**Instituto Tecnológico Colonia CTC**

**Software para Frutas y Verduras Frescas S.A.**

**Autores: Bruno Ríos y Matías Erramouspe**

**Tutor: Andrés Klett**

**2023**

# Índice

[Índice 2](#_Toc135758757)

[Abstract 4](#_Toc135758758)

[Anteproyecto 5](#_Toc135758759)

[Introducción 5](#_Toc135758760)

[Presentación del cliente 6](#_Toc135758761)

[Presentación del problema 8](#_Toc135758762)

[Lista de necesidades 9](#_Toc135758763)

[Actores Involucrados 10](#_Toc135758764)

[Lista de Objetivos 11](#_Toc135758765)

[Requerimientos 12](#_Toc135758766)

[Requerimientos funcionales 12](#_Toc135758767)

[No Funcionales 14](#_Toc135758768)

[Descripción del entorno 15](#_Toc135758769)

[Alcance y limitaciones 16](#_Toc135758770)

[Alcances 16](#_Toc135758771)

[Limitaciones 16](#_Toc135758772)

[Estudio de alternativas 17](#_Toc135758773)

[Alternativa 1: 17](#_Toc135758774)

[Alternativa 2: 17](#_Toc135758775)

[Arquitectura 17](#_Toc135758776)

[Análisis de factibilidad 18](#_Toc135758777)

[Alternativa 1: 18](#_Toc135758778)

[Alternativa 2: 23](#_Toc135758779)

[Selección de alternativa 25](#_Toc135758780)

[Análisis de riesgo 26](#_Toc135758781)

[Plan de proyecto 32](#_Toc135758782)

[Definición del proceso 32](#_Toc135758783)

[Metodología 32](#_Toc135758784)

[Ciclo de vida elegido 34](#_Toc135758785)

[Etapas del ciclo de vida elegido 34](#_Toc135758786)

[Iteraciones definidas 35](#_Toc135758787)

[Integrantes y roles 37](#_Toc135758788)

[Descripción y selección de herramientas 38](#_Toc135758789)

[Plan SQA 39](#_Toc135758790)

[Plan de SCM 44](#_Toc135758791)

[Plan de capacitación 45](#_Toc135758792)

[Cronograma de trabajo y criticidad 46](#_Toc135758793)

[Contrato de trabajo 51](#_Toc135758794)

[Glosario 53](#_Toc135758795)

# Abstract

Se propone el desarrollo de un software que permita a los clientes realizar sus pedidos a través de una plataforma en línea, con una interfaz sencilla e intuitiva. El software incluirá un catálogo de productos actualizado y completo, y un sistema de seguimiento de pedidos pudiendo comprobar su ubicación. Además, el software generará informes y estadísticas para que la empresa pueda tomar decisiones sobre su inventario y sus procesos de producción y entrega.

Actualmente, la empresa 'Frutas y Verduras Frescas S.A.' tiene dificultades para llevar un control efectivo de los pedidos, la gestión de los productos y entregas de productos a sus clientes, lo que genera retrasos en las entregas y dificulta la atención al cliente.

Esta empresa es una distribuidora mayorista de productos agrícolas en Montevideo, Uruguay. Esta obtiene sus productos de una red de proveedores locales y regionales que se dedican a la producción y cosecha de frutas y verduras frescas.

La empresa cuenta con una planta de almacenamiento y procesamiento de productos en las afueras de la ciudad de Montevideo. Allí, los productos se reciben, se clasifican y se preparan para su distribución a los clientes.

Los clientes de 'Frutas y Verduras Frescas S.A.' son principalmente comercios mayoristas, supermercados, restaurantes y hoteles en la ciudad de Montevideo y sus alrededores. La empresa tiene una amplia base de clientes en la zona y busca expandirse a otras regiones del país.

Los productos se distribuyen a los clientes utilizando una flota de camiones de la empresa, que realizan entregas diarias según los pedidos realizados por los clientes. Los pedidos se realizan por teléfono o por correo electrónico y se coordinan con el departamento de logística de la empresa.

Con el uso de este software, se espera mejorar la eficiencia en la gestión de pedidos y entregas, reducir los errores en los pedidos, mejorar la atención al cliente y aumentar las ventas de la empresa 'Frutas y Verduras Frescas S.A.' en el mercado local de Montevideo y sus alrededores.

# Anteproyecto

# Introducción

La empresa 'Frutas y Verduras Frescas S.A.' es una distribuidora mayorista de productos agrícolas en Montevideo, Uruguay, que se dedica a proveer a comercios mayoristas, supermercados, restaurantes y hoteles en la ciudad y sus alrededores. Sin embargo, actualmente enfrenta dificultades para llevar un control efectivo de los pedidos, la gestión de los productos y entregas, lo que genera retrasos en las entregas y dificulta la atención al cliente.

Por lo tanto, se plantea la necesidad de desarrollar un software que permita a los clientes realizar sus pedidos de manera más eficiente y sencilla, a través de una plataforma en línea. Además, se busca mejorar el control de los pedidos, la atención al cliente, los tiempos de entrega y garantizar que el pedido entregado sea el correcto. Para ello, el software incluirá un catálogo de productos actualizado y completo, y un sistema de seguimiento de pedidos para comprobar su ubicación.

Además, el programa permitirá llevar un mejor control de los productos y sus características, como su origen, pesticidas utilizados, fecha de caducidad, fertilizantes utilizados, fecha de producción, stock entre otros datos relevantes. Con este software, se espera que la empresa pueda tomar decisiones informadas sobre su inventario y sus procesos de producción y entrega, y así mejorar su eficiencia y aumentar las ventas en el mercado local de Montevideo y sus alrededores.

# Presentación del cliente

Frutas y Verduras Frescas S.A. es una empresa dedicada a la distribución y venta de frutas y verduras frescas, fundada en Montevideo en el año 2005. La empresa cuenta con una amplia trayectoria en el sector y se ha consolidado como líder del mercado, gracias a su compromiso con la calidad y la innovación constante en sus procesos productivos.

El equipo de Frutas y Verduras Frescas S.A. está conformado por un grupo de profesionales altamente capacitados y especializados en diversas áreas. Entre ellos se encuentran ingenieros agrónomos, técnicos en producción, expertos en logística y distribución, y personal de ventas y marketing.

La empresa cuenta con una amplia red de proveedores en todo el territorio nacional, lo que le permite asegurar la disponibilidad y la calidad de sus productos durante todo el año. Además, ha desarrollado alianzas estratégicas con productores locales y regionales para ofrecer una amplia variedad de frutas y verduras frescas de alta calidad y de temporada.

Los clientes de Frutas y Verduras Frescas S.A. son principalmente comercios mayoristas, supermercados, restaurantes y hoteles en la ciudad de Montevideo y sus alrededores. La empresa tiene una amplia base de clientes en la zona y busca expandirse a otras regiones del país. Los pedidos son realizados por los clientes a través de llamadas telefónicas o correos electrónicos. Los pedidos son procesados y entregados dentro de los tiempos acordados con los clientes.

Los productos son recolectados por los camiones de Frutas y Verduras Frescas S.A. y transportados a la planta de procesamiento y almacenamiento de la empresa, esta se encuentra en las afueras de la ciudad de Montevideo. Allí, los productos son recibidos y sometidos a un proceso de selección, clasificación y almacenamiento. El almacenamiento se realiza en cámaras refrigeradas y las condiciones de temperatura y humedad son monitoreadas constantemente para asegurar la calidad y frescura de los productos.

La empresa cuenta con una flota de camiones refrigerados, también conocidos como camión frigorífico, estos se encargan de la distribución de los productos y están diseñados para mantener una temperatura específica y conservar la mercancía fresca hasta su llegada. Los camiones frigoríficos se utilizan para carga perecedera, como frutas y verduras, que deben llegar en óptimas condiciones. La empresa cuenta con una gran variedad de camiones, marcas y diferencias en su capacidad, pero aproximadamente la carga útil de cada camión es de 31.000 kg y un cubicaje de 85 m3. Los camiones son operados por choferes capacitados para manejar productos frescos y mantener las condiciones óptimas de transporte. El personal de la empresa también se encarga de la carga y descarga de los productos en los camiones, asegurando que los productos sean transportados y entregados sin dañar su calidad. La manera en la que se calcula los precios es diferente dependiendo el producto, por ejemplo, en algunos productos como indicador de cuanto se ha de pagar se usa el pesaje, y en otros por unidad. El tomate en esta empresa se vende por kg, y la piña se vende por unidad.

Además de la distribución de frutas y verduras frescas, Frutas y Verduras Frescas S.A. también ofrece servicios de asesoramiento a sus clientes en cuanto a la selección de productos y su almacenamiento. El personal de la empresa está capacitado para asesorar a los clientes en cuanto el mejor método de almacenar los productos para maximizar su vida útil y calidad.

Los servicios que ofrece Frutas y Verduras Frescas S.A. están enfocados en brindar soluciones integrales a sus clientes, que van desde la producción hasta la distribución y la venta de los productos. Algunos de los servicios que ofrece son:

Producción y cultivo: La empresa cuenta con un equipo de profesionales especializados en producción y cultivo de frutas y verduras, que trabaja en conjunto con los productores locales y regionales para asegurar la calidad y la frescura de los productos. Se realizan estudios de suelos y se aplican técnicas de agricultura sostenible para garantizar la producción de alimentos saludables y de alta calidad.

Logística y distribución: Frutas y Verduras Frescas S.A. cuenta con una flota propia de vehículos refrigerados y un sistema de logística eficiente que garantiza la entrega de los productos en tiempo y forma. Además, ofrece servicios de almacenamiento y distribución a sus clientes, lo que les permite mantener un stock de productos frescos durante todo el año.

Ventas y marketing: La empresa cuenta con un equipo de ventas y marketing altamente capacitado que trabaja en conjunto con sus clientes para identificar oportunidades de negocio y desarrollar estrategias de venta efectivas. Se realizan estudios de mercado y se diseñan campañas de promoción y publicidad para dar a conocer los productos y aumentar su demanda.

Asesoramiento técnico integral: La empresa brinda asesoramiento a sus productores especializado en agricultura de precisión y nutrición vegetal, diseñando estrategias adecuadas para cada tipo de cultivo y ambiente. De esta manera la empresa se asegura de brindar los conocimientos para asegurar que las cosechas sean de calidad.

Innovación y tecnología: Frutas y Verduras Frescas S.A. está comprometida con la innovación constante en sus procesos productivos, por lo que invierte en tecnología de última generación para garantizar la calidad y la frescura de los productos. Se realizan estudios de nuevas técnicas de producción y se aplican tecnologías de vanguardia en la cadena de producción para garantizar la calidad y la frescura de los productos.

La empresa tiene un enfoque en la sostenibilidad y el medio ambiente. Trabaja de manera cercana con los productores locales y regionales para asegurar que los productos sean cultivados de manera sostenible y respetuosa con el medio ambiente. Además, la empresa ha implementado prácticas ambientales en su propia planta de procesamiento y almacenamiento, como la utilización de sistemas de energía renovable y la reducción de residuos.

Frutas y Verduras Frescas S.A. es una empresa en constante evolución y mejora. Siempre está buscando maneras de mejorar sus procesos y servicios para sus clientes. La empresa ha implementado tecnologías de información y comunicación en su operación, y está en proceso de implementar un software de gestión de pedidos para mejorar su capacidad de procesar pedidos y satisfacer las necesidades de los clientes de manera más eficiente.

# Presentación del problema

El problema principal que la empresa presenta son los retrasos o perdidas de los pedidos a entregar causados por una mala gestión de estos, como problemas de ubicación o de organización, que llevan a entregas tardías, una cantidad equivocada de producto o entregas en diferentes localizaciones. La empresa necesita un sistema para poder gestionar de manera correcta los envíos de los pedidos, y donde el cliente pueda comprobar los detalles, la ubicación, y el tiempo estimado de entrega de su pedido.

Otro problema es la atención al cliente, esta se realiza por teléfono o mail, lo que a veces puede resultar en problemas de entendimiento, lo que ocasiona que el cliente reciba un pedido no correspondido. La empresa requiere de un sistema donde los clientes puedan pedir sus pedidos sin necesidad de depender de otra persona, aquí los clientes podrán seleccionar todo lo que necesiten, y si el cliente pide de más de lo que se tiene en stock se le notificara tanto al cliente como a los administradores, de esta forma minimizando los malentendidos

La empresa cuenta con un mal control de los productos que importan, como la ubicación, estado, fertilizantes, fecha de producción, stock, etc. Se necesita un sistema donde se puedan ingresar las características de los cargamentos importados para así tener un mejor control a la hora de vender los productos.

# Lista de necesidades

Luego de varias reuniones concluimos en las siguientes necesidades del cliente:

1. La empresa debe poder gestionar los pedidos.
2. Los clientes deben poder comprobar los detalles de sus pedidos.
3. Los clientes deben poder pedir producto/s sin la necesidad de otra persona.
4. Los clientes podrán acceder a todo el catálogo que ofrece la empresa.
5. La empresa debe poder controlar los productos importados con sus características.
6. La empresa debe poder generar informes y estadísticas sobre las ventas y el rendimiento de los productos para tomar decisiones informadas.
7. El sistema debe contar con un sistema de seguimiento de entregas para que los clientes puedan saber en qué momento recibirán sus pedidos.
8. La empresa debe poder realizar un seguimiento de los productos en stock y recibir notificaciones cuando se acerquen a su fecha de caducidad o se estén quedando sin stock.
9. El sistema debe contar con una funcionalidad de calificación y revisión de productos para que los clientes puedan compartir sus experiencias con otros clientes.

# Actores Involucrados

Luego de varias reuniones concluimos en los siguientes actores involucrados:

* Administrador (Dueño de Frutas y Verduras Frescas S.A.) Tendrá acceso a todo el sistema de administración, donde podrá observar todos los procesos y podrá consultar sobre estadísticas de la empresa, además de poder asignar roles.
* Administrador de productos (Gestores de stock de Frutas y Verduras Frescas S.A.) Tendrán acceso al sistema de gestión de stock, en el cual podrán administrar las distintas frutas importadas.
* Usuario (Clientes de Frutas y Verduras Frescas S.A.) Tendrán acceso al catálogo de productos de la empresa, además podrán comprobar la ubicación de su pedido/s.
* Administrador de pedidos (Encargados de envíos de Frutas y Verduras Frescas S.A.) Tendrán acceso al sistema de gestión de pedidos, donde podrán comprobar los pedidos pendientes, actualizar el estado del pedido de los usuarios y organizar los pedidos para que sea enviado al Administrador de flota.
* Administrador de flota (Gerentes de flota de Frutas y Verduras Frescas S.A.) Tendrán acceso a la lista de pedidos confirmados y a la gestión de camiones y camioneros, donde gestionara los pedidos y se entregarán a los usuarios con la flota de camiones de envíos, una vez entregado finalizaran el estado del pedido.

# Lista de Objetivos

1. Evitar retrasos en los pedidos
2. Mejorar los tiempos de entrega de pedidos en un 15 %.
3. Aumentar la precisión de los datos a almacenar en un 50 %.
4. Optimizar los tiempos de búsqueda de datos un 25% para el personal.
5. Mejorar la eficiencia del sistema de gestión de inventario
6. Aumentar la eficiencia del sistema de gestión de pedidos
7. Mejorar la comunicación con el cliente un 50%.
8. Asegurar una experiencia de pedido más intuitiva y fácil para los clientes.
9. Mejorar la atención al cliente.
10. Aumentar las ventas un 10%.
11. Reducir el costo de los envíos en un 5%.
12. Aumentar la eficiencia en la gestión del inventario en un 20%.
13. Ofrecer a los clientes la opción de recibir actualizaciones en tiempo real sobre el estado de sus pedidos.
14. Incrementar la satisfacción del cliente en un 15%.
15. Disminuir los errores de empaquetado en un 10%.
16. Implementar un programa de fidelización de clientes para mejorar la retención y lealtad de los mismos.
17. Aumentar la transparencia en el proceso de entrega para los clientes.

# Requerimientos

## Requerimientos funcionales

1. El sistema permitirá a los usuarios registrarse e ingresar en su cuenta personal proporcionando la información necesaria. Los admin acedaran al sistema ingresando sus datos.
2. El sistema permitirá a los usuarios y admins cambiar la contraseña de su cuenta.
3. El sistema deberá encriptar las contraseñas de los usuarios y admins como método de seguridad.
4. El software permitirá a los usuarios ver el catálogo de los productos, y además se podrán destacar los productos más vendidos, en oferta, o en temporada.
5. El software permitirá a los usuarios pedir los productos que ofrece la empresa desde un catálogo online.
6. El sistema permitirá a los usuarios filtrar los productos por categoría, precio, nombre o cualquier otra característica para facilitar la búsqueda de productos específicos.
7. El sistema deberá poder dejar ver a los usuarios el estado actual de su pedido/s además de sus pedidos ya entregados.
8. El sistema enviara una notificación al usuario cuando su pedido cambie de estado.
9. El sistema debe tener un mecanismo de validación de datos para evitar errores al momento de ingresar información de pedidos y clientes.
10. El sistema debe tener un sistema de seguimiento de gastos que permita a los usuarios ver cuánto han gastado en la plataforma y ver su historial de compra.
11. El sistema tendrá distintos administradores:
    1. Administrador global: Tendrá acceso total al software, permitiéndole gestionar todas las listas existentes. Este administrador podrá comprobar el estado actual de la empresa mediante consultas de stock o de ventas realizadas. Además, este administrador podrá asignarles crear a otros administradores.
    2. Administrador de productos: Tendrá acceso al sistema de gestión de stock, además este también se encargará de la administración de las distintas frutas que sean importadas.
    3. Administrador de pedidos: Tendrá acceso al sistema de gestión de pedidos, en el cual podrá comprobar pedidos pendientes y confirmar sus detalles, además de actualizar el estado del pedido. Si los datos del pedido son correctos el Administrador de pedidos le enviara el pedido y los detalles a un Administrador de flota.
    4. Administrador de flota: Tendrá acceso a la lista de pedidos confirmados por el Administrador de pedidos y al gestor de camiones y camioneros. El Administrador de flota gestionará esos pedidos para que sean cargados a un camión y puedan partir hacia su destino, una vez entregado finalizaran el estado del pedido.
12. El software permitirá a los admins globales consultar la lista de clientes y eliminar uno o varios de estos.
13. El sistema debe tener un gestor de admins controlado por el admin global.
14. El sistema tendrá un gestor de camiones y camioneros, en el cual tendrá todos los datos de los camiones y camioneros, además de su estado libre u ocupado.
15. El sistema deberá tener un gestor de granjas, fertilizantes, pesticidas y depósitos.
16. El sistema deberá tener un gestor de los productos registrados dependiendo de que granja provienen (Producen), en esta tabla se registrara el stock, la fecha de producción el precio, los pesticidas y fertilizantes utilizados.
17. El sistema debe contar con una funcionalidad para la gestión de inventario, el cual será controlado por un administrador de producto (también podrá acceder el administrador global) permitiendo a la empresa agregar, eliminar o modificar un producto de esta forma llevar un control actualizado de la cantidad de productos disponibles y recibir alertas cuando se estén agotando el stock o se acerca la fecha de caducidad.
18. El sistema tendrá un gestor de pedidos el cual dejará a los administradores de pedidos y global poder ver el estado de todos los pedidos actuales y concluidos, en este gestor podrán eliminar pedidos, modificar pedidos, confirmar pedido, asignar pedido y finalizar pedido.
    1. Se podrá eliminar tanto pedidos concluidos como pendientes, pero para eliminar pedidos pendientes se tiene que dar una razón válida y esta solicitud de eliminación de pedido debe ser revisada por un admin global. Si es eliminado por un admin global no se requeriría la confirmación, pero se pedirá el motivo por lo que se borra el pedido
    2. Confirmar pedido: Cuando un usuario hace un pedido tiene que ser revisado y confirmado por un administrador de productos.
    3. Asignar pedido: Cuando un administrador de productos confirma un pedido, un administrador de flota debe asignarle un camión y camionero.
    4. Finalizar pedido: Cuando un pedido es entregado a un cliente, el pedido debe ser finalizado por un administrador de flota.
19. El sistema debe contar con un mecanismo de alertas para los usuarios, en cual los admins global y flota podrán avisar a los usuarios en caso de que se prevea un retraso en la entrega del pedido.
20. El sistema debe contar con una funcionalidad para la generación de reportes de ventas y estadísticas, que permita a la empresa analizar el rendimiento de sus productos y tomar decisiones informadas.
21. El sistema deber contar con una sección de auditoría, en el cual se deben registrar todos los movimientos del sistema con la base de datos, esta solo será accesible por el admin global.

