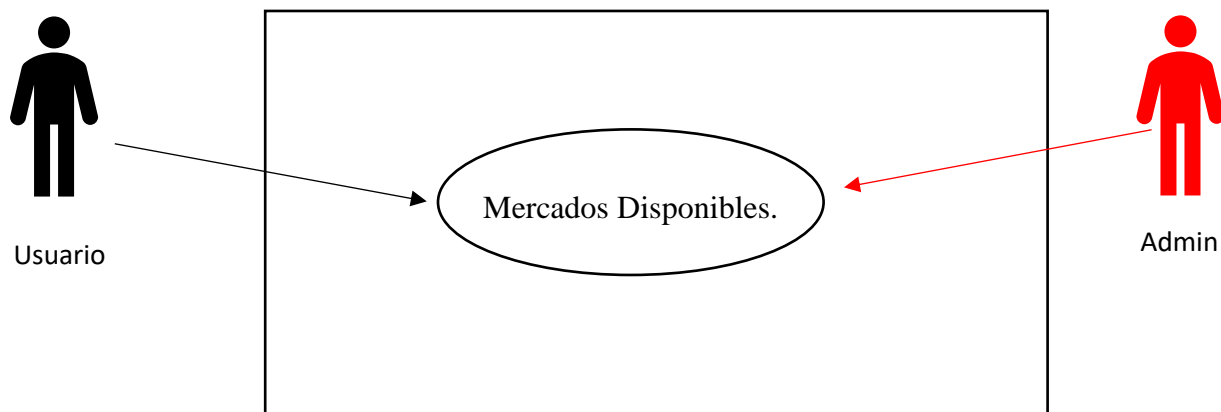


Usuario es el que usa la app.

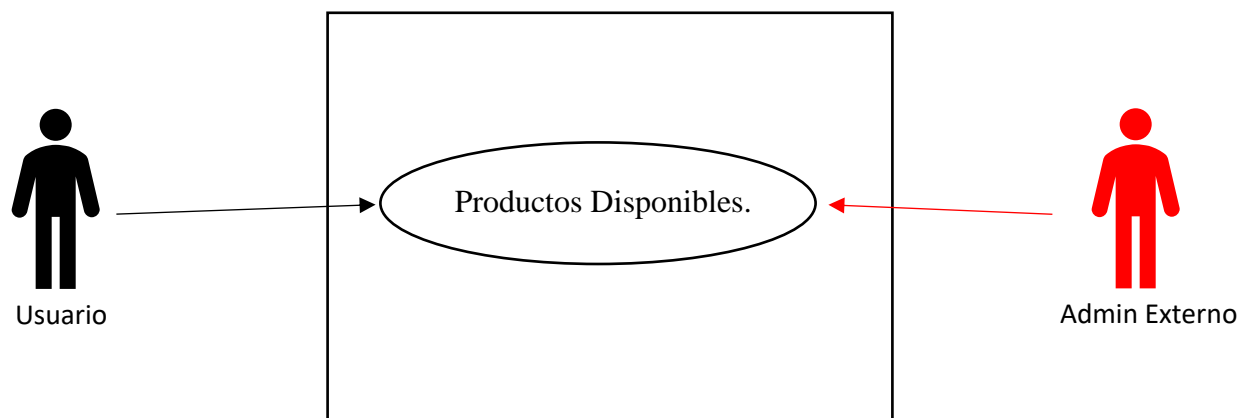
Administrador externo es aquel administrador que no trabaja para la aplicación (administrador de los supermercados, bancos etc.)

Administrador de la aplicación es aquel que trabaja para la app manejando las cuentas, seguridad, diseño etc.