## No Funcionales

1. El sistema debe accederse desde los navegadores web.
2. El sistema debe estar conectado a internet.
3. El sistema ya debe tener creado al menos un administrador global.
4. El sistema debe estar disponible en su totalidad en español.
5. El sistema debe ser fácil de usar y contar con una interfaz intuitiva para los usuarios, tanto para los clientes como para el personal de la empresa.
6. El sistema deberá correr en Windows 8, 10 y 11.
7. EL sistema deberá guardar los datos en una base de datos.

# Descripción del entorno

La empresa está abierta de lunes a viernes en el horario de 8 a 12 hrs y de 14 a 18 hrs, y el sábado de 8 a 12 hrs. Los empleados de la empresa a diario a trabajar en ese horario y realizan sus tareas.

Los camioneros de importación se encargan de ir a distintas zonas del país a traer cargamentos de frutas y verduras para llevarlas a la zona de descarga de la empresa.

Los almaceneros se encargan de descargar las flotas de camiones y organizar el stock dentro de los almacenes, todo esto al mando de los gestores de stock.

Los gestores de stock se encargan de administrar los distintos cargamentos que llegan a la empresa y anotar las características para que queden guardadas.

Los encargados de envíos se encargan de revisar los mails y responder las llamadas telefónicas de los clientes que desean hacer un pedido, corroborar el stock, anotar todos los datos, y enviárselos a los gerentes de flota para que se encarguen del envió

Los gerentes de flota se encargan de armar los pedidos que se van a enviar y anotarlos en la lista del camionero de envíos.

Los camioneros de envíos se encargan de hacer los viajes hasta las distintas ubicaciones de los pedidos y entregarlos, para luego comunicarle a un gerente de flota.

# Alcance y limitaciones

## Alcances

El sistema tendrá un sistema de login para distinguir a los administradores de los usuarios.

El sistema obligara al usuario a registrarse para hacer un pedido.

El sistema tendrá una interfaz amigable la cual mostrará ayudas al usuario en pantalla.

Los usuarios que se registren en el software podrán ver sus pedidos pasados y el estado de sus pedidos actuales, pudiendo rastrearlos.

Se diferencian varios tipos de Administradores:

* Los Administradores global teniendo acceso total al programa.
* Los administradores de productos tendrán acceso al sistema de gestión de productos y stock.
* Los Administradores de pedidos tendrán acceso al sistema de gestión de pedidos.
* Los Administradores de flota tendrán acceso a la lista de pedidos confirmados, asignar un camión, finalizar pedidos y al gestor de camiones y camioneros.

Los usuarios de tipo usuario solo podrán ver la información relacionados con ellos mismos.

El sistema permitirá a los administradores globales ver estadísticas relacionadas con las ventas.

El sistema permitirá a los administradores globales y a los administradores de pedidos buscar uno o varios pedidos por cualquiera de sus características.

## Limitaciones

En principio el sistema no contará con registro y logueo de errores.

El sistema no es de compra por lo que no pedirá método de pago y no efectuará una factura.

# Estudio de alternativas

## Alternativa 1:

Lenguaje de programación a utilizar C#, backend y estilo ASP.NET Core, frontend auto generado por ASP.NET Core en HTML/CSS/JavaScript con Bootstrap. Los datos se guardarán en una base de datos Microsoft SQL server Express 2018, la totalidad del programa se encontrará en un server propio en uno de los depósitos de la empresa.

## Alternativa 2:

Utilizar Genexus, esta es una plataforma de desarrollo que facilita la creación del Software. El programa se encontrará en un servidor alquilado de GoDaddy.

## Arquitectura



Para ambas alternativas la computadora del usuario o administrador se conecta y lleva la información atreves del internet (ver diagrama), esta es recibida por el server procesándola y renviando la respuesta del programa. La diferencia entre las alternativas es en cuanto a los servidores, en una el servidor es de terceros y en la otra el servidor es comprado e instalado por la empresa.

## Análisis de factibilidad

### Alternativa 1:

#### Operativa:

Este software implica un gran cambio a la hora de la gestión de datos de la empresa.

¿Pero esto implica que se requiere una alta capacitación del personal para el uso adecuado del software?

No, dado que el software se realizará con los requerimientos de los futuros administradores de la página, por lo que de alguna manera ya conocerán de qué es capaz el programa, lo que sí es necesario es una o varias etapas de capacitación para capacitar a los trabajadores de las distintas áreas para entender lo que tienen que hacer y sus limitaciones. Por lo tanto, operativamente es factible el proyecto.

#### Técnica:

El software debe ser hosteado en algún sitio para que puedan acceder tanto usuarios como administradores, dado que es una empresa líder en el mercado, la información es muy valiosa por lo tanto para proteger sus datos lo mejor es crear un servidor en alguno de sus depósitos, dado que si se usa un servicio público o privado hay la probabilidad de que se filtre la información a través de un hackeo o que la propia empresa filtre la información. De esta manera se vuelve responsabilidad de la empresa su seguridad y no la de terceros, esta idea cumple con los requerimientos, pero se deben instalar y ajustar los servidores por lo que podría aumentar el precio del proyecto, pero sigue siendo factible y disminuye los riesgos causados por terceros.

#### Legal:

El software que se desarrolla utiliza una base de datos y en Uruguay las bases de datos que contienen datos de personas físicas y/o jurídicas se deben inscribir. El trámite se puede realizar por internet o físicamente y no tiene costo. Por lo tanto, este proyecto es factible legalmente.

#### Económica:

Toda la tecnología utilizada para el desarrollo es Open-source por lo que no es necesario de una licencia para la creación de Software, por lo tanto, no hay que utilizar capital para esto, aparte los miembros del proyecto ya cuentan con computadoras capaces de utilizar correctamente estos softwares.

En donde si hay que utilizar capital es en instalar el servidor y en comprar computadoras básicas para que los administradores puedan utilizar el programa.

##### Estimación de esfuerzo

Para calcular el esfuerzo requerido para completar el proyecto, utilizaremos el método de “Puntos de fusión”.

Este enfoque nos permite cuantificar el trabajo a realizar al asignar un valor numérico al tamaño de las funcionalidades del software mediante la suma ponderada de sus características. La evaluación se lleva a cabo utilizando los siguientes parámetros:

* Entradas externas: Se cuenta cada entrada que ofrece datos al software.
* Salidas externas: Son las pantallas, reportes, mensajes de error, sin contar los elementos dentro de cada salida.
* Consultas externas: Son las entradas interactivas en respuestas a salidas interactivas.
* Archivos lógicos internos: Es el número de archivos maestros lógicos utilizados (no se cuentan los índices).
* Archivos interface externa: Cada archivo o estructura destinada a transmitir datos a otros sistemas.

Los puntos de función permiten estandarizar las mediciones del tamaño del software, obteniendo estimaciones de mayor exactitud, frente a las que podemos obtener basándonos en nuestra experiencia.

El conteo total es la suma de todos los puntos de función que se obtienen en la siguiente tabla:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Factor de ponderación | | | |
| Parámetro de medición | Bajo | Medio | Complejo | Subtotal |
| Número de entradas externas | x3 | x4 | x6 |  |
| Número de salidas externas | x4 | x5 | x7 |  |
| Número de consultas externas | x3 | x4 | x6 |  |
| Número de archivos lógicos internos | x7 | x10 | x15 |  |
| Número de interfaces externas | x5 | x7 | x10 |  |
| Cuenta = Total |  | | | |

Fi es un factor de ajuste que surge de las respuestas del siguiente cuestionario.

Cada respuesta tiene un valor de 0 a 5.

Escala:

0. Sin influencia

1. Incidental

2. Moderado

3. Medio

4. Significativo

5. Esencial

Cuestionario:

1) ¿El sistema requiere respaldo y recuperación confiables?

2) ¿Se requieren comunicaciones de datos especializadas para transferir información a la aplicación, u obtenerla de ella?

3) ¿Hay funciones distribuidas de procesamiento?

4) ¿El desempeño es crítico?

5) ¿El sistema se ejecutará en un entorno existente que tiene un uso pesado de operaciones?

6) ¿El sistema requiere entrada de datos en línea?

7) ¿La entrada de datos en línea requiere que la transacción de entrada se construya en varias pantallas u operaciones?

8) ¿Los ALI se actualizan en línea?

9) ¿Las entradas, las salidas, los archivos o las consultas son complejos?

10) ¿Es complejo el procesamiento interno?

11) ¿El código diseñado será reutilizable?

12) ¿Se incluyen la conversión e instalación en el diseño?

13) ¿Está diseñado el sistema para instalaciones múltiples en diferentes organizaciones?

14) ¿La aplicación está diseñada para facilitar el cambio y para que el usuario lo use fácilmente?

Entradas externas (36): Login, Registrarse, eliminar usuario, modificar contraseña, modificar dirección, alta de admin, modificar admin, eliminar admin, alta de caminero, modificar caminero, eliminar caminero, alta producto, modificar producto, eliminar producto, alta de camión, modificar camión, eliminar camión, alta de pedido, modificar pedido, eliminar pedido, alta granja, eliminar granja, modificar granja, alta fertilizante, eliminar fertilizante, modificar fertilizante, alta pesticida, eliminar pesticida, modificar pesticida, alta depósito, baja depósito, modificar depósito, alta producen, eliminar producen, modificar producen.

Salidas externas (72): Mostrar mensaje al usuario o admin cuando este tenga una interacción con el sistema. Mensajes de confirmación o rechazo para cada interacción (72).

Consultas externas (11): Validar usuario o admin, listado de productos, listado de usuarios, listado de admins, listado de camioneros, listado de camiones, listado de pedidos, listado granjas, listado fertilizantes, listado pesticidas, listado depósito, listado de producen.

Interface externa (0)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Factor de ponderación | | | |
| Parámetro de medición | Bajo | Medio | Complejo | Subtotal |
| Número de entradas externas | 0x3 (0) | 30x4 (120) | 6x6 (36) | 156 |
| Número de salidas externas | 72x4 (288) | x5 | x7 | 208 |
| Número de consultas externas | 7x3 (21) | 4x4 (16) | x6 (0) | 37 |
| Número de archivos lógicos internos | 10x7 (70) | 3x10 (30) | x15 | 100 |
| Número de interfaces externas | 0x5 (0) | 0x7 (0) | 0x10 (0) | 0 |
| Cuenta = Total | 501 | | | |

|  |  |
| --- | --- |
| Pregunta | Pregunta Respuesta (0 – 5) |
| 1 | 1 |
| 2 | 2 |
| 3 | 2 |
| 4 | 1 |
| 5 | 2 |
| 6 | 3 |
| 7 | 3 |
| 8 | 4 |
| 9 | 2 |
| 10 | 2 |
| 11 | 2 |
| 12 | 1 |
| 13 | 0 |
| 14 | 3 |

Total = 28

Cálculo de puntos de función:

PF (Puntos de función)

PF = 501x [0.65 + (0.01 x28)]

PF = 465,93

Cálculo de líneas de código:

KLDC (Líneas de código)

KLDC = 465,93 x 50 /1000

KLDC = 23,2965

En el desarrollo se estima un esfuerzo de aproximadamente 5,82 KLDC por mes, por lo que estaría programado en 4 meses.

### Alternativa 2:

#### Operativa:

El sistema va a generar cambios a la hora de gestionar los datos de la empresa, este estudio de factibilidad busca analizar los cambios y verificar si el proyecto sigue siendo factible a pesar de ellos.

El proyecto no va a requerir de una gran capacitación para poder entender su funcionamiento, esta estará orientada para los administradores de los distintos sectores y no requerirá mucho tiempo, ya que el sistema será intuitivo y muy sencillo. Para los clientes el sistema será más sencillo e intuitivo y con un par de opciones, así que estos no deberán tener ninguna capacitación para entenderlo.

Podemos decir que el proyecto es operativamente factible, reduciendo el tiempo de ingreso de datos como también el tiempo que el cliente tarda en comprar sus productos.

#### Legal:

Un proyecto se considera factible legalmente si cumple con todas las leyes, regulaciones y normativas aplicables, así como si se han obtenido todos los permisos y licencias necesarios antes de su implementación.

El software que se desarrolla utiliza una base de datos y en Uruguay las bases de datos que contienen datos de personas físicas y/o jurídicas se deben inscribir, El trámite se puede realizar por internet o físicamente y no tiene costo. Este proyecto cumple con todas las leyes, así que se puede considerar que el proyecto es factible legalmente.

#### Técnicamente:

Hace referencia a problemas técnicos que puedan surgir durante la implementación o el funcionamiento del sistema. El proveedor de hosting será la página <https://www.godaddy.com/> el cual cuesta USD 4.99 el primer año, y el resto USD 19.99, costo que el cliente está al tanto y de acuerdo en mantener. Podemos decir que esto no afectará la factibilidad del sistema.

#### Económicamente:

Este estudio analiza la inversión que tendremos que hacer en dinero y recursos durante el desarrollo.

Las herramientas de desarrollo a utilizar son de pago, por lo tanto, el costo de las licencias va para la empresa, hay varios planes mensuales para elegir, pero estos varían entre los 100 USD hasta los 300 USD, además hay que sumarle el precio del server que sería USD 4.99 el primer año, y el resto USD 19.99. Para la empresa no es un gasto muy elevado pero este proyecto busca ser lo más económico posible por lo que este estudio económico dictamina que podría llegar a ser un punto en contra de utilizar esta alternativa.

En cuanto a hardware, ambos integrantes cuentan con computadoras suficientemente potentes para soportar las herramientas de desarrollo de software por lo que no sería un problema el gasto para de equipo.

## Selección de alternativa

La alternativa seleccionada fue la primera, utilizar ASP.NET Core con C# y Microsoft SQL server 2018.

Unas de las razones más importantes para utilizar esta alternativa es que las tecnologías a utilizar son Open-source por lo tanto que no se requiere pagar por las licencias.

Durante la carrera se hicieron varios proyectos con ASP.NET Core con C# y Microsoft SQL parecidos a lo que este proyecto plantea desarrollar, por lo que los alumnos ya cuentan con una buena base para desarrollar este software, además es una buena idea para refrescar lo aprendido anteriormente.

C# es un lenguaje simple, eficaz y con seguridad de tipos que permite desarrollar aplicaciones con velocidad y mantener la expresividad y la elegancia de los lenguajes de estilo de C, además, este lenguaje de programación es ampliamente utilizado para el desarrollo de videojuegos, aplicaciones (tanto de escritorio, como web y Mobile) y realidad virtual.

# Análisis de riesgo

En cualquier proyecto, ya sea de desarrollo de software o de otro tipo, existen riesgos potenciales que pueden afectar su progreso. Por esta razón, es importante realizar un análisis previo al inicio del proyecto para identificar estos riesgos y tener planes de contingencia para mitigarlos o controlarlos en caso de que ocurran. Sin embargo, también es importante tener en cuenta que pueden surgir riesgos imprevistos durante la fase de construcción y es crucial detectarlos tempranamente para minimizar su impacto en el cronograma y los costos del proyecto.

**Probabilidad e impacto**:

Para cada riesgo se deben definir la probabilidad y el impacto del mismo.

La probabilidad puede ser:

* Muy alta
* Alta
* Moderada
* Baja
* Muy baja

El impacto puede ser:

* Critico: Gran impacto al proyecto, pudiendo aumentar costos y tiempos, y significaría el no cumplimiento de los requerimientos y objetivos planteados.
* Alto: Aquí pertenecen riesgos que generan retrasos del proyecto y aumento de gastos, además de traducirse como errores inesperados dentro del sistema.
* Moderado: Retrasos y aumentos de costos, pero las consecuencias de estos riesgos están ligadas a aspectos no fundamentales del sistema.
* Bajo: Leves retrasos y aumentos de los costos.

Riesgos detectados:

|  |  |  |
| --- | --- | --- |
| **Riesgo** | **Probabilidad** | **Impacto** |
| Mala interpretación de necesidades, objetivos y requerimientos. | Moderada | Critico |
| Funcionamiento incorrecto del sistema. | Baja | Critico |
| Abandono del proyecto por parte de un integrante del grupo. | Muy baja | Critico |
| Mala comunicación entre los integrantes del grupo | Moderada | Critico |
| Cambios de administración de la empresa del cliente. | Muy baja | Critico |
| Cambio de tecnología | Baja | Alto |
| Dificultad al estimar tiempos y recursos | Moderada | Alto |
| Incumplimiento de objetivos y requerimientos | Baja | Alto |
| Falta de experiencia en el lenguaje de programación o tecnologías. | Moderada | Moderado |
| Cliente poco satisfecho con el resultado final | Moderada | Moderado |
| Información vulnerable | Baja | Bajo |

* **Mala interpretación de necesidades, objetivos y requerimientos.**

**Probabilidad:** Moderada.

**Impacto:** Critico.

**Descripción:** Los integrantes del grupo poseen poco conocimiento sobre las características y particularidades del entorno donde se ejecutará el software.

**Consecuencia:** El sistema puede llegar a no ser útil para el cliente, debido al incumplimiento de los requerimientos, objetivos y las necesidades, lo que ocasiona el rechazo del sistema por parte del cliente.

**Mitigación:** Asegurarse de dedicar tiempo a la investigación y el análisis de la problemática y sus soluciones, mantener una comunicación fluida con el cliente a través de entrevistas periódicas para mantenerlo al tanto de los avances, y validar cada una de las necesidades, objetivos y requerimientos con el cliente.

**Plan de contingencia:** Programar reuniones con el cliente para repasar requerimientos, objetivos y necesidades.

* **Funcionamiento incorrecto del sistema.**

**Probabilidad:** Baja.

**Impacto:** Critico.

**Descripción:** El sistema presenta fallos en sus funcionalidades.

**Consecuencia:** El sistema no funciona correctamente, bajando su rendimiento y funcionabilidad, pudiendo ser inútil para el cliente.

**Mitigación:** Asegurarse de poseer un buen Plan de Testing, y realizar un análisis de pruebas de caja blanca y negra, validándolo con el cliente.

**Plan de contingencia:** Dedicar algunas horas a encontrar y solucionar errores en el sistema.

* **Abandono del proyecto por parte de un integrante del grupo.**

**Probabilidad:** Muy baja.

**Impacto:** Critico.

**Descripción:** Puede ser que, por razones ajenas a uno de los integrantes del grupo, el otro integrante abandone el proyecto, lo cual imposibilita avanzar en el proyecto.

**Consecuencia:** La finalización del proyecto se vería en peligro, ya que el cronograma de trabajo se basa en que el proyecto se realizara por dos integrantes.

**Mitigación:** Cuando no se trabaje en equipo, se deberá informar uno al otro del trabajo realizado, así ambos tendrán una idea general del sistema y su funcionamiento, por si un integrante abandona el proyecto, este podría ser concluido.

**Plan de contingencia:** Hacer ajustes en el cronograma de trabajo, descartando funcionalidades extras y enfocarse en los puntos importantes.

* **Mala comunicación entre los integrantes del grupo.**

**Probabilidad:** Moderada.

**Impacto:** Critico.

**Descripción:** La escasa comunicación entre los integrantes del equipo.

**Consecuencia:** Puede generar malentendidos o desorganización a la hora de ejecutar tareas, lo que puede generar errores, poca eficiencia, y muchas demoras.

**Mitigación:** Se debe reportar con una frecuencia los avances logrados al otro compañero del grupo y al cliente.

**Plan de contingencia:** Aumentar la frecuencia de reuniones con el compañero.

* **Cambios de administración de la empresa del cliente.**

**Probabilidad:** Muy baja.

**Impacto:** Critico.

**Descripción:** Cambio de personal administrativo en la empresa donde se va a desarrollar el software.

**Consecuencia:** La nueva administración puede considerar innecesario el software, lo que finalizaría el proyecto y evitaría su conclusión.

**Mitigación:** Realizar reuniones informativas con el nuevo personal administrativo y explicarles el proyecto.

**Plan de contingencia:** Crear un contrato con el cliente para que se comprometa a recibir y financiar el software terminado.

* **Cambio de tecnología.**

**Probabilidad:** Baja.

**Impacto:** Alto.

**Descripción:** Son necesarios cambios de tecnología para llevar a cabo el proyecto.

**Consecuencia:** El sistema no pudo implementarse correctamente con los sistemas actuales.

**Mitigación:** Se deberá usar tecnología actualizada para la ejecución del proyecto en lo posible, según las opciones del cliente y el equipo de desarrollo.

**Plan de contingencia:** Se debe quedar en acuerdo con el cliente que él debe asumir los costos extra por el uso de nuevas tecnologías.

* **Dificultad al estimar tiempos y recursos.**

**Probabilidad:** Moderado.

**Impacto:** Alto.

**Descripción:** La falta de experiencia del equipo en análisis, diseño y desarrollo de software puede dar lugar a problemas en la estimación del tamaño del proyecto, así como en los tiempos estimados para su elaboración y en la identificación de los recursos necesarios.

**Consecuencia:** Modificaciones en el cronograma de trabajo.

**Mitigación:** Hacer el diagrama de Gantt muy completo, en el que se tenga en cuenta las posibles pérdidas de tiempo por imprevistos.

**Plan de contingencia:** Establecer tiempos mayores de los estimados para cada una de las tareas.

* **Incumplimiento de objetivos y requerimientos.**

**Probabilidad:** Baja.

**Impacto:** Alto.

**Descripción:** El sistema no posee algunas funcionalidades planificadas y tampoco cumple con los objetivos planteados, lo que hace que el funcionamiento del sistema no este completo.

**Consecuencia:** No aporta ningún valor al cliente, ya que los requerimientos y objetivos no están completos debido a la falta de desarrollo.

**Mitigación:** Validación de requerimientos y objetivos por parte del equipo de desarrollo.

**Plan de contingencia:** Programar nuevas reuniones con el cliente y repasar los requerimientos y objetivos, y dedicar tiempo extra en la implementación de estos.

* **Falta de experiencia en el lenguaje de programación o tecnologías.**

**Probabilidad:** Moderada.

**Impacto:** Moderado.

**Descripción:** El equipo de desarrollo no posee la suficiente experiencia en el lenguaje de programación o tecnologías a trabajar.

**Consecuencia:** Errores a la hora de calcular el tamaño de los requerimientos, también mala estimación de tiempo e incumplimiento de algunos de los requerimientos.

**Mitigación:** Analizar y estimar por separado y en conjunto la dificultad de cada funcionalidad para tener un promedio y discutir las diferencias.

**Plan de contingencia:** Programar una reunión con el cliente para llegar a un acuerdo respecto a las entregas y sus tiempos.

* **Cliente poco satisfecho con el resultado final.**

**Probabilidad:** Moderada.

**Impacto:** Moderado.

**Descripción:** Luego de finalizar el desarrollo, el cliente se presenta insatisfecho con el producto final, o no cumple con sus necesidades esperadas o es poco práctico para su operación.

**Consecuencia:** El programa contara con poca o nula utilización, o ni siquiera la implementación en la empresa por el poco valor que aporta.

**Mitigación:** Realizar un chequeo de necesidades, aumentar la frecuencia de reuniones con el cliente para mostrar avances y detectar diversas correcciones.

**Plan de contingencia:** Programar reuniones con el cliente para mostrar avances y tomar nota de las críticas que hace el cliente.

* **Información vulnerable.**

**Probabilidad:** Baja.

**Impacto:** Bajo.

**Descripción:** Se debe brindar seguridad y confidencialidad de los datos de la empresa.

**Consecuencia:** La información privada de la empresa corre riesgo, quedando expuesta a ataques de terceros.

**Mitigación:** Investigar si existe la posibilidad de exponer los datos de la empresa, utilizar buenas prácticas de seguridad además de estar atento a posibles vulnerabilidades.

**Plan de contingencia:** Tomar acciones correctivas en caso de ser necesario.

# Plan de proyecto

## Definición del proceso

### Metodología

Para la planificación de las tareas necesarias para la elaboración de un software de calidad para el cliente, es necesario definir un tipo de metodología que sirva como guía para llevar a cabo el proyecto en tiempo y forma.

Para este proyecto se han tenido en cuenta estas metodologías:

#### Modelo espiral:

El modelo espiral es un enfoque para el desarrollo de software que combina elementos de otros modelos y se enfoca en la gestión de riesgos. Consta de ciclos repetitivos de análisis, evaluación, desarrollo y planteamiento.



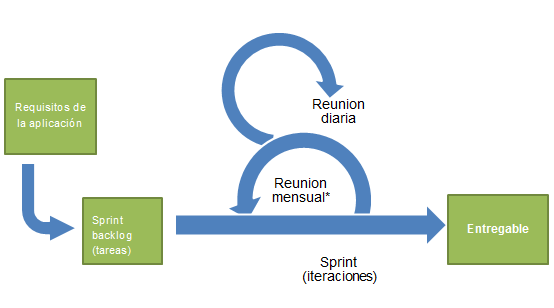
#### Desarrollo incremental

El desarrollo incremental es un enfoque para el desarrollo de software en el que el proyecto se divide en partes más pequeñas llamadas incrementos. Cada incremento representa una parte de la funcionalidad total del producto y se desarrolla por separado. Este enfoque permite entregar rápidamente software operativo y probar sus funciones de forma interactiva.



#### Metodología SCRUM

Scrum es un marco utilizado por los equipos para administrar el trabajo y resolver problemas de forma colaborativa en ciclos cortos. Scrum implementa los principios de Agile como un conjunto concreto de artefactos, prácticas y roles



### Ciclo de vida elegido

Para este proyecto se decidió utilizar la metodología de desarrollo incremental con un ciclo de vida basado en el modelo espiral.

El modelo elegido permitirá dividir el proyecto del sistema en partes pequeñas, para poder tener un mejor control del avance del proyecto y del cumplimiento de los requerimientos.

El modelo espiral permite la reducción de los riesgos asociados al proyecto debido a que los mismos han de ser considerados para cada etapa el desarrollo. Además, fuerza una estrecha comunicación con el cliente reduciendo también los riesgos asociados a la poca comunicación entre el cliente y los desarrolladores.

### Etapas del ciclo de vida elegido

1. Comunicación con el cliente: Mantener un contacto con el cliente para que valide en cada ciclo los requerimientos o los modifique.
2. Planificación: Planificar y administrar los tiempos y recursos necesarios para el desarrollo del ciclo.
3. Análisis de riesgos: Considerar los riesgos que pudieran existir en la etapa de desarrollo y tomar las medidas necesarias para mitigarlos en caso de que existieran.
4. Programación: Desarrollo y testeo de la versión del sistema correspondiente al ciclo.
5. Planificación de la siguiente etapa y entrega al cliente.

### Iteraciones definidas

Posterior a comenzar el ciclo de las iteraciones, se le entregará una copia del Anteproyecto al cliente que podrá validar el inicio del desarrollo. El cliente formará parte de la verificación conjunta de los resultados de cada iteración, lo que permitirá ir realizando los ajustes necesarios.

Iteración Nro. 1:

Creación de Proyecto y la Base de Datos

Creación de la interfaz gráfica

ABM:

* Admin
* Usuario / Cliente
* Camionero
* Depósito

Testing

Consulta con el cliente

Iteración Nro. 2:

ABM:

* Productos
* Camiones
* Granja
* Fertilizante
* Pesticida
* Produce

Login funcional

Testing

Consulta con el cliente

Iteración Nro. 3:

Limitar lo que cada usuario o admin puede hacer o ver

Catalogo

Sistema de pedidos

Testing

Consulta con el cliente

Iteración Nro. 4:

Listados/Reportes

Funcionalidades faltantes

Testing

Iteración Nro. 5:

Validaciones finales

Testing

Ajustes finales

Consulta con el cliente

Grado de satisfacción del cliente

### Integrantes y roles

Los integrantes de este proyecto somos Bruno Ríos y Matías Erramouspe, ambos estudiantes de la carrera de Analista Programador del instituto CTC, con la tutoría del profesor Andrés Klett. Para la elaboración del proyecto, ambos estudiantes realizaron de forma equitativa las tareas de documentación, diseño, desarrollo, análisis y testeo, para así avanzar más rápido en el proyecto. En varias etapas, durante la elaboración, se realizarán reuniones informativas para analizar lo trabajado por cada una de las partes y ajustar los detalles necesarios para concluir un proyecto de software de buena calidad. En las reuniones con el cliente y con el tutor participaran ambos integrantes.

### Descripción y selección de herramientas

En base a lo especificado en el estudio de alternativas, para el desarrollo del proyecto se utilizarán las siguientes herramientas:

* Para la gestión del desarrollo se utilizará Microsoft Proyect, ya que es ideal para la construcción y seguimiento del trabajo.
* Para el desarrollo del código, tanto como de front-end y back-end se utilizará el editor de código Visual Studio 2022 y se utilizará el framework ASP.NET.
* Para el front-end el framework ASP.NET Core permite desarrollar utilizando HTML, CSS y JavaScript. Además, se utilizará la librería de Bootstrap para el diseño responsive.
* Para el back-end ASP.NET Core nos permite desarrollar el código en el lenguaje de C#, el cual utilizaremos.
* Para el desarrollo y mantenimiento de la base de datos en SQL Server utilizaremos Microsoft SQL Server Management Studio.
* Todo tipo de documentación se realizará en Microsoft Word, y la generación de diagramas de clase y casos de uso se utilizará StarUML.

### Plan SQA

El plan de SQA es crear estándares de calidad que se deben cumplir a lo largo del proyecto, ya sea en la documentación, código, o en los diagramas. Además de lo anteriormente mencionado también se encuentra el Plan de testing para poder asegurarse de que el proyecto creado sea de calidad.

#### Estándares definidos y convenciones

##### Documentación

El trabajo se basará en los criterios establecidos por el Documento 302 de CTC "Normas específicas para la presentación de trabajos finales de carrera" y en contendió se utilizará la “Guía de proyectos finales AP CLAEH 2023”.

Se usará Calibri(cuerpo) 11 para la mayor parte del trabajo y Calibri Light (título) de color azul de tamaño 16 para los títulos y entre 13 y 11 para los subtítulos (dependiendo del nivel de subtítulo). Además, se dejará un doble espaciado después de cada título o párrafo

Cada sección ocupará una o más hojas, y no se podrá incluir dos secciones diferentes en la misma hoja. Por ejemplo, al terminar de explicar la sección "Estudio de factibilidad", se dejará el resto de la hoja en blanco y se empezará a escribir la siguiente sección en la hoja siguiente.

Cada punto en el índice estará enlazado con un hipervínculo a la página del documento con el punto mencionado en el índice.

##### Diagrama de clases

Tanto el diagrama de clases como otros diagramas utilizados como ayuda se construirán usando el estándar de modelado UML.

##### Casos de Uso

Los nombres se utilizarán en infinitivo, los actores empiezan con mayúscula, además se tendrá un identificador que comienza con “CU” y el número de caso de uso en tres dígitos, por ejemplo, CU 001.

##### Base de datos

Los nombres de las tablas empezaran en mayúscula y serán en plural y se utilizara el estilo lowerCamelCase para los atributos de la tabla. Cada tabla contendrá un número Identificado (Id) aleatorio y único generado por el programa. El tipo de dato que se utilizaran en los atributos se adaptara dependiendo al tipo de dato requerido y a su tamaño.

##### Desarrollo

A la hora de asignar el nombre de atributos y métodos se utilizará el estilo lowerCamelCase, los nombres de las clases se utilizará el estilo UpperCamelCase y serán en singular. Las interfaces tendrán como prefijo la letra “I” Ejemplo: IUsuario.

##### Interfaz de usuario

Es fundamental que el programa siga la misma línea de diseño durante todo el programa por lo cual el equipo junto al usuario definirá los colores y estilos a utilizar.

Todas las pantallas tendrán un menú fijo, desde el cual se podrá acceder a todas las funcionalidades disponibles para el usuario. Como norma general, las pantallas deben respetar los principios básicos de usabilidad, siendo estas sencillas y de fácil navegabilidad para cualquier usuario.

La creación de la interfaz tendrá como objetivo una adecuada interacción y usabilidad del usuario, orientados a conseguir que el usuario disfrute de una buena experiencia y pueda explorar y encontrar el producto deseado lo antes posible. Además, el diseño debe ser adaptable para ajustarse a la resolución.

Al acceder a la URL del sitio, lo primero que se presentará es una pantalla Home la cual será la pantalla principal, una vez autenticado el usuario o admin este será redirigido a al mismo Home, pero ahora podrá acceder a las distintas páginas y vistas que tenga permitido.

#### Plan de testing

Para comprobar que la calidad del producto sea la mayor posible y evitar errores se realizara un testeo al programa antes de finalizar cada iteración. Estos testeos se realizan regularmente para asegurar que no se escapara ningún error o que se generen nuevos con el código nuevo generado.

Todas las funciones y métodos deben ser puestos a prueba, conociendo de ante mano los parámetros que recibe y que parámetros devuelve. Se debe simular todas las posibles situaciones y usos del programa para poder otorgar una mayor fiabilidad al software desarrollado.

Luego de cerrada la iteración y lista la versión validada, es hora de generar un release para el cliente a un ambiente de pruebas, donde pueda realizar sus pruebas y retroalimentarnos si está acorde a lo esperado o es necesario realizar algún ajuste.

La documentación también será revisada durante el análisis y diseño del proyecto, de esta forma reducir los errores en redacción en el documento y los errores en la programación, además de disminuir las diferencias del código esperado por la documentación y el código que se desarrolló.

##### Pruebas de caja negra

La prueba de caja negra es una forma de probar el software desde el punto de vista del usuario teniendo como objetivo encontrar errores. Para cada ABM se generarán juegos de datos para probar la mayor cantidad de posibilidades de uso del sistema, los casos de pruebas se definirán dentro de la iteración a desarrollar y se validará cerrada la iteración.

##### Pruebas de caja blanca

Los criterios de prueba se establecerán durante la iteración que se va a desarrollar y se verificarán al finalizar la misma. Para las pruebas de caja blanca, se utilizarán en las funcionalidades que contengan los algoritmos del sistema. Así, nos aseguramos de que los cambios en el código no afecten al funcionamiento correcto de los algoritmos, que deben devolver los valores esperados.

##### Control de pruebas

Para cada uno de los formularios:

1. Verificar el acceso al formulario.

a. Comprobar si se puede abrir, y como, probar desde todas las referencias.

b. Probar manejo de usuarios.

c. Comprobar cómo se cierra, que pasa cuando redirige a otros formularios.

2. Verificar el correcto funcionamiento del formulario.

a. Testear botones, campos de texto, listas, validadores y demás elementos.

3. Verificar entrada de datos.

a. Verificar correcto funcionamiento de la validación frontend y backend.

b. Comprobar funcionamientos (ABM, búsquedas, filtros), verificando validaciones

backend. Esto se debe hacer en conjunto con el punto 4.

4. Verificar salida de datos.

a. Probar listados, búsquedas y filtros. Esto se hace en conjunto con el punto 3.

Las pruebas se controlarán mediante una planilla electrónica, en donde se registrará el resultado de cada una de ellas, registrando:

• el número de prueba

• la fecha de la prueba

• la descripción de la prueba

• el resultado esperado

• el resultado obtenido

• la prueba que repite, si corresponden

• un indicador de si la prueba ha sido finalizada con un resultado aceptable

• Módulo en que se realiza la prueba

• Responsable de la prueba

Ejemplo:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Número de prueba | Fecha de la prueba | Descripción de la prueba | Resultado esperado | Resultado obtenido | Antecedentes | Resultado | Modulo | Responsable |
| 1 | 20/6/2023 | Ingreso de un nuevo cliente | Ingresar un cliente | Se consiguió ingresar un cliente | S/D | Aceptable | 0.1.0 | Bruno Ríos |
| 2 | 20/6/2023 | Eliminar cliente | Eliminar cliente | Se elimino el cliente | S/D | Aceptable | 0.1.0 | Matías Erramouspe |

### Plan de SCM

El objetivo del plan de SCM es controlar los cambios del software, centralizar la información, y también definir y organizar ramas de trabajo.

#### Gestión de configuración

Existen muchas herramientas que permiten cumplir con el objetivo del plan de SCM, en nuestro caso se decidió utilizar la herramienta GitHub, la cual es un sitio web donde se pueden crear repositorios y subir código fuente. Elegimos esta opción ya que el grupo de desarrollo cuenta con experiencia en la plataforma, además de ser gratuita y permitir guardar todo el código fuente.

#### Control de versionado

Hay distintos tipos de versionado, en nuestro caso se decidió utilizar el versionado semántico el cual nos resulta fácil de usar y de entender.

El versionado será de forma incremental como por ejemplo 0.1.0, 0.1.1, 0.1.2 teniendo prohibido pasar a una versión más baja para tener un orden en el versionado.

El versionado iniciara en la versión 0.1.0 en la que el 1 indica el número de iteración, una vez en producción el versionado iniciara en la versión 1.0.0.

A nivel de Git tendremos una rama principal llamada “main” en la cual se subirán los commits.

Ejemplo:

1- Se crea la rama 0.1.0 en la rama de “main”.

2- Los integrantes trabajan sobre la rama 0.1.0.

3- Finalizado el trabajo sobre la iteración número 1 esta rama se fusionará a la rama “main”.