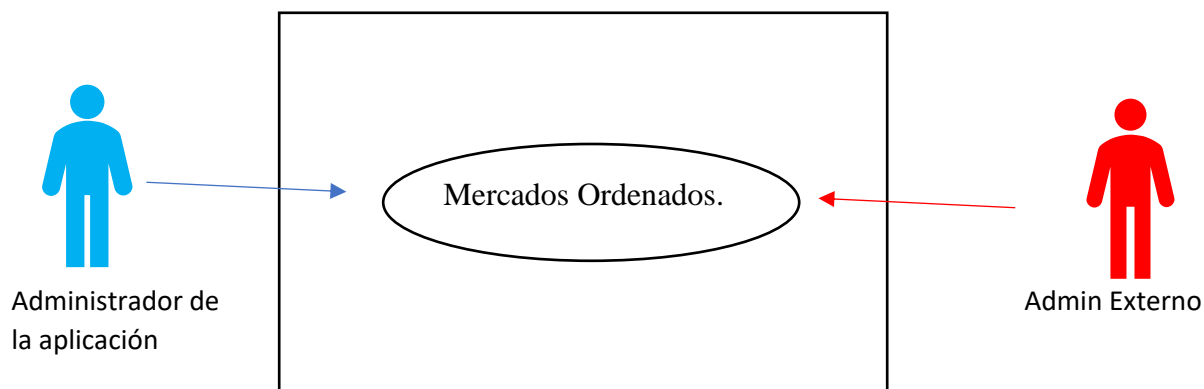
Casos de uso



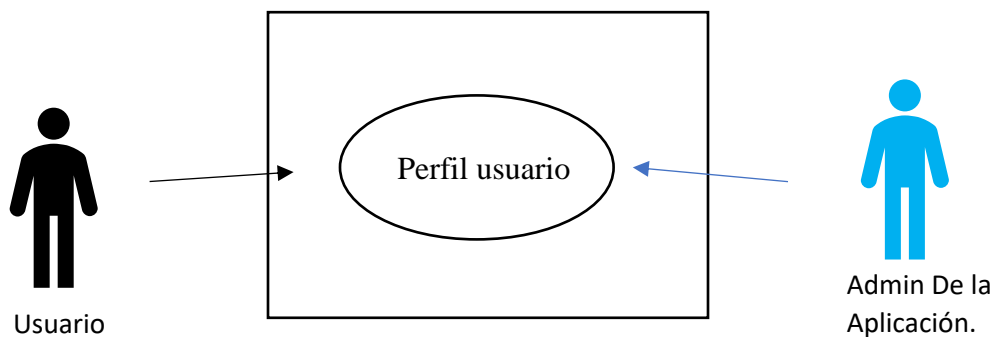
Nombre.	Mercados Disponibles.
Actores.	Usuario, Admin externo.
Pre-condición.	Tener en la base de datos guardados los supermercados afiliados a la app, tener guardadas las ubicaciones de los mismos.
Postcondición.	Se obtienen los supermercados más cercanos al usuario disponibles para la compra mediante la app.
Flujo Normal	<ol style="list-style-type: none"> 1- El sistema se conecta con la base de datos del sistema gestor del mapa. 2- Se transfiere la información a la base de datos de la aplicación como un campo temporal. 3- La aplicación muestra en el mapa la información obtenida, siendo esta la ubicación de los supermercado mas cercanos.
Flujo alterno	<ol style="list-style-type: none"> 1- Si no está activada la ubicación en el celular, la aplicación mostrara un mensaje pidiendo que esta se active. 2- Si el sistema gestor del mapa no está disponible, se ve un mensaje en pantalla pidiendo al usuario que intente unos minutos después. 3- Si no existen supermercados en su localización, se muestra un mensaje en la aplicación para avisar al usuario.



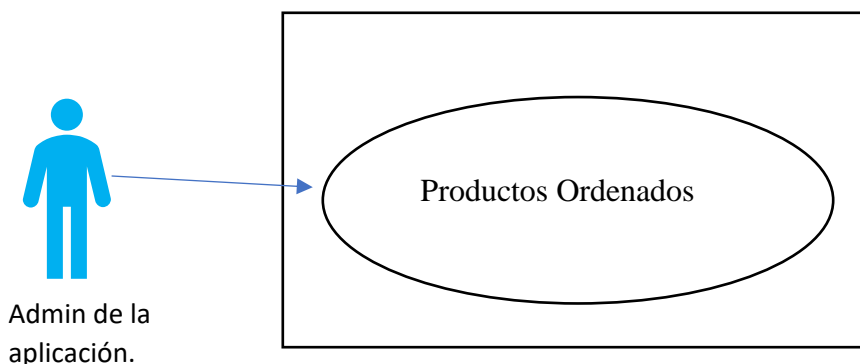
Nombre.	Productos Disponibles.
Actores.	Usuario, Administrador externo.
Pre-condición.	Tener acceso a los inventarios actuales del supermercado seleccionado, guardar los inventarios en la base de datos de la app.
Postcondición.	Se obtienen los productos disponibles, se muestran en pantalla con su información (precio, marca , cantidad)
Flujo Normal	1- El sistema accede a la base de datos de los supermercados afiliados. 2- Se traslada la información obtenida a un campo temporal en la base de datos de la aplicación. 3- Se muestra en la aplicación la información obtenida de los productos disponibles.
Flujo alterno	1- Si no hay productos disponibles, aparece un mensaje en la aplicación informando al usuario. 2- S hay un error en cualquier base de datos o de conexión,(ajeno al usuario), se informa al usuario con un mensaje sugiriendo el intentar de nuevo en unos minutos.



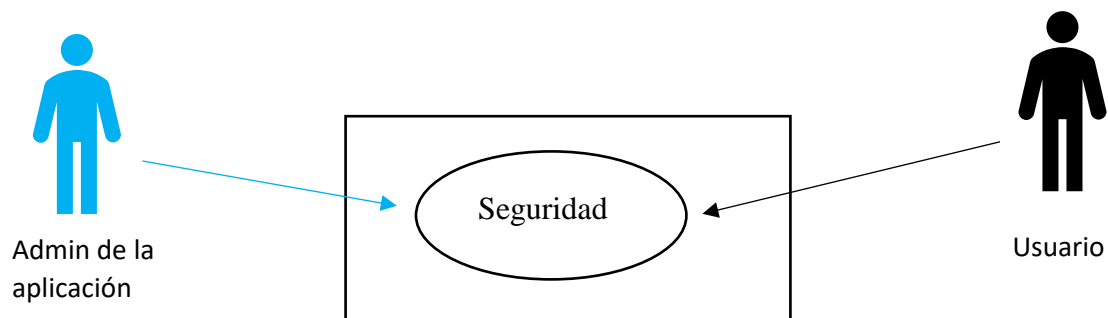
Nombre.	Mercados Ordenados.
Actores.	Administrador de la aplicación, Administrador externo.
Pre-condición.	Tener acceso a los inventarios actuales de todos los supermercados afiliados a la app para poder checar los precios de la canasta básica de cada uno. Caso de uso Mercados Disponibles.
Postcondición.	Se juntan todos los costos de cada mercado, para hacer un presupuesto aproximado de los productos, de esta manera se ordena de menor a mayor costo los supermercados.
Flujo Normal	1- Se accede a lavase de datos de toso los supermercados afiliados. 2- Se recopilan los precios de productos específicos. 3- Cada presupuesto se guarda en una columna diferente de la tabla temporal en la base de datos. 4- Se ordena mediante un algoritmo del presupuesto más barato, al más caro. 5- Se muestra en pantalla lo obtenido en el paso anterior.
Flujo alterno	1- Si no se puede obtener el presupuesto de algún supermercado, en el mensaje de la aplicación aparecerá como presupuesto no disponible. 2- Si existen fallos en las conexiones entre bases de datos, se informa al usuario con un mensaje sugiriendo el intentar de nuevo en unos minutos.