4- Se crea la rama 0.2.0 y se repite el ciclo.

### Plan de capacitación

En el caso del grupo de desarrollo de software no se va a plantear un plan de capacitación, ya que el grupo ya posee experiencia en las tecnologías a utilizar.

La capacitación al usuario no será muy extensa, la interfaz del programa será intuitiva y fácil de aprender a utilizarla, además se tomará en consideración que el usuario tenga conocimientos previos sobre la empresa y su funcionamiento. La capacitación será acompañada de un manual de uso que presentará ayuda al usuario en cada parte del programa a utilizar.

### Cronograma de trabajo y criticidad

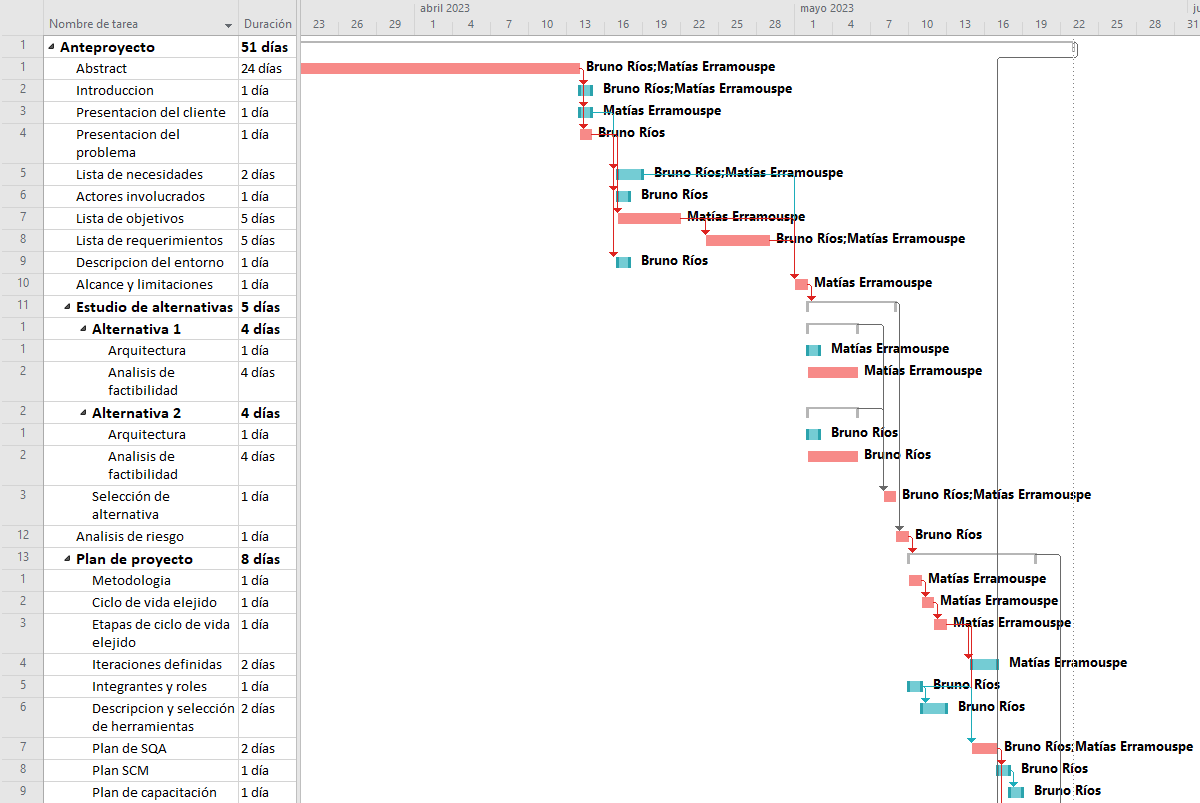


Diagrama de Gantt (parte 1)

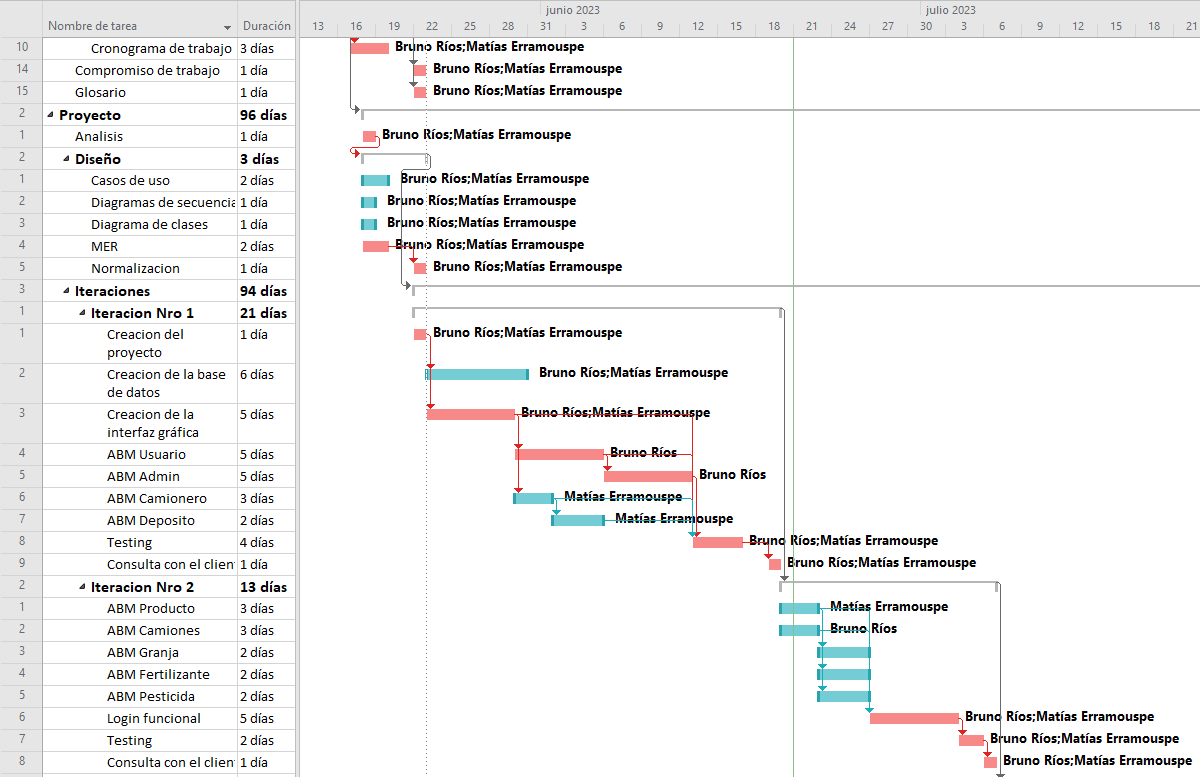


Diagrama de Gantt (parte 2)

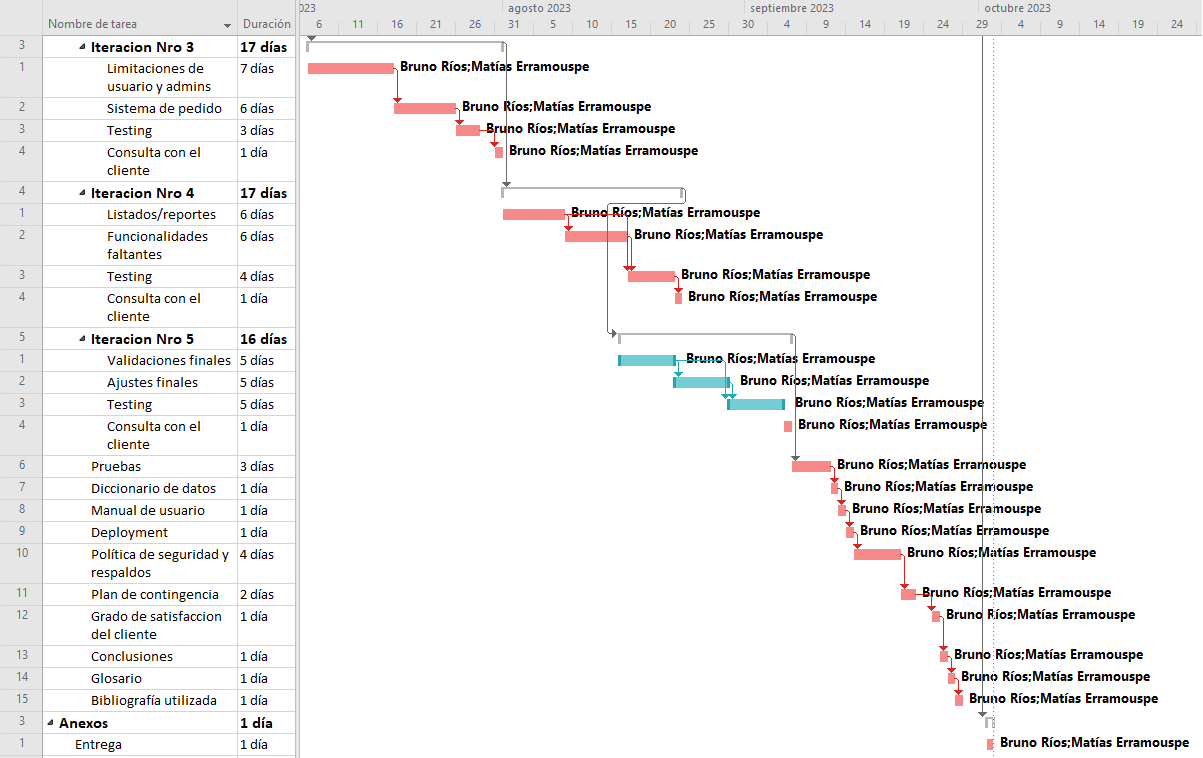


Diagrama de Gantt (parte 3)

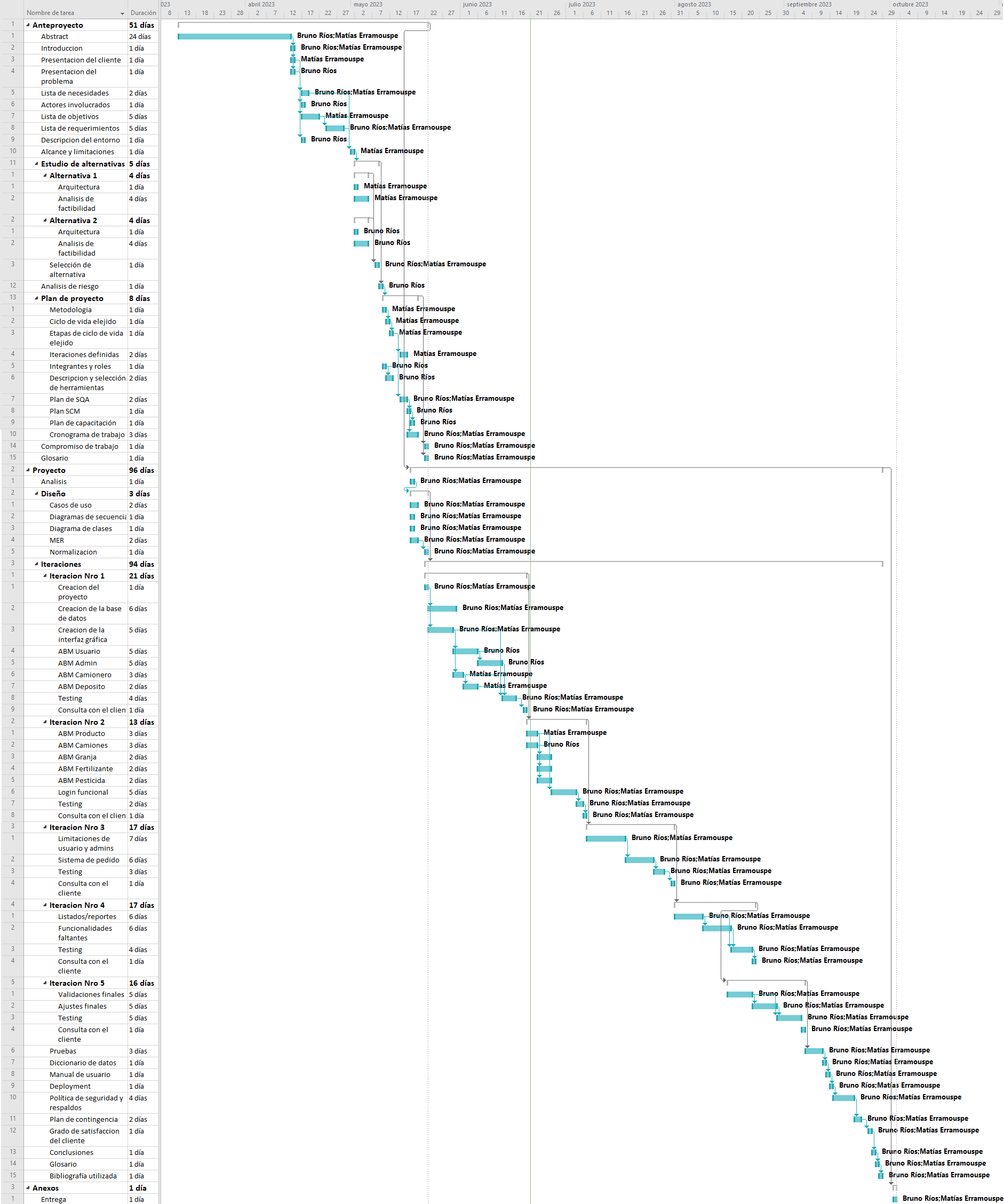


Diagrama de Gantt (completo)

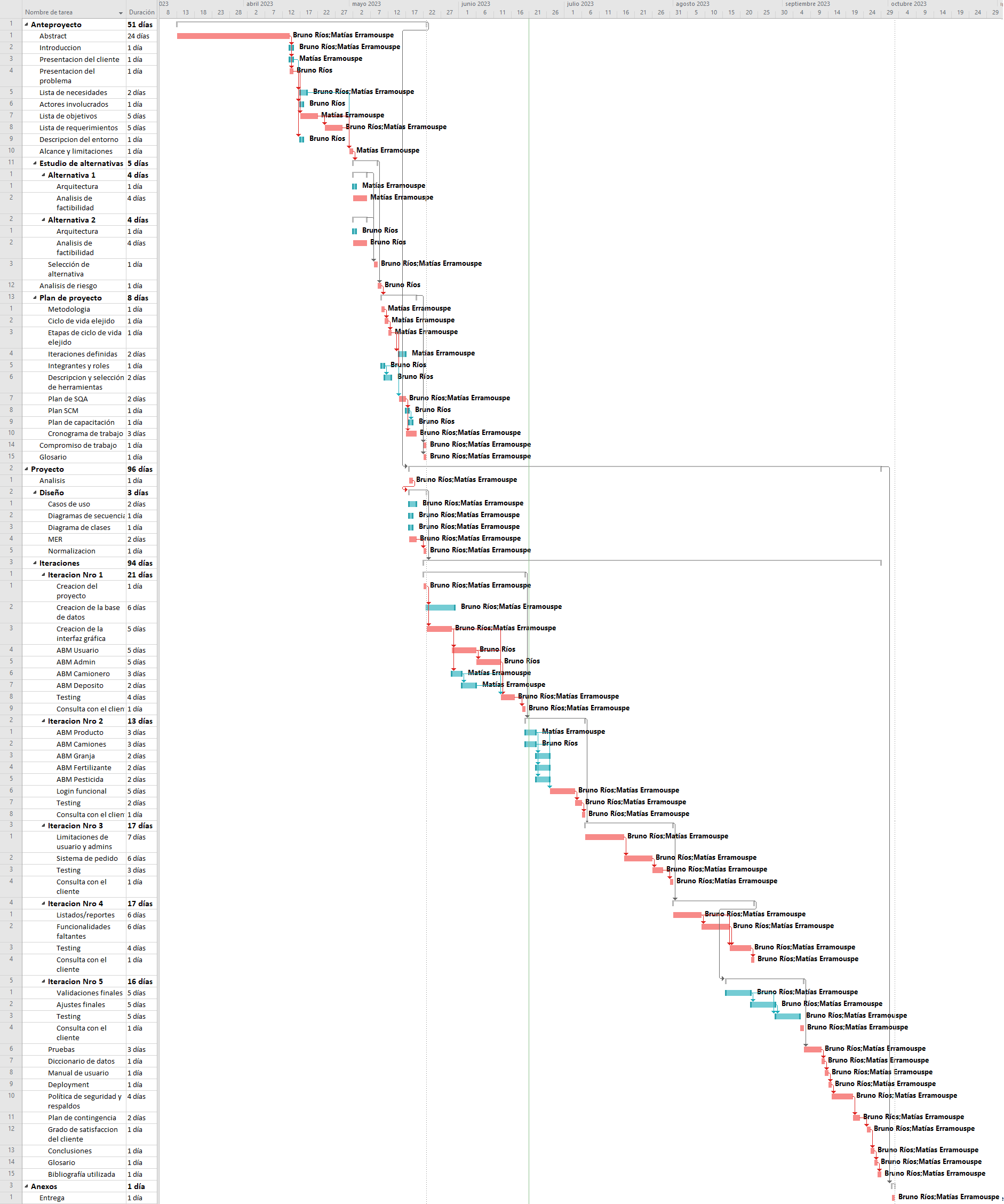


Diagrama de Gantt (ruta crítica)

# Contrato de trabajo

Contrato de Desarrollo de Software

Entre Frutas y Verduras S.A., con domicilio en [dirección], representada por [nombre y cargo del representante], en adelante "el cliente", Matías Erramouspe y Bruno Ríos, con domicilio en [dirección], en adelante "los desarrolladores", se celebra el presente contrato de desarrollo de software, sujeto a las siguientes cláusulas:

Primera: Objeto del contrato

El cliente encarga a los desarrolladores el diseño, estructuración y codificación de un software a medida, con las características y finalidades que se detallan en este documento El software se utilizará para la gestión de pedidos e inventario de la empresa Frutas y Verduras S.A., así como para obtener el título de analista programador por parte de los desarrolladores.

Segunda: Plazo de ejecución

Los desarrolladores se comprometen a entregar el software al cliente en un plazo máximo de cinco meses, contados desde el 23/5/2023 hasta el 23/10/2023. El software se entregará en un soporte digital adecuado, junto con la documentación técnica y el código fuente. El cliente podrá solicitar modificaciones o correcciones al software dentro de los quince días siguientes a la entrega, siempre que no alteren sustancialmente el objeto del contrato.

Tercera: Precio y forma de pago

El precio total del desarrollo del software es de [cantidad en números y letras], que se abonará de la siguiente forma: un 30% al inicio del contrato, un 40% al finalizar el desarrollo y un 30% al entregar el software y la documentación correspondiente. Los pagos se realizarán mediante transferencia bancaria a la cuenta indicada por los desarrolladores.

Cuarta: Propiedad intelectual

Los desarrolladores ceden al cliente todos los derechos de propiedad intelectual sobre el software desarrollado, incluyendo los derechos de reproducción, distribución, comunicación pública y transformación. El cliente podrá utilizar el software con fines comerciales o no comerciales, sin limitación alguna. Los desarrolladores garantizan que el software es original y no infringe los derechos de terceros.

Quinta: Confidencialidad

Los desarrolladores se obligan a mantener la confidencialidad sobre toda la información que reciban del cliente o que generen durante el desarrollo del software, así como a no divulgarla ni utilizarla para fines distintos al objeto de este contrato. Esta obligación subsistirá aun después de finalizado el contrato.

Sexta: Responsabilidad

Los desarrolladores se responsabilizan por el correcto funcionamiento del software y por la solución de cualquier error o defecto que pueda presentar. Asimismo, se comprometen a prestar al cliente el soporte técnico necesario durante un año desde la entrega del software. El cliente se responsabiliza por el uso adecuado del software y por cumplir con las normas legales aplicables.

Séptima: Resolución anticipada

Cualquiera de las partes podrá resolver anticipadamente este contrato en caso de incumplimiento grave o reiterado de las obligaciones asumidas por la otra parte. En ese caso, la parte incumplidora deberá indemnizar a la otra por los daños y perjuicios causados.

Octava: Legislación y jurisdicción aplicable

Este contrato se regirá e interpretará de acuerdo con las leyes de Uruguay. Cualquier controversia o reclamación derivada de este contrato o relacionada con él se someterá a la jurisdicción de los tribunales competentes de Uruguay, Montevideo.

En prueba de conformidad, las partes firman este contrato por duplicado, en el lugar y fecha indicados al inicio.

Firma del cliente \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Firma de los desarrolladores \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Glosario

* Encriptar: Ocultar datos mediante una clave para que no puedan ser interpretados por los que no la tienen.
* Login: En el ámbito de seguridad informática, log in o log on es el proceso que controla el acceso individual a un sistema informático mediante la identificación del usuario utilizando credenciales provistas por el usuario.​
* Backend: Es la parte del desarrollo web que se encarga de que toda la lógica de una página web funcione.
* Frontend: Consiste en la conversión de datos en una interfaz gráfica para que el usuario pueda ver e interactuar con la información de forma digital.
* Hostear: Alojar en tu canal el de otra persona durante un tiempo limitado.
* Open-source: El software de código abierto es el software cuyo código fuente y otros derechos que normalmente son exclusivos para quienes poseen los derechos de autor, son publicados bajo una licencia de código abierto o forman parte del dominio público.
* Testeo: Es una actividad más en el proceso de control de calidad.
* Framework: Un entorno de trabajo​ o marco de trabajo​ es un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar.
* Release: Actualizaciones de un programa la cual soluciona errores y agregan nuevas funcionalidades.
* ABM: Es un término muy utilizado en el mundo informático para hacer referencia a formularios que permite dar de alta, baja y modificar registros en una tabla de una base de datos.
* Responsive: Una página web responsiva o responsive es una técnica del diseño web que permite a los sitios web adaptarse a diferentes pantallas reorganizando la distribución de texto e imágenes.

# Proyecto

# Análisis

Para poder identificar las necesidades de la empresa y poder cumplir con los requerimientos, se llevarán a cabo reuniones con todos los involucrados en el proceso de gestión de pedidos, tanto camioneros, como gestores de stock, encargados de envíos, entre otros.

El programa permitirá el alta, baja y modificaciones de camioneros, camiones, pedidos, productos, admins, granjas, fertilizantes, pesticidas, depósitos y producen. Los clientes se podrán registrar ellos mismos y un admin podrá eliminar estos clientes, los clientes pueden realizar pedidos atreves del catálogo online y los admins deberán confirmar el pedido y asignarle un camión.

Para poder acceder al sistema hay que iniciar sesión, el sistema tiene en consideración si es un admin o un cliente, dentro de los admins hay diferentes tipos de estos, cada uno cumple una función diferente y puede acceder a diferentes partes del sistema, el único administrador que puede acceder al programa en su totalidad es el administrador global. Los clientes podrán cambiar la dirección con la cual se registraron y tanto estos como los admins podrán cambiar su contraseña en caso de olvidarse de esta.

A continuación, se detallarán el proceso por el cual se basa todo el sistema de pedidos actualmente en la empresa, los pedidos son tomados a mano en una planilla de una libreta por los encargados de envíos, estos pedidos son tomados vía email o vía telefónica. Además, las características de cada producto son anotadas de la misma manera que los pedidos. A continuación, se muestra las planillas con la que se guardan los pedidos en la actualidad.

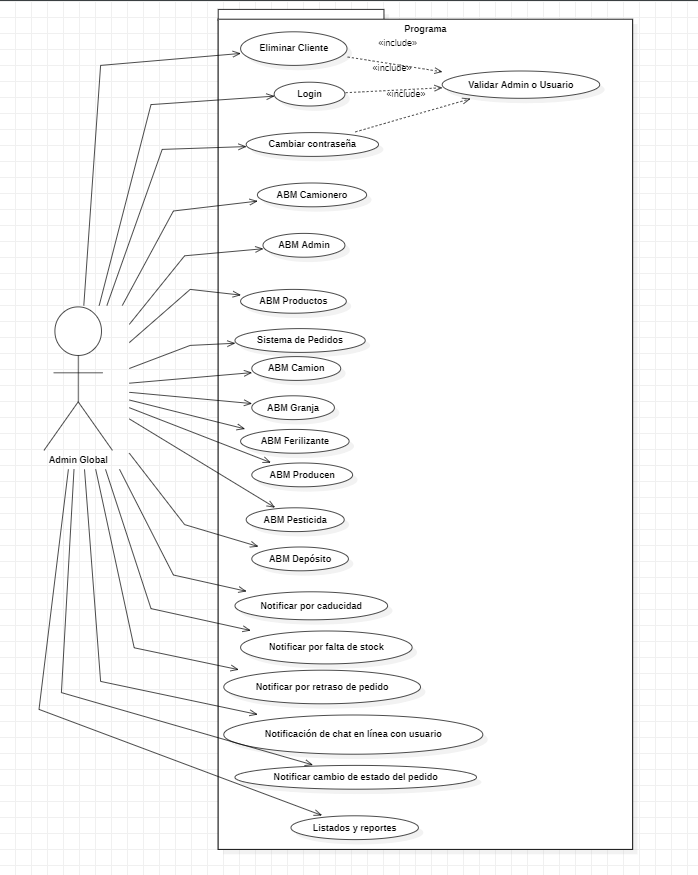
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Pedido | Producto | Cantidad | Costo | Deposito | Cliente | Camión | Camionero | Estado |
| 1 | Tomates | 5 kg | X | 1 | Jose Manuel | 13 | Juan carlos | Asignado |
| 1 | Papas | 5kg | x | 1 | Jose Manuel | 13 | Juan carlos | Asignado |
| 2 | Pepinos | 2kg | x | 2 | Gabriel reyes |  |  | Confirmado |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Producto | Nombre | Tipo | Stock | Precio (KG) | Granja | Pesticida | Fertilizante |
| 1 | Tomate | Hortaliza | 20 kg | 70 | x | x | x |
| 2 | Pepinos | Verduras | 30 kg | 75 | x | x | x |

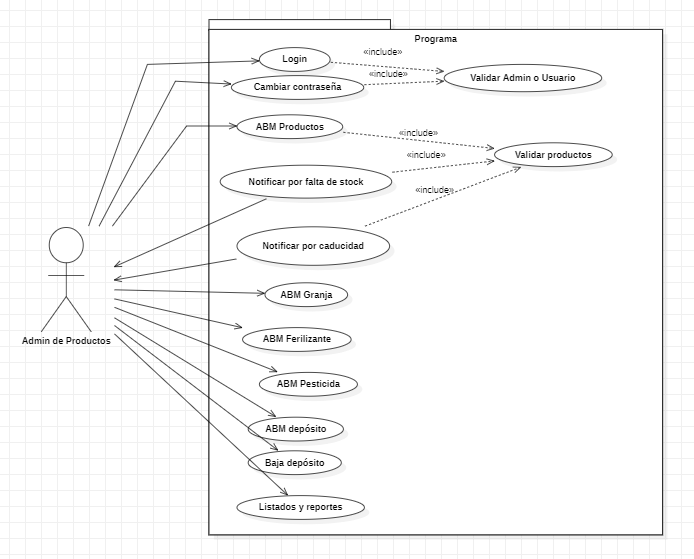
Se diseño una interfaz web para poder guardar estos datos y permitir a los usuarios realizar pedidos de manera más sencilla. Para realizar esto se utilizará ASP.NET con C# con CSS y React para el diseño web, y para guardar todos los datos se utilizará Microsoft SQL server 2018.

# Diseño

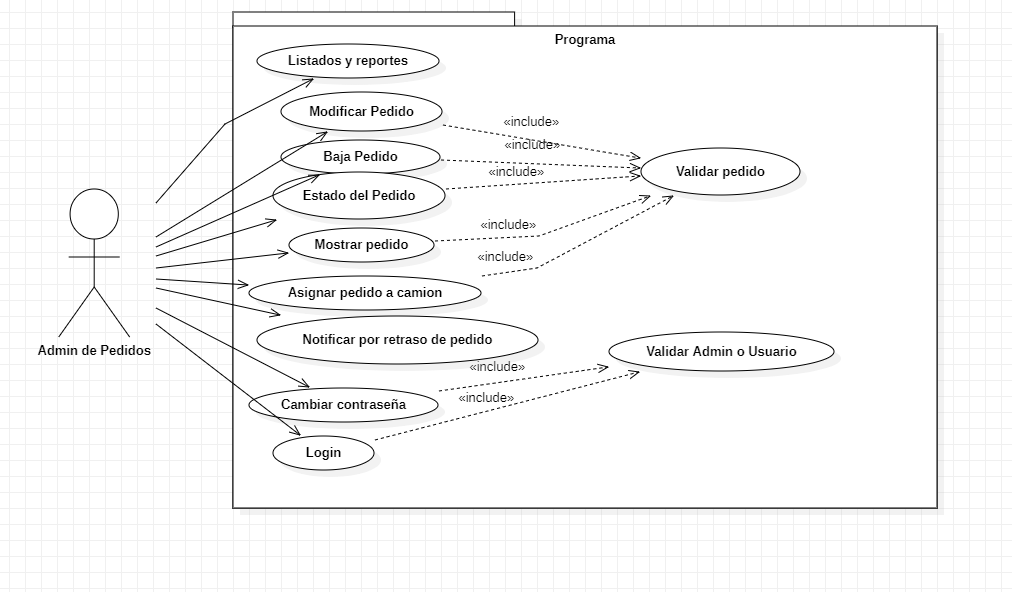
Caso de uso administrador global



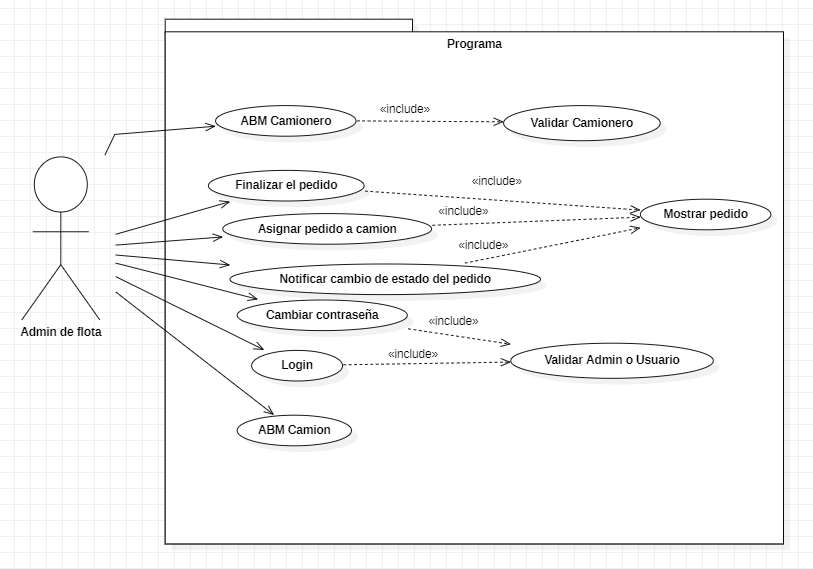
Caso de uso administrador de productos



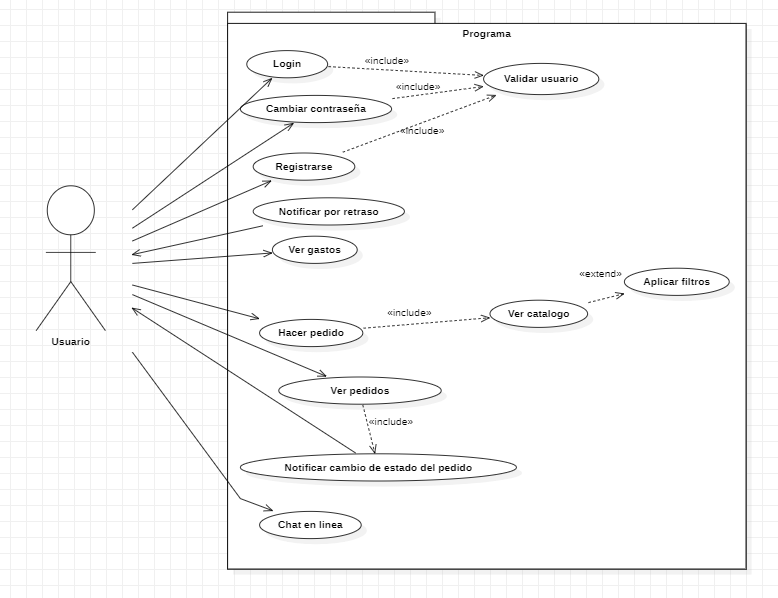
Caso de uso de administrador de pedidos



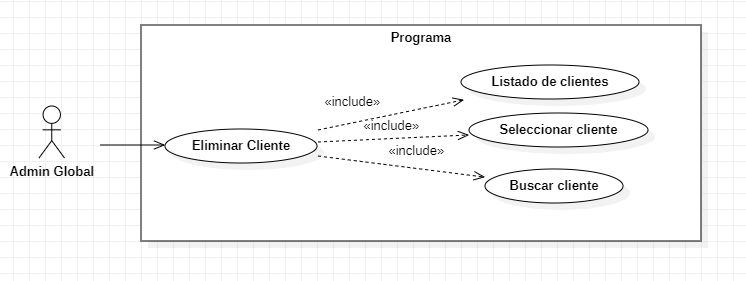
Caso de uso administrador de flota



Caso de uso Clientes



**Eliminar Cliente**



**Identificador:** CU 001

**Nombre del CU:** Eliminar Cliente

**Actores:** Admin global

**Descripción:** Permite eliminar un cliente del sistema.

**Referencia a Requerimientos Funcionales: 12**

**Precondición:** El admin global debe haber iniciado sesión y el cliente a eliminar debe estar registrado.

**Desencadenante:** El admin global decide eliminar a un cliente.

**Curso normal:**

1. El admin global ingresa al listado con todos los clientes registrados.
2. El admin global selecciona al cliente a eliminar y selecciona la opción que dice “Eliminar”.
3. El sistema solicita una confirmación para proceder a dar de baja al cliente.
4. El admin global confirma la baja.
5. El sistema valida que el cliente no tenga pedidos pendientes.
6. Si el cliente no tiene pedidos pendientes lo da de baja.
7. Fin del caso de uso.

**Curso alternativo:**

2.1. El cliente no existe por lo que no se pudo eliminar.

4.1. El admin global cancela la baja del cliente y el caso de uso queda sin efecto.

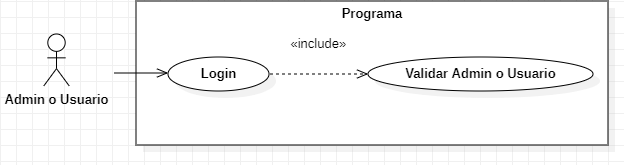
5.1. El cliente tiene pedidos pendientes por lo que el caso de uso queda sin efecto.

**Postcondición:** Se elimina un cliente del sistema.

**Validaciones:** El sistema verifica si el cliente tiene algún pedido pendiente o y si existe antes de eliminarlo.

**Frecuencia de uso:** Baja.

**Login**



**Identificador** CU 002

**Nombre del CU:** Login

**Actores:** Todos los usuarios (incluyendo admins)

**Descripción:** El usuario se identifica con su nombre de usuario y contraseña, accediendo a las funcionabilidades del sistema correspondientes.

**Referencia a Requerimientos Funcionales:** 1, 3

**Precondición:** El usuario debe haberse registrado en el sistema y no debe estar con sesión iniciada.

**Desencadenante:** El usuario decide ingresar al sistema.

**Curso normal:**

1. El usuario accede a la sección de Login, donde debe ingresar su usuario y contraseña.
2. El usuario ingresa los datos.
3. El sistema valida que los campos no estén vacíos.
4. El sistema valida las credenciales.
5. El sistema envía el usuario a la página principal.
6. Fin del caso de uso.

**Curso alternativo:**

3.1. El sistema detecta un campo vacío e informa al usuario.

3.1.1. El usuario completa los campos vacíos y continua el flujo normal.

4.1. El sistema detecta que las credenciales no corresponden a ningún usuario existente en el sistema y no permite el ingreso del usuario al sistema.

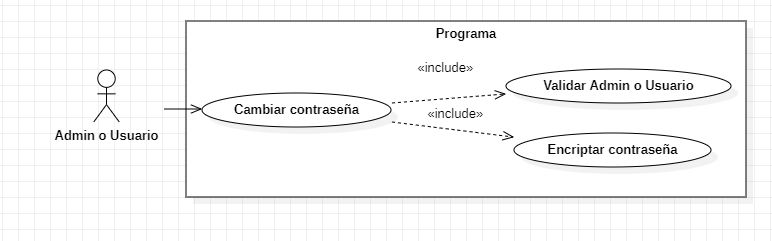
4.1.1. El usuario ingresa los datos de manera correcta y continua el flujo normal.

**Postcondición:** El usuario queda logueado en el sistema y puede acceder a las funcionabilidades correspondientes.

**Validaciones:** El sistema valida que el usuario exista en el sistema, que las credenciales coincidan con los registrados en la base de datos y que los campos no estén vacíos.

**Frecuencia de uso:** Alta.

**Cambiar contraseña**



**Identificador:** CU 003

**Nombre del CU:** Cambiar contraseña

**Actores:** Todos los usuarios (incluyendo admins)

**Descripción:** Permite al usuario cambiar la contraseña de su cuenta.

**Referencia a Requerimientos Funcionales:** 2, 3

**Precondición:** El usuario debe estar con la sesión iniciada.

**Desencadenante:** El usuario decide cambiar su contraseña.

**Curso normal:**

1. El usuario ingresa a la función que dice “Cambiar contraseña”.
2. El sistema redirige al usuario a una página donde le pide su contraseña actual y su contraseña nueva.
3. El usuario ingresa la información requerida.
4. El sistema valida que los campos no estén vacíos.
5. El sistema valida que la contraseña actual sea correcta.
6. El sistema valida la nueva contraseña.
7. El sistema actualiza la nueva contraseña encriptada en la base de datos.
8. El sistema avisa al usuario que la contraseña fue cambiada.
9. Fin del caso de uso.