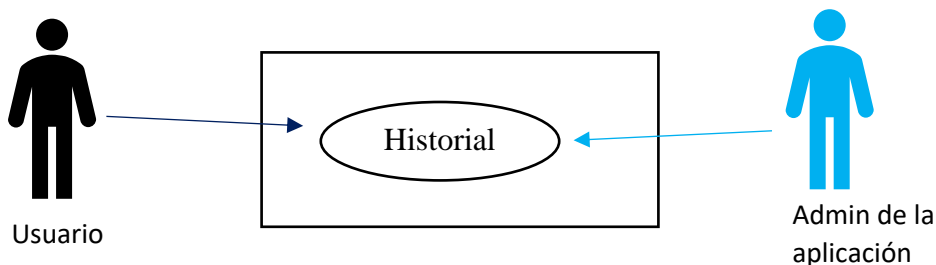
Nombre.	Perfil usuario
Actores.	Usuario, Administrador de la aplicación.
Pre-condición.	Se busca en la base de datos si existe un usuario registrado con la información que se ingresó en el inicio de sesión.
Postcondición.	De no encontrarse algún registro, se pasa al caso de uso Nuevo Usuario. Se muestra en pantalla el perfil, con sus datos (Nombre, dirección, correo) y se puede empezar a comprar.
Flujo Normal	1- El usuario inicio sesión mediante su cuenta. 2- La base de datos almacena la información del inicio de sesión. 3- La base de datos relaciona toda la información que se guarde(Tarjetas, compras, calle etc.) en el perfil, y lo relaciona con el inicio de sesión previamente guardado.
Flujo alternativo	1- Si no se realizó el inicio de sesión correcto, se avisa al usuario que algún campo o coincide. 2- Si se intenta muchas veces el inicio de sesión sin éxito, se le dice al usuario que espere unos minutos o inicie sesión mediante el caso de uso seguridad.



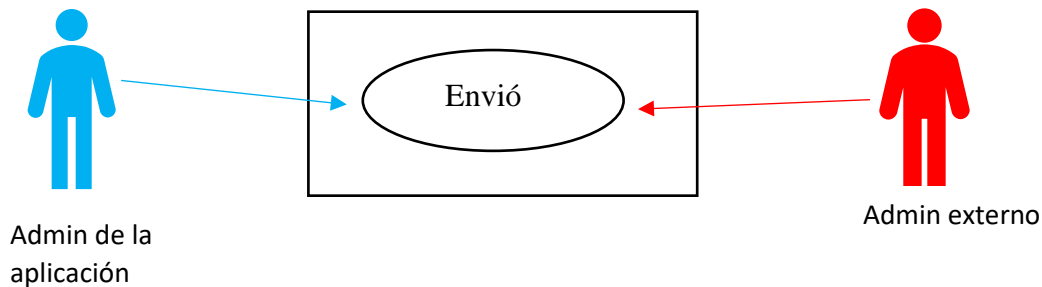
Nombre.	Productos Ordenados
Actores.	Administrador de la aplicación.
Pre-condición.	Tener acceso a los inventarios actuales del supermercado seleccionado, guardar los inventarios en la base de datos de la app.
Postcondición.	Se muestran en pantalla los productos por secciones, estas secciones son categorías (higiene, frutas, lácteos, etc.) con su información (precio, marca).
Flujo Normal	<ol style="list-style-type: none"> 1- Se recopilan todos los productos guardados en la base de datos del supermercado seleccionado. 2- Se separan por secciones dichos productos. 3- Se ordenan mediante un algoritmo por precio, cada sección pasa por el algoritmo. 4- Se muestra en pantalla los productos, ya ordenados por secciones y precio.
Flujo alternativo	<ol style="list-style-type: none"> 1- En caso de que no existan productos de alguna sección(Se acaben) no aparece la sección. 2- En caso de que solo exista un producto disponible en alguna sección, esta no pasa por el algoritmo de orden de precio. 3- Si existe algún problema con la base de datos u los algoritmos de separación y orden de precio, se le da la opción al usuario de intentar mas tarde, o ver los productos disponibles sin ordenarlos.