**Curso alternativo:**

4.1. El sistema detecta un campo vacío e informa al usuario.

4.1.1. El usuario completa el campo vacío y continua el flujo normal.

5.1. El sistema detecta que la contraseña actual es incorrecta e informa al usuario.

5.1.1 El usuario escribe la contraseña correctamente y continua el flujo normal.

6.1 El sistema detecta que la nueva contraseña es igual que la actual e informa al usuario.

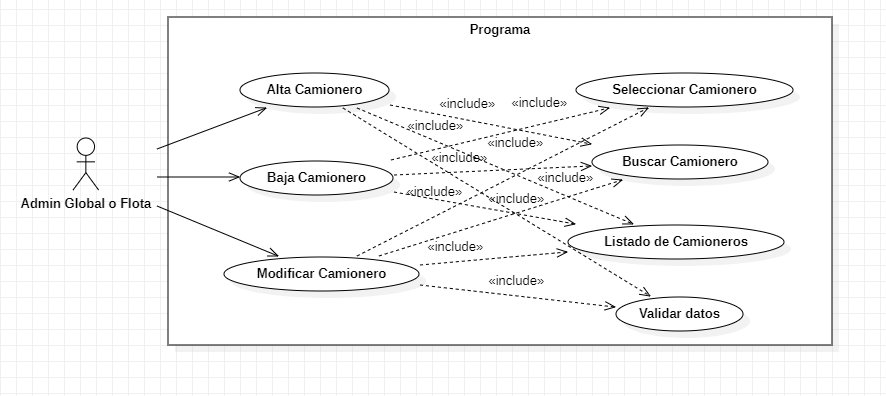
6.1.1 El usuario cambia la contraseña nueva y continua el flujo normal.

**Postcondición:** La contraseña del usuario se actualiza en la base de datos.

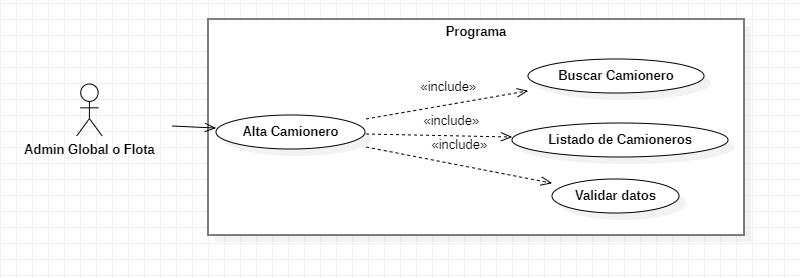
**Validaciones:** El sistema verifica que los campos no estén vacíos, que la contraseña actual sea correcta y que la nueva contraseña no sea igual a la actual.

**Frecuencia de uso:** Baja.

**ABM de Camioneros**



**Alta Camionero**



**Identificador:** CU 004

**Nombre del CU:** Alta Camionero

**Actores:** Admin global/Admin de flota

**Descripción:** Permite dar de alta un camionero.

**Referencia a Requerimientos Funcionales:** 14

**Precondición:** El admin debe estar con la sesión iniciada.

**Desencadenante:** El admin decide agregar un camionero.

**Curso normal:**

1. El admin ingresa al listado de camioneros.
2. El sistema redirige al admin al listado de camioneros
3. El admin ingresa a “Agregar un camionero”
4. El sistema redirige al admin a un formulario para agregar un camionero.
5. El admin ingresa los datos.
6. El sistema verifica que los campos no estén vacíos.
7. El sistema verifica que el nuevo camionero no exista en la base de datos.
8. El sistema ingresa el nuevo camionero a la base de datos.
9. El sistema redirige a la página de gestión de camioneros y se actualiza el listado.
10. Fin del caso de uso.

**Curso alternativo:**

6.1. El sistema detecta un campo vacío e informa al admin.

6.1.1. El admin completa el campo vacío y continua con el flujo normal.

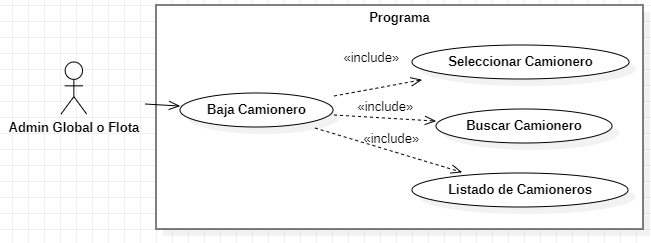
7.1. El sistema detecta que el nuevo camionero ya existe en la base de datos y avisa a través de un mensaje al admin.

**Postcondición:**  Se agrega un nuevo camionero al sistema.

**Validaciones:** El sistema verifica que los campos no estén vacíos y que el camionero no exista en la base de datos.

**Frecuencia de uso:** Media.

**Baja Camionero**



**Identificador:** CU 005

**Nombre del CU:** Baja Camionero

**Actores:** Admin global/Admin de flota

**Descripción:** Permite eliminar un camionero del sistema.

**Referencia a Requerimientos Funcionales:** 14

**Precondición:** El admin debe estar con la sesión iniciada y debe existir algún camionero en el sistema.

**Desencadenante:** El admin decide eliminar un camionero.

**Curso normal:**

1. El admin ingresa al listado de camioneros.
2. El sistema redirige al admin al listado de camioneros.
3. El admin selecciona un camionero para eliminar.
4. El sistema solicita una confirmación de baja.
5. El admin confirma la baja.
6. El sistema verifica que el camionero exista en la base de datos.
7. El software verifica que el camionero no tenga un viaje pendiente o esté realizando un viaje
8. Si existe y no tiene un viaje pendiente o no está realizando un pedido, el camionero es eliminado.
9. El sistema recarga la página actualizando el listado.
10. Fin del caso de uso.

**Curso alternativo:**

5.1. El admin cancela la baja, dejando el caso de uso sin efecto.

6.1. El sistema detecta que el camionero no existe e informa al admin.

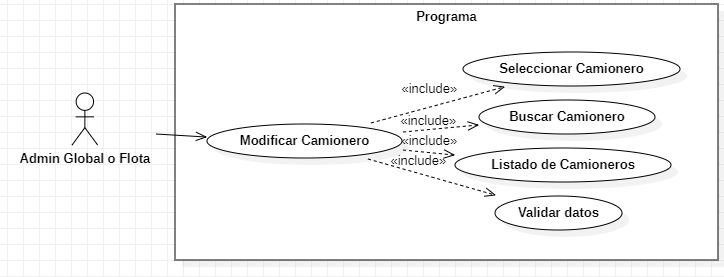
7.1. El sistema detecta que el camionero tiene un pedido pendiente o está en un viaje, por lo que el sistema avisa que se deberá esperar a que termine los pedidos o se elimine estos para continuar con la eliminación del camionero.

**Postcondición:** Se elimina un camionero del sistema.

**Validaciones:** El sistema verifica que el camionero exista en la base de datos y que no tenga pedidos pendientes o esté en un viaje para eliminarlo.

**Frecuencia de uso:** Baja.

**Modificar Camionero:**



**Identificador:** CU 006

**Nombre del CU:** Modificar Camionero

**Actores:** Admin global/Admin de flota

**Descripción:** Permite modificar un camionero del sistema.

**Referencia a Requerimientos Funcionales:** 14

**Precondición:** El admin debe estar con la sesión iniciada y debe existir algún camionero en el sistema.

**Desencadenante:** El admin decide modificar un camionero.

**Curso normal:**

1. El admin ingresa al listado de camioneros.
2. El sistema redirige al admin al listado de camioneros.
3. El admin selecciona un camionero para modificar.
4. El sistema redirige al admin a un formulario con los datos del camionero.
5. El admin modifica los datos y finaliza el proceso.
6. El sistema solicita una confirmación de modificación.
7. El admin confirma la modificación.
8. El sistema verifica que no haya algún campo vacío.
9. El sistema verifica que el camionero exista en la base de datos.
10. El sistema actualiza el camionero en la base de datos.
11. El sistema redirige al admin al listado de camioneros actualizándolo.
12. Fin del caso de uso.

**Curso alternativo:**

7.1. El admin cancela la modificación, dejando el caso de uso sin efecto.

8.1. El sistema detecta un campo vacío e informa al admin.

8.1.1. El admin completa el campo vacío y continua con el flujo norma

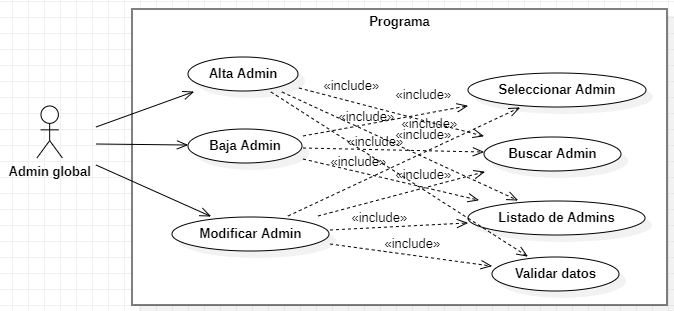
9.1. El sistema detecta que el camionero no existe e informa al admin.

**Postcondición:** Se modifica un camionero en el sistema.

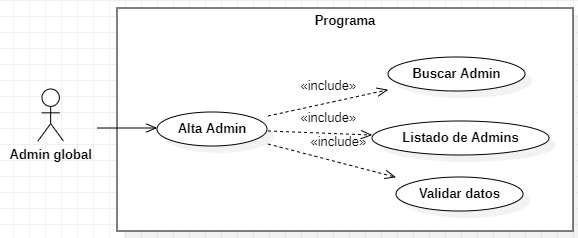
**Validaciones:** El sistema verifica que los campos no estén vacíos y que el camionero exista en la base de datos.

**Frecuencia de uso:** Baja.

**ABM de Admins**



**Alta Admin**



**Identificador:** CU 007

**Nombre del CU:** Alta Admin

**Actores:** Admin global

**Descripción:** Permite dar de alta un admin.

**Referencia a Requerimientos Funcionales: 13**

**Precondición:** El admin global debe estar con la sesión iniciada.

**Desencadenante:** El admin global decide agregar un admin.

**Curso normal:**

1. El admin ingresa al listado de Admins.
2. El sistema redirige al admin al listado de Admins.
3. El admin global ingresa a la función que dice “Agregar un admin”.
4. El sistema redirige al admin global a un formulario para agregar un admin.
5. El admin global completa los datos requeridos y selecciona el tipo de admin
6. El sistema verifica que los datos no estén vacíos.
7. El sistema verifica que el nuevo admin no exista en la base de datos.
8. El sistema ingresa al nuevo admin a la base de datos.
9. El sistema redirige a la página de gestión de admins al admin global y se actualiza el listado.
10. Fin del caso de uso.

**Curso alternativo:**

6.1. El sistema detecta un dato vacío e informa al admin global.

6.1.1. El admin global completa el dato faltante y continua con el flujo normal.

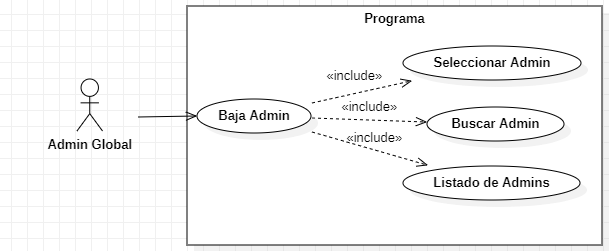
7.1. El sistema detecta que el nuevo admin ya existe en la base de datos e informa al admin global.

**Postcondición:**  Se agrega un nuevo admin al sistema.

**Validaciones:** El sistema verifica que los datos no estén vacíos y que el admin no exista en la base de datos.

**Frecuencia de uso:** Baja.

**Baja Admin**



**Identificador:** CU 008

**Nombre del CU:** Baja Admin

**Actores:** Admin global

**Descripción:** Permite eliminar un admin del sistema.

**Referencia a Requerimientos Funcionales: 13**

**Precondición:** El admin global debe estar con la sesión iniciada y debe existir algún admin en el sistema.

**Desencadenante:** El admin global decide eliminar un admin.

**Curso normal:**

1. El admin ingresa al listado de Admins.
2. El sistema redirige al admin al listado de Admins.
3. El admin global selecciona un admin para eliminar.
4. El sistema solicita una confirmación de baja.
5. El admin global confirma la baja.
6. El sistema verifica que el admin exista en la base de datos.
7. Si existe el admin es eliminado.
8. El sistema recarga la página actualizando el listado.
9. Fin del caso de uso.

**Curso alternativo:**

5.1. El admin global cancela la baja, dejando el caso de uso sin efecto.

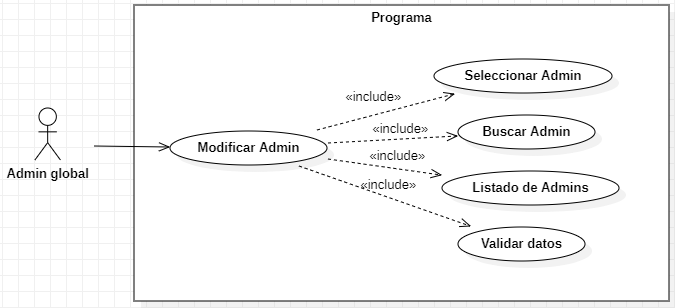
6.1. El sistema detecta que el admin no existe e informa al admin global.

**Postcondición:** Se elimina un admin del sistema.

**Validaciones:** El sistema verifica que el admin exista en la base de datos para eliminarlo.

**Frecuencia de uso:** Baja.

**Modificar Admin:**

**Identificador:** CU 009

**Nombre del CU:** Modificar Admin

**Actores:** Admin global

**Descripción:** Permite modificar un admin del sistema.

**Referencia a Requerimientos Funcionales: 13**

**Precondición:** El admin global debe estar con la sesión iniciada y debe existir algún admin en el sistema.

**Desencadenante:** El admin global decide modificar un admin.

**Curso normal:**

1. El admin ingresa al listado de Admins.
2. El sistema redirige al admin global al listado de los Admins.
3. El admin global selecciona un admin para modificar.
4. El sistema redirige al admin global a un formulario de admins, dejándolo modificar.
5. El admin global modifica los datos.
6. El sistema solicita una confirmación de modificación.
7. El admin global confirma la modificación.
8. El sistema verifica que no hay datos vacíos.
9. El sistema verifica que el admin exista en la base de datos.
10. El sistema actualiza el admin en la base de datos.
11. El sistema redirige al admin global al listado de admins actualizándolo.
12. Fin del caso de uso.

**Curso alternativo:**

7.1 El admin global cancela la modificación, dejando el caso de uso sin efecto.

8.1 El sistema verifica que hay datos vacíos e informa al admin global.

8.1.1 El admin global completa los datos vacíos y continua con el flujo normal.

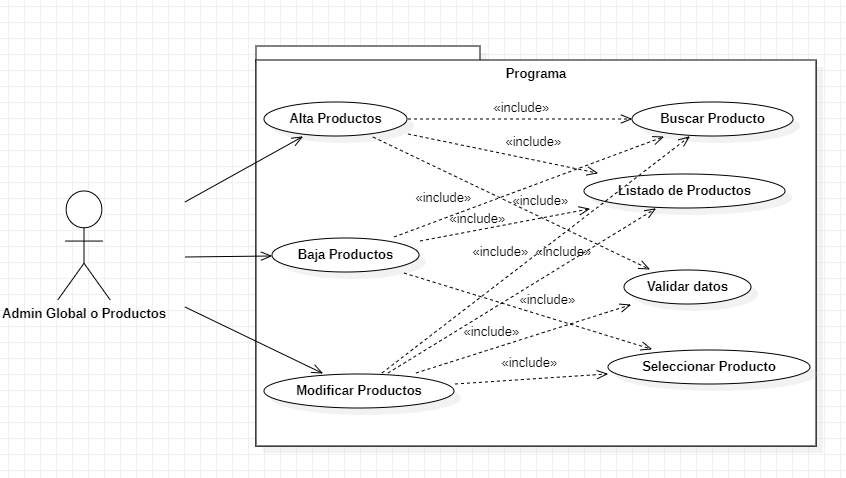
9.1 El sistema detecta que el admin no existe e informa al admin global.

**Postcondición:** Se modifica un admin en el sistema.

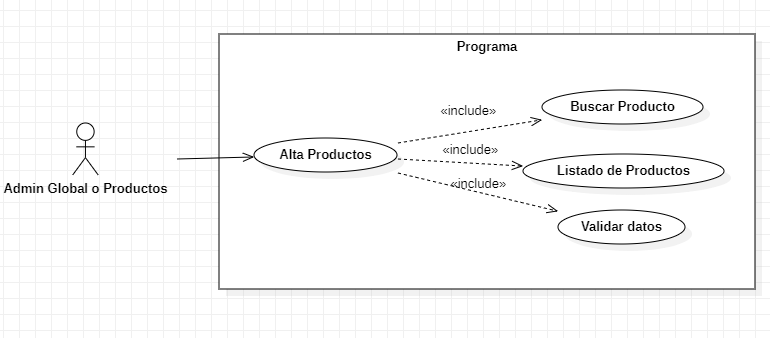
**Validaciones:** El sistema verifica que no hay datos vacíos y que el admin exista en la base de datos.

**Frecuencia de uso:** Baja.

**ABM de Productos**



**Alta Productos**



**Identificador:** CU 010

**Nombre del CU:** Alta Producto

**Actores:** Admin global, Admin de productos

**Descripción:** Permite dar de alta un producto.

**Referencia a Requerimientos Funcionales: 17**

**Precondición:** El admin debe estar con la sesión iniciada.

**Desencadenante:** El admin decide agregar un producto.

**Curso normal:**

1. El admin ingresa al listado de productos.
2. El sistema redirige al admin al listado de productos.
3. El admin ingresa a la función que dice “Agregar producto”.
4. El sistema redirige al admin a un formulario para agregar un producto.
5. El admin ingresa los datos.
6. El sistema verifica que los campos no estén vacíos.
7. El sistema verifica que el nuevo producto no exista en la base de datos.
8. El sistema ingresa el nuevo producto a la base de datos.
9. El sistema redirige a la página de gestión de productos al admin y se actualiza el listado.
10. Fin del caso de uso.

**Curso alternativo:**

6.1. El sistema detecta un campo vacío e informa al admin.

6.1.1. El admin completa el campo vacío y continua con el flujo normal.

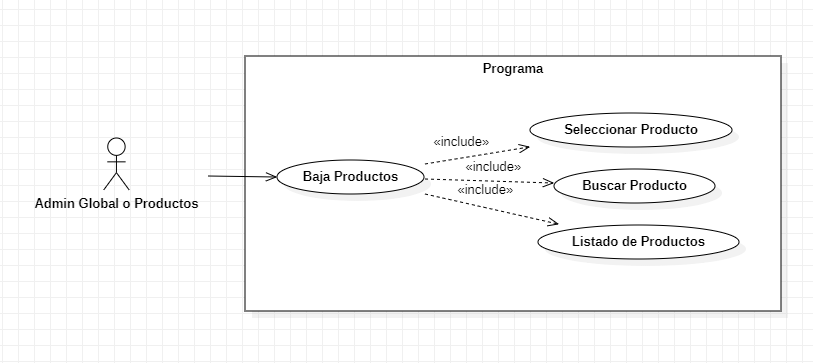
7.1. El sistema detecta que el nuevo producto ya existe en la base de datos e informa al admin.

**Postcondición:**  Se agrega un nuevo producto al sistema.

**Validaciones:** El sistema verifica que los campos no estén vacíos y que el producto no exista en la base de datos.

**Frecuencia de uso:** Alta.

**Baja Productos**



**Identificador:** CU 011

**Nombre del CU:** Baja Productos

**Actores:** Admin global, Admin de productos

**Descripción:** Permite eliminar un producto del sistema.

**Referencia a Requerimientos Funcionales: 17**

**Precondición:** El admin debe estar con la sesión iniciada y debe existir algún producto en el sistema.

**Desencadenante:** El admin decide eliminar un producto.

**Curso normal:**

1. El admin ingresa al listado de productos.
2. El sistema redirige al admin al listado de productos.
3. El admin selecciona un producto para eliminar.
4. El sistema solicita una confirmación de baja.
5. El admin confirma la baja.
6. El sistema verifica que el producto exista en la base de datos.
7. El sistema verifica que el producto no exista en ningún pedido,
8. Si existe el producto y no se encuentra en ningún pedido, el producto es eliminado.
9. El sistema recarga la página actualizando el listado.
10. Fin del caso de uso.

**Curso alternativo:**

5.1. El admin cancela la baja, dejando el caso de uso sin efecto.

6.1. El sistema detecta que el producto no existe e informa al admin.

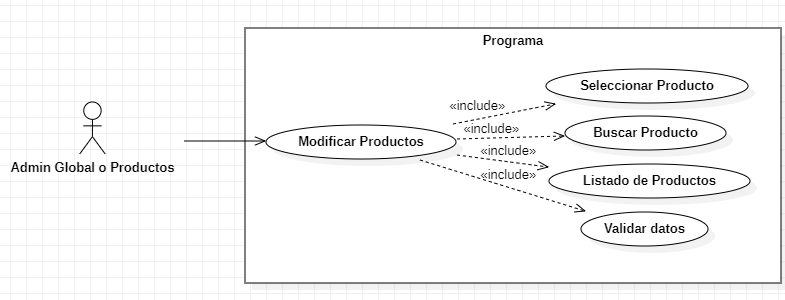
7.1. El sistema detecta que el producto se encuentra dentro de un pedido, para seguir con el curso normal hay que eliminar el producto del pedido.

**Postcondición:** Se elimina un producto del sistema.

**Validaciones:** El sistema verifica que el producto exista en la base de datos y que no exista dentro de un pedido para eliminarlo.

**Frecuencia de uso:** Baja.

**Modificar Productos:**



**Identificador:** CU 012

**Nombre del CU:** Modificar Productos

**Actores:** Admin global, Admin de productos

**Descripción:** Permite modificar un producto del sistema.

**Referencia a Requerimientos Funcionales: 17**

**Precondición:** El admin debe estar con la sesión iniciada y debe existir algún producto en el sistema.

**Desencadenante:** El admin decide modificar un producto.

**Curso normal:**

1. El admin ingresa al listado de productos.
2. El sistema redirige al admin al listado de los productos.
3. El admin selecciona un producto para modificar.
4. El sistema redirige al admin a un formulario con los datos del producto dejándolos modificar.
5. El admin modifica los datos y finaliza el proceso.
6. El sistema solicita una confirmación de modificación.
7. El admin confirma la modificación.
8. El sistema verifica que no haya algún campo vacío.
9. El sistema verifica que el producto exista en la base de datos.
10. El sistema actualiza el producto en la base de datos.
11. El sistema redirige al admin al listado de productos actualizándolo.
12. Fin del caso de uso.

**Curso alternativo:**

7.1 El admin cancela la modificación, dejando el caso de uso sin efecto.

8.1 El sistema detecta un campo vacío e informa al admin.

8.1.1 El admin completa el campo vacío y continua con el flujo normal.

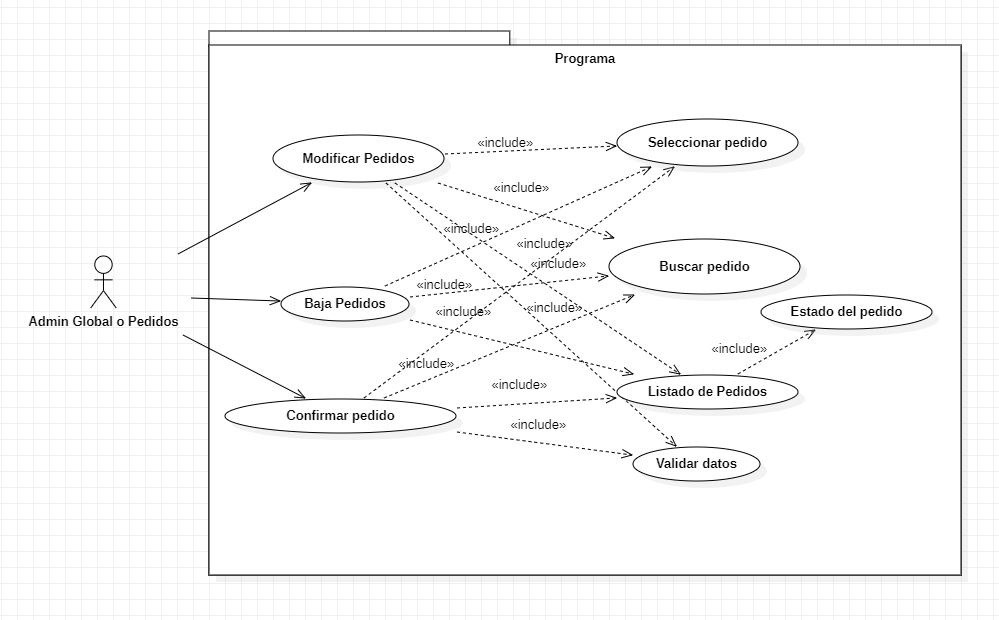
9.1 El sistema detecta que el producto no existe e informa al admin.

**Postcondición:** Se modifica un producto en el sistema.

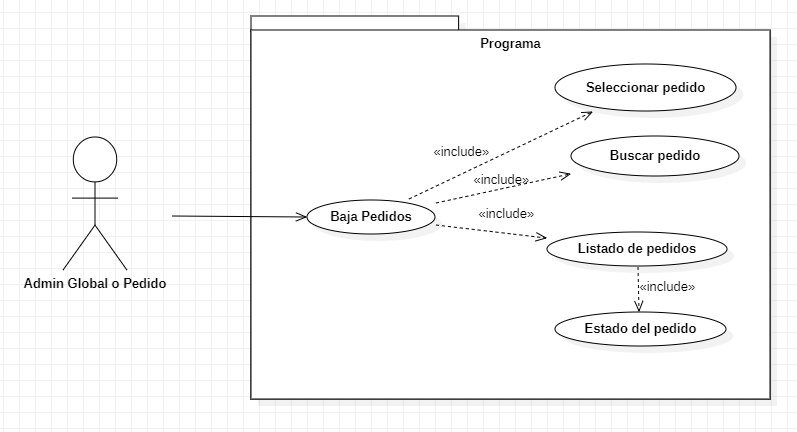
**Validaciones:** El sistema verifica que los campos no estén vacíos y que el producto exista en la base de datos.

**Frecuencia de uso:** Baja.

**Sistema de pedidos**



**Baja pedidos**



**Identificador:** CU 013

**Nombre del CU:** Baja Pedidos

**Actores:** Admin global, Admin de Pedidos

**Descripción:** Permite eliminar un pedido del sistema.

**Referencia a Requerimientos Funcionales: 18**

**Precondición:** El admin debe estar con la sesión iniciada y debe existir algún pedido en el sistema.

**Desencadenante:** El admin decide eliminar un pedido.

**Curso normal:**

1. El admin ingresa al listado de pedidos.
2. El sistema redirige al admin al listado de pedidos.
3. El admin selecciona un pedido para eliminar.
4. El sistema solicita una confirmación de baja.
5. El admin confirma la baja.
6. El sistema verifica que el pedido exista en la base de datos.
7. El sistema verifica que el pedido no este confirmado o asignado.
8. Si existe el pedido y no se encuentra activo, el pedido es eliminado.
9. El sistema recarga la página actualizando el listado.
10. Fin del caso de uso.

**Curso alternativo:**

5.1. El admin cancela la baja, dejando el caso de uso sin efecto.

6.1. El sistema detecta que el pedido no existe e informa al admin.

7.1. El sistema detecta que el pedido se encuentra activo, por lo que para eliminarlo se debe dar una razón y esta petición de eliminación debe ser autorizada por un admin global. Si la petición de eliminar el pedido es de un admin global puede saltarse la verificación, pero de igual manera el sistema pedirá una razón de la eliminación.

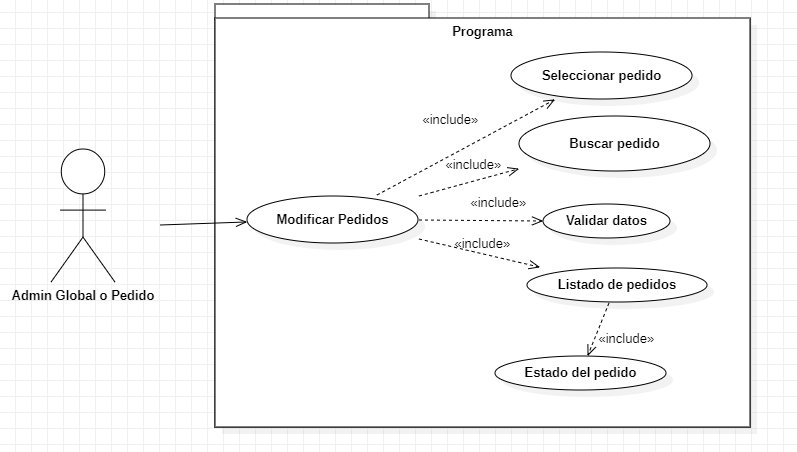
7.1.1. Si se da la razón al sistema y esta es verificada con un admin global, se continua con el flujo normal.

**Postcondición:** Se elimina un pedido del sistema.

**Validaciones:** El sistema verifica que el pedido exista en la base de datos y que el pedido no este activo para eliminarlo.

**Frecuencia de uso:** Baja.

**Modificar pedidos**



**Identificador:** CU 014

**Nombre del CU:** Modificar pedidos

**Actores:** Admin global, Admin de pedidos

**Descripción:** Permite modificar un pedido del sistema.

**Referencia a Requerimientos Funcionales: 18**

**Precondición:** El admin debe estar con la sesión iniciada y debe existir algún pedido en el sistema.

**Desencadenante:** El admin decide modificar un pedido.

**Curso normal:**

1. El admin ingresa al listado de pedidos.
2. El sistema redirige al admin al listado de los pedidos.
3. El admin selecciona un pedido para modificar.
4. El sistema redirige al admin a un formulario con los datos del pedido
5. Antes de modificar, el admin debe dejar la razón por la cual se está modificando el pedido.
6. El admin escribe la razón, modifica los datos y finaliza el proceso.
7. El sistema solicita una confirmación de modificación.
8. El admin confirma la modificación.
9. El sistema verifica que no haya algún campo vacío.
10. El sistema verifica que el pedido exista en la base de datos.
11. El sistema actualiza el pedido en la base de datos.
12. El sistema redirige al admin al listado de productos actualizándolo.
13. Fin del caso de uso.

**Curso alternativo:**

6.1. El sistema detecta que falta la razón e informa al admin

6.1.1 El admin escribe la razón y continúa con el flujo normal.

8.1. El admin cancela la modificación, dejando el caso de uso sin efecto.

9.1. El sistema detecta un campo vacío e informa al admin.

9.1.1. El admin completa el campo vacío y continua con el flujo normal.

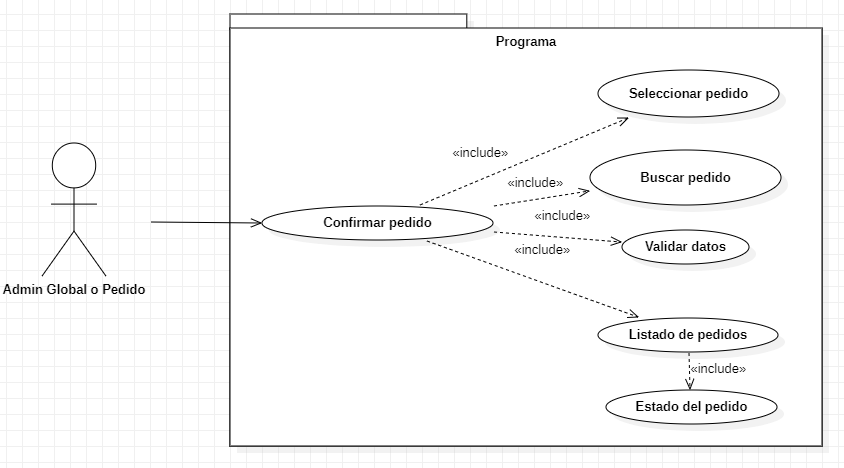
10.1. El sistema detecta que el pedido no existe e informa al admin.

**Postcondición:** Se modifica un pedido en el sistema.

**Validaciones:** El sistema verifica que los campos no estén vacíos y que el pedido exista en la base de datos.

**Frecuencia de uso:** Baja.

**Confirmar pedido**

**Identificador:** CU 015

**Nombre del CU:** Confirmar pedidos

**Actores:** Admin global, Admin de pedidos

**Descripción:** Permite confirmar un pedido del sistema.

**Referencia a Requerimientos Funcionales: 18**

**Precondición:** El admin debe estar con la sesión iniciada y debe existir algún pedido en el sistema.

**Desencadenante:** El admin decide confirmar un pedido.

**Curso normal:**

1. El admin ingresa al listado de pedidos.
2. El sistema redirige al admin al listado de los pedidos.
3. El admin ingresa a la función que dice “Pedidos sin confirmar”.
4. El sistema redirige al admin al listado de los pedidos sin confirmar.
5. El admin selecciona un pedido para confirmar.
6. El sistema redirige al admin a un formulario con los datos del pedido.
7. El admin y el sistema verifica que la cantidad de producto registrada en el sistema sea igual o mayor a la requerida por el pedido.
8. El sistema solicita una confirmación.
9. El admin da la confirmación.
10. El sistema verifica que el pedido exista en la base de datos.
11. El sistema actualiza el pedido en la base de datos.
12. El sistema redirige al admin al listado de “Pedidos sin confirmar” actualizándolo.
13. Fin del caso de uso.

**Curso alternativo:**

7.1. El pedido a confirmar requiere una cantidad mayor comparada con la cantidad de stock actualmente registrada en el sistema, por lo que el admin puede confirmar el pedido con menos cantidad o aplazar hasta que se ingrese más stock al sistema.

7.1.1. El pedido se confirmó a pesar del problema y vuelve al curso normal

9.1. El admin cancela la confirmación, dejando el caso de uso sin efecto.

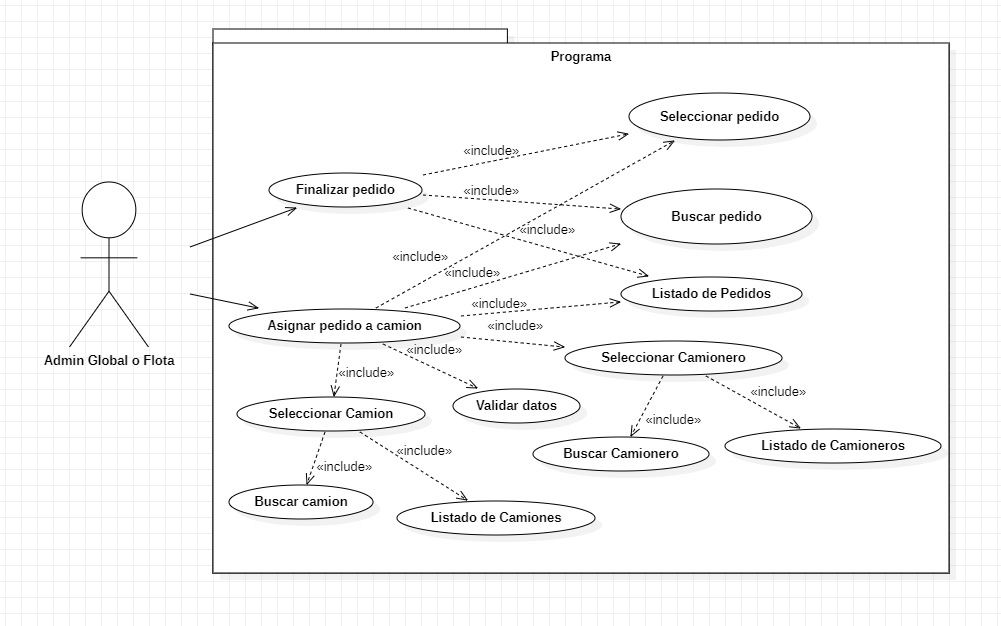
10.1. El sistema detecta que el pedido no existe e informa al admin.

**Postcondición:** Pasa de estar sin confirmar ha confirmado.

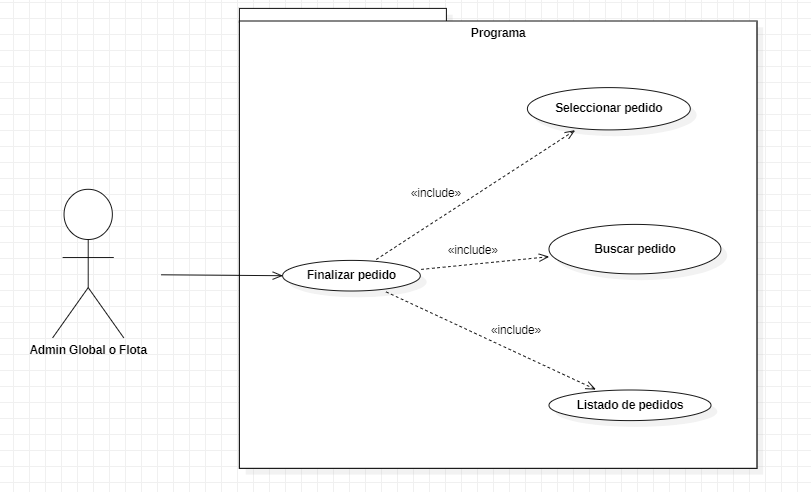
**Validaciones:** El sistema verifica que el pedido exista en la base de datos y que la cantidad de producto registrada en el sistema sea igual o mayor a la requerida por el pedido.

**Frecuencia de uso: Alta**.

**Etapa final del pedido**



**Finalizar pedido**



**Identificador:** CU 016

**Nombre del CU:** Finalizar pedido

**Actores:** Admin global, Admin de flota

**Descripción:** Permite finalizar un pedido del sistema.

**Referencia a Requerimientos Funcionales: 18**

**Precondición:** El admin debe estar con la sesión iniciada, debe existir algún pedido en el sistema, el pedido debe del tipo asignado y el pedido debe ser entregado.

**Desencadenante:** El admin decide finalizar un pedido.

**Curso normal:**

1. El admin ingresa al listado de pedidos.
2. El sistema redirige al admin al listado de los pedidos.
3. El admin ingresa a la función que dice “Finalizar pedido”.
4. El sistema redirige al admin al listado de pedidos de tipo asignado.
5. El admin selecciona un pedido para finalizarlo.
6. El sistema solicita una confirmación.
7. El admin da la confirmación.
8. El sistema verifica que el pedido exista en la base de datos.
9. El sistema verifica que el pedido este confirmado y que tenga un camión asignado.
10. Si existe el pedido y se encuentra confirmado y asignado, el pedido es finalizado.
11. El sistema recarga la página actualizando el listado.
12. Fin del caso de uso.