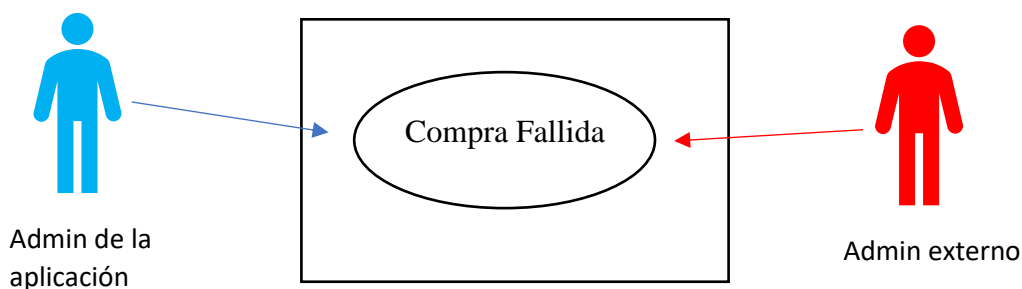
Nombre.	Seguridad
Actores.	Usuario, Admin de la aplicación.
Pre-condición.	Antes de entrar al perfil, se pide verificar la identidad con la forma de seguridad que el usuario eligió.
Postcondición.	Se muestra en pantalla un mensaje solicitando el ingreso ya se dé pregunta de seguridad, huella, contraseña etc.
Flujo Normal	<ol style="list-style-type: none"> 1- Se conecta a la base de datos para recopilar la información del perfil y saber que forma de seguridad se eligió. 2- El sistema pide al usuario que se utilice el método escogido para entrar, mostrando un mensaje en la aplicación. 3- Al conectarse con éxito, se accede al perfil del usuario y las diferentes características de la aplicación.
Flujo alterno	<ol style="list-style-type: none"> 1- Si el sistema no es capaz de recopilarla información por error(ajeno al usuario), se muestra aun mensaje para informar al usuario. 2- Si el usuario introdujo de forma incorrecta la verificación, se pide que la ingrese de nuevo. 3- En caso de no lograr tras muchos intentos la conexión correcta. Se pide al usuario que recupere su contraseña.



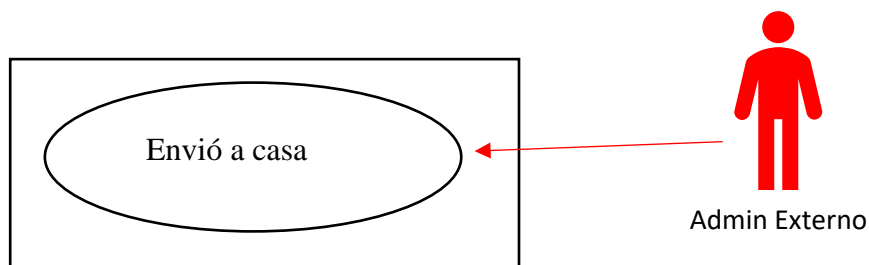
Nombre.	Historial
Actores.	Usuario, Administrador de la aplicación
Pre-condición.	La aplicación busca en la base de dato el historial de hasta 5 compras del usuario que tenga la sesión iniciada, mediante un algoritmo se copia dicha información.
Postcondición.	En pantalla aparece el historial de compra, con todas sus características (fecha de compra, productos, precio, etc.).
Flujo Normal	1- Se conecta a la base de datos de la aplicación con la información del usuario. 2- Se accede a la tabla Historial la cual contiene las compras. 3- Se muestra mediante un mensaje en la aplicación lo recopilado en el paso anterior.
Flujo alterno	1- Si la tabla historial esta vacía, se muestra un mensaje el cual dice que no se han realizado compras. 2- Si no se puede conectar a la información del usuario, aparece un mensaje en la aplicación informándole del error al usuario.



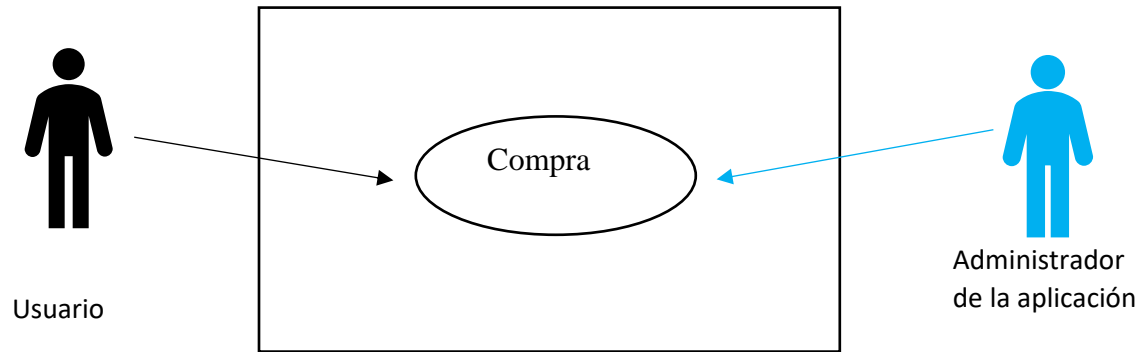
Nombre.	Envío
Actores.	Administrador de la aplicación, Administrador externo.
Pre-condición.	Guarda la información que el usuario elige para él envío, y los pasa a la base de datos del mercado para su envío.
Postcondición.	Muestra en pantalla toda la información que el usuario eligió.
Flujo Normal	<ol style="list-style-type: none"> 1- Se recolecta la información relevante para el envío del usuario(dirección, forma de pago, productos, etc.). 2- Se enlaza la información con la base de datos de la empresa afiliada para los envíos. 3- La empresa reenvía una tabla actualizada hacia la base de datos de la aplicación, pero ahora con la información del envío. 4- Se muestra en pantalla un mensaje con la información actualizada.
Flujo alterno	<ol style="list-style-type: none"> 1- En caso de que los envíos no estén disponibles, se muestra un mensaje al usuario de que no es posible realizar esa acción de momento. 2- En caso de que el envío no esté disponible para su dirección(por cuestiones dela empresa de envíos), se le dice al usuario que no se realizan envíos a su dirección.



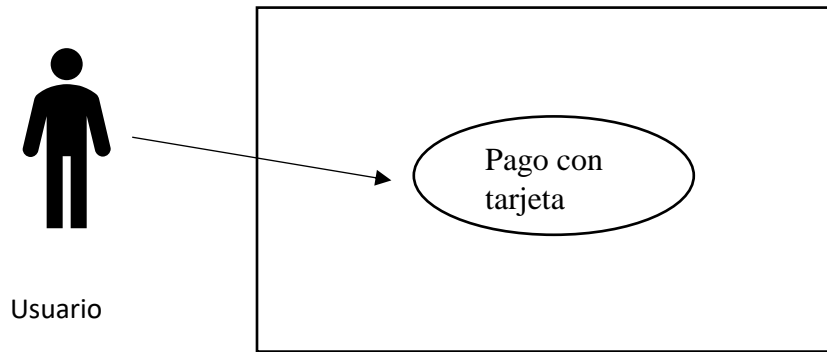
Nombre.	Compra Fallida
Actores.	Administrador de la aplicación, Administrador externo.
Pre-condición.	Guarda la información que el usuario elige para el envío, y los pasa a la base de datos del mercado para su envío.
Postcondición.	Muestra en pantalla toda la información que el usuario eligió, al no estar disponible, la compra falla y aparece en pantalla un menú para cambiar fecha u hora, datos de la compra etc.
Flujo Normal	1- Se conecta a la base de datos del supermercado elegido. 2- Se genera una tabla con la información que el cliente eligió. 3- Se compara la tabla creada con la información de la base de datos del supermercado. 4- Como no está disponible alguna elección del cliente, la compra se cancela.
Flujo alternativo	1- Si los productos no están disponibles, la compra se procesa y el caso de uso compra fallida, no entraría.



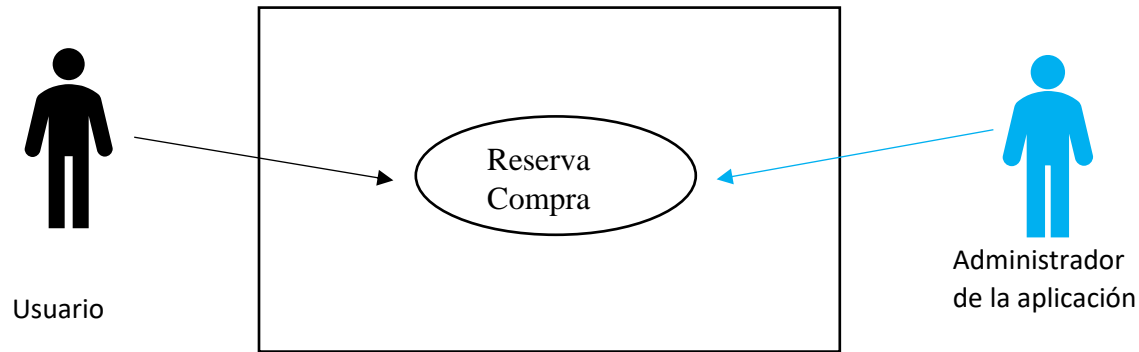
Nombre.	Envió a casa
Actores.	Administrador externo.
Pre-condición.	Caso de uso envió.
Postcondición.	Se muestra en pantalla el proceso de envió.
Flujo Normal	<ol style="list-style-type: none"> 1- Se utiliza el caso de uso envió. 2- Con la información recolectada del caso de uso envio, se genera una interfaz desde la aplicación la cual contiene los tiempos aproximados del envió. 3- Se conecta con el sistema del mapa afiliado a la aplicación. 4- Se crea un mapa que se muestra al usuario, utilizando el punto 2 y 3 para que el usuario vea gráficamente el tiempo aproximado de espera.
Flujo alterno	<ol style="list-style-type: none"> 1- Si el sistema de mapa no esta disponible, se muestra únicamente el tiempo aproximado de envió. 2- Si el tiempo de envió no está disponible, aparece un mensaje diciéndole al usuario que espere unos minutos para resolver el problema. 3- Si la base de datos de la aplicación falla , aparece un mensaje al usuario diciéndole que lo vuelva a intentar en unos minutos.



Nombre.	Compra
Actores.	Usuario, Administrador de la aplicación.
Pre-condición.	Caso de uso Productos.
Postcondición.	Se muestra en pantalla los productos seleccionados
Flujo Normal	1- El sistema se conecta a la base de dato de la aplicación. 2- Se revisa que los productos seleccionados por el usuario estén disponibles. 3- Se pasa a la sección de pago para realizar la compra.
Flujo alternativo	1- Si los productos no están disponibles, se le notifica al usuario mediante un mensaje en pantalla. 2- Si la base de datos falla, se le notifica al usuario mediante un mensaje en pantalla.



Nombre.	Pago con tarjeta
Actores.	Usuario
Pre-condición.	Caso de uso compra
Postcondición.	Muestra en pantalla la selección de forma de pago.
Flujo Normal	<ol style="list-style-type: none"> 1- El sistema se conecta a la base d datos para ver la información de los bancos afiliados. 2- Se conecta a la base de datos para obtener la información de la tarjeta del usuario. 3- Se valida los datos en el check del sistema.
Flujo alternativo	<ol style="list-style-type: none"> 1- Si la tarjeta no se acepta, aparece el mensaje con error en pantalla. 2- Si el banco no autoriza el pago, aparece un mensaje en pantalla para informar al usuario.



Nombre.	Reserva Compra
Actores.	Usuario, Administrador de la aplicación.
Pre-condición.	Se guarda en la base de datos la información de la compra seleccionada por el usuario.
Postcondición.	Se muestra en pantalla los productos seleccionados en la reserva.
Flujo Normal	1- El sistema se conecta ala base de datos del mercado seleccionado. 2- Se verifica que los productos seleccionados estén disponibles. 3- Se completa la reserva guardando la información en la base de datos.
Flujo alterno	1- Si los productos no están disponibles aparece un mensaje avisando al usuario.