**Curso alternativo:**

7.1. El admin cancela la finalización, dejando el caso de uso sin efecto.

8.1. El sistema detecta que el pedido no existe e informa al admin.

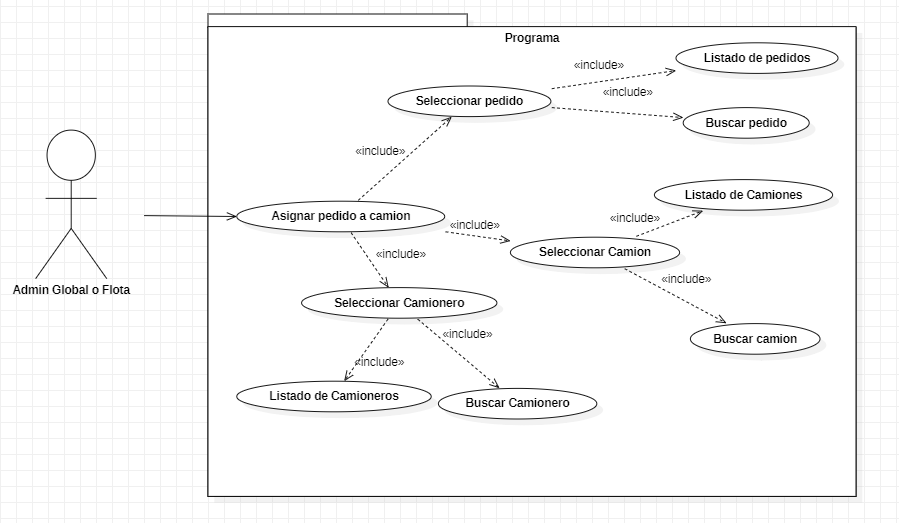
9.1. El sistema detecta que el pedido no se encuentra confirmado o no tiene un camión asignado, para volver al curso normal el pedido debe ser confirmado y debe tener un camión asignado

**Postcondición:** Se marca como entregado un pedido en el sistema.

**Validaciones:** El sistema verifica que el pedido exista en la base de datos, que el pedido este confirmado y que el pedido tenga asignado un camión para finalizarlo.

**Frecuencia de uso: Alta**.

**Asignar pedido a camión**



**Identificador:** CU 017

**Nombre del CU:** Asignar pedido a camión.

**Actores:** Admin global, Admin de Flota.

**Descripción:** Permite asignar un camión a un pedido.

**Referencia a Requerimientos Funcionales: 18**

**Precondición:** El admin debe estar con la sesión iniciada y deben haber registrados camiones, camioneros y pedidos confirmados.

**Desencadenante:** El admin decide asignar un camión a un pedido confirmado.

**Curso normal:**

1. El admin ingresa al listado de pedidos.
2. El sistema redirige al admin al listado de los pedidos.
3. El admin ingresa a la función que dice “Pedidos Confirmados”.
4. El admin selecciona un pedido confirmado.
5. El sistema redirige al admin a un formulario para agregar los datos faltantes.
6. El admin ingresa un camionero y un camión.
7. El sistema verifica que los campos no estén vacíos.
8. El sistema verifica que el pedido, camionero y el camión existan en la base de datos.
9. El sistema verifica que el pedido este confirmado.
10. El sistema ingresa el pedido a pedidos asignados a la base de datos.
11. El sistema redirige a la página de gestión de pedidos al admin y se actualiza el listado.
12. Fin del caso de uso.

**Curso alternativo:**

7.1. El sistema detecta un campo vacío e informa al admin.

7.1.1. El admin completa el campo vacío y continua con el flujo normal.

8.1. El sistema detecta que el producto, camionero o camión no existe en la base de datos e informa al admin.

9.1 El pedido no está confirmado por lo que finaliza el caso de uso.

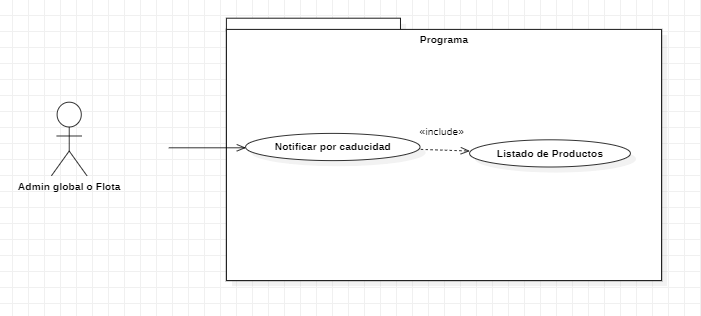
9.1.1 El pedido se confirmó por lo que se continua con el curso normal.

**Postcondición:**  Se cambia el pedido de confirmado a pedido asignado.

**Validaciones:** El sistema verifica que los campos no estén vacíos y que el producto, camionero o camión exista en la base de datos, además el producto debe ser de tipo confirmado.

**Frecuencia de uso:** Alta.

**Notificar por caducidad**



**Identificador:** CU 018

**Nombre del CU:** Notificar por caducidad

**Actores:** Admin global, Admin de flota

**Descripción:** Permite al sistema notificar por productos caducados.

**Referencia a Requerimientos Funcionales: 17**

**Precondición: D**ebe haber productos registrados.

**Desencadenante:** El sistema encuentra un producto caducado.

**Curso normal:**

1. El sistema recorre todos los productos.
2. El sistema encuentra un producto caducado.
3. El sistema notifica a los admins globales y de productos en el menú de inicio de estos.
4. El sistema actualiza la base de datos cambiando el producto como caducado.

**Curso alternativo:**

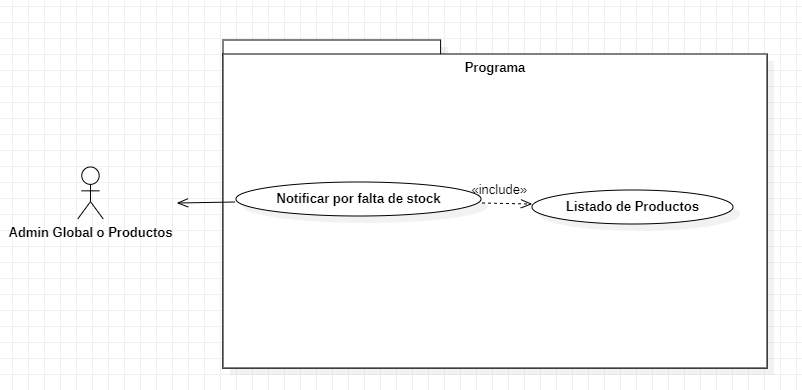
2.1 El sistema no encuentra un producto caducado, por lo tanto, queda anulado el caso de uso

**Postcondición:** El producto dejara de estar disponible para realizar pedidos.

**Validaciones:** El sistema recorre todos los productos y verifica que la fecha actual sea menor que la fecha de caducidad

**Frecuencia de uso:** Media.

**Notificar por falta de stock**



**Identificador:** CU 019

**Nombre del CU:** Notificar por falta de stock

**Actores:** Admin global, Admin de productos

**Descripción:** Permite al sistema notificar por productos sin stock.

**Referencia a Requerimientos Funcionales: 17**

**Precondición:** Debe haber productos registrados.

**Desencadenante:** El sistema encuentra un producto sin stock.

**Curso normal:**

1. El sistema recorre todos los productos.
2. El sistema encuentra un producto sin stock.
3. El sistema notifica a los admins globales y de productos en el menú de inicio de estos.
4. El sistema actualiza la base de datos cambiando el producto como sin stock.

**Curso alternativo:**

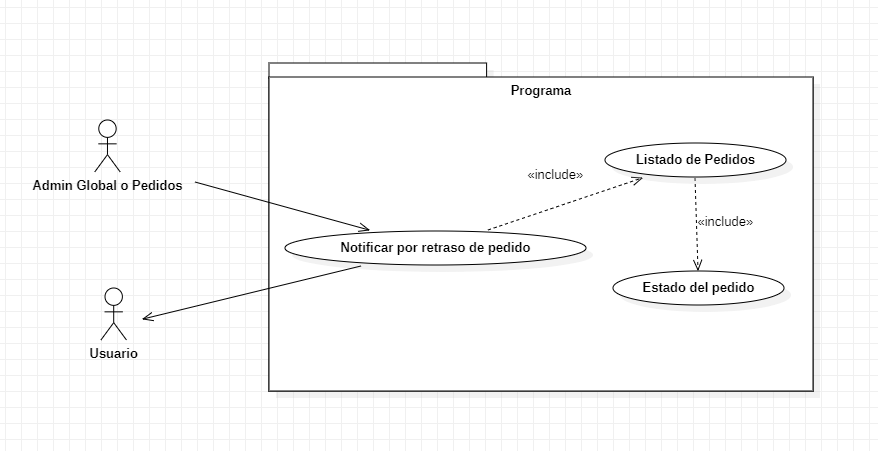
2.1. El sistema no encuentra un producto con falta de stock, dejando el caso de uso sin efecto.

**Postcondición:**  El producto dejara de estar disponible para realizar pedidos.

**Validaciones:** El sistema recorre todos los productos y verifica que el stock del producto sea igual o mayor a 3kg.

**Frecuencia de uso:** Media.

**Notificar por retraso de pedido**



**Identificador:** CU 020

**Nombre del CU:** Notificar por retraso de pedido

**Actores:** Admin global, Admin de pedidos, usuario

**Descripción:** Permite notificar a un cliente sobre el retraso de su paquete.

**Referencia a Requerimientos Funcionales: 19**

**Precondición:** El admin debe estar con la sesión iniciada y debe haber pedidos asignados o confirmado en el sistema.

**Desencadenante:** El admin detecta un retraso en un paquete y desea informarle al cliente al respecto.

**Curso normal:**

1. El admin ingresa al listado de pedidos.
2. El sistema redirige al admin al listado de los pedidos.
3. El admin ingresa a la función que dice “Notificar por retraso de pedido”
4. El sistema redirecciona al admin al listado de pedidos asignados o confirmados.
5. El admin selecciona un pedido para notificar por retraso.
6. El admin agrega la razón del retraso del pedido.
7. El sistema verifica que exista la razón.
8. El sistema solicita una confirmación para proceder a notificar al cliente.
9. El admin confirma la solicitud.
10. El sistema verifica que el pedido existe y que sea del tipo confirmado o asignado.
11. El sistema envía una notificación al cliente sobre el retraso de su pedido.
12. El sistema redirecciona al admin al listado de pedidos.
13. Fin del caso de uso.

**Curso alternativo:**

**7.1 El** admin no agregó la razón del retraso del pedido, por lo tanto, el sistema notifica al admin de la usencia de razón.

7.1.1 El admin agrega la razón por lo que se retoma el curso normal.

8.1. El admin cancela la solicitud de notificar por retraso, dejando el caso de uso sin efecto.

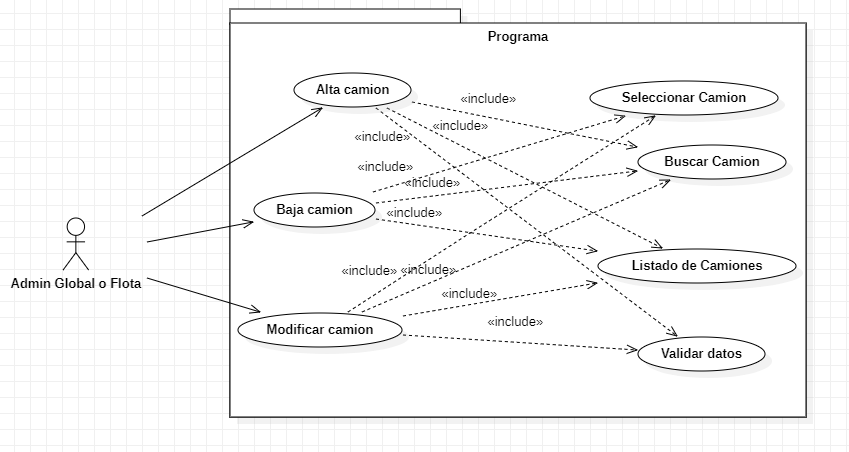
10.1 El sistema detecta que no existe el pedido o el tipo del pedido no es del tipo confirmado o asignado.

**Postcondición:** El pedido pasara a aparecer como retrasado y se le notifica al cliente.

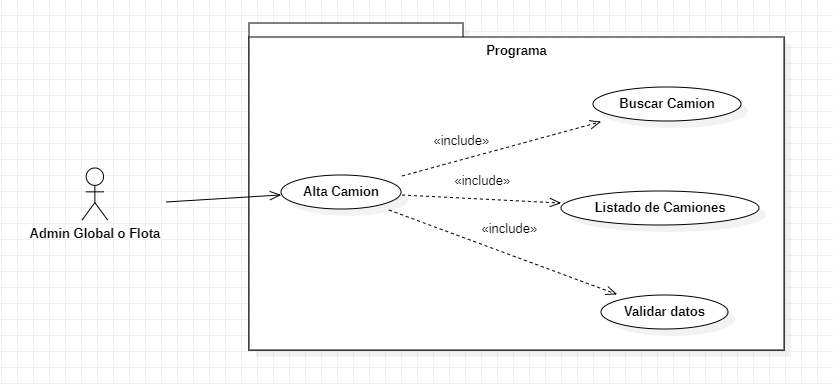
**Validaciones:** El pedido debe existir, debe ser del tipo asignado o confirmado y se debe agregar la razón del retraso del pedido

**Frecuencia de uso:** Baja.

**ABM de camiones**



**Alta camión**



**Identificador:** CU 021

**Nombre del CU:** Alta Camionero

**Actores:** Admin global/Admin de flota

**Descripción:** Permite dar de alta un camión.

**Referencia a Requerimientos Funcionales:** 14

**Precondición:** El admin debe estar con la sesión iniciada.

**Desencadenante:** El admin decide agregar un camión.

**Curso normal:**

1. El admin ingresa al listado de camiones.
2. El sistema redirige al admin al listado de los camiones.
3. El admin ingresa a la función que dice “Agregar un camión”.
4. El sistema redirige al admin a un formulario para agregar un camión.
5. El admin ingresa los datos.
6. El sistema verifica que los campos no estén vacíos.
7. El sistema verifica que el nuevo camión no exista en la base de datos.
8. El sistema ingresa el nuevo camión a la base de datos.
9. El sistema redirige a la página de gestión de camión y se actualiza el listado.
10. Fin del caso de uso.

**Curso alternativo:**

6.1. El sistema detecta un campo vacío e informa al admin.

6.1.1. El admin completa el campo vacío y continua con el flujo normal.

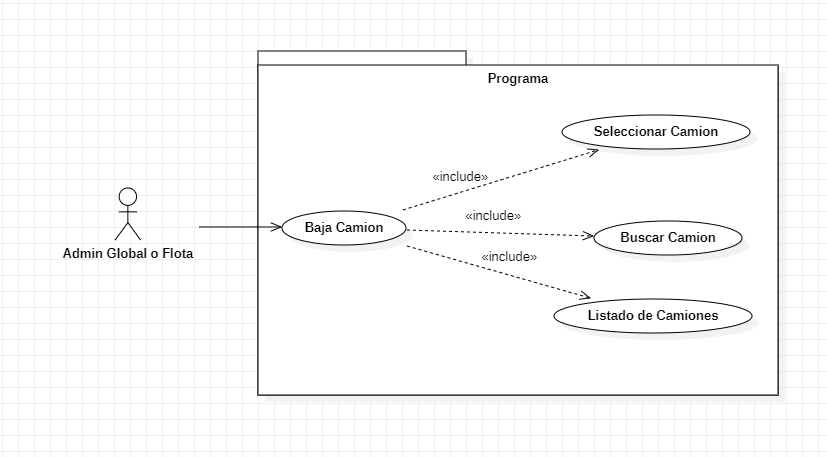
7.1. El sistema detecta que el nuevo camión ya existe en la base de datos y avisa a través de un mensaje al admin.

**Postcondición:**  Se agrega un nuevo camión al sistema.

**Validaciones:** El sistema verifica que los campos no estén vacíos y que el camión no exista en la base de datos.

**Frecuencia de uso:** Media.

**Baja camión**



**Identificador:** CU 022

**Nombre del CU:** Baja camión

**Actores:** Admin global/Admin de flota

**Descripción:** Permite eliminar un camión del sistema.

**Referencia a Requerimientos Funcionales:** 14

**Precondición:** El admin debe estar con la sesión iniciada y debe existir algún camión en el sistema.

**Desencadenante:** El admin decide eliminar un camión.

**Curso normal:**

1. El admin ingresa al listado de camiones.
2. El sistema redirige al admin al listado de los camiones.
3. El admin selecciona un camión para eliminar.
4. El sistema solicita una confirmación de baja.
5. El admin confirma la baja.
6. El sistema verifica que el camión exista en la base de datos.
7. El software verifica que el camión no tenga un viaje pendiente o esté realizando un viaje
8. Si existe el camión y no tiene un viaje pendiente o no está realizando un pedido, el camión es eliminado.
9. El sistema recarga la página actualizando el listado.
10. Fin del caso de uso.

**Curso alternativo:**

5.1. El admin cancela la baja, dejando el caso de uso sin efecto.

6.1. El sistema detecta que el camión no existe e informa al admin.

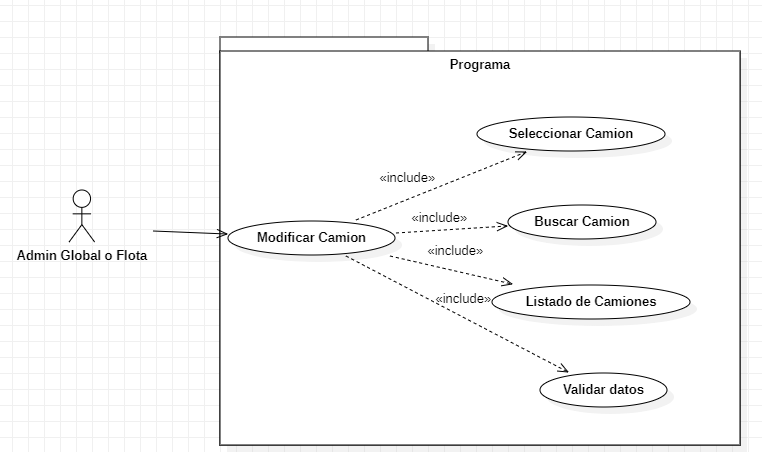
7.1. El sistema detecta que el camión tiene un pedido pendiente o está en un viaje, por lo que el sistema avisa que se deberá esperar a que termine el/los pedidos o se elimine estos para continuar con la eliminación del camión.

**Postcondición:** Se elimina un camión del sistema.

**Validaciones:** El sistema verifica que el camión exista en la base de datos y que no tenga pedidos pendientes o esté en un viaje para eliminarlo.

**Frecuencia de uso:** Baja.

**Modificar camión**



**Identificador:** CU 023

**Nombre del CU:** Modificar camión

**Actores:** Admin global/Admin de flota

**Descripción:** Permite modificar un camión del sistema.

**Referencia a Requerimientos Funcionales:** 14

**Precondición:** El admin debe estar con la sesión iniciada y debe existir algún camión en el sistema.

**Desencadenante:** El admin decide modificar un camión.

**Curso normal:**

1. El admin ingresa al listado de camiones.
2. El sistema redirige al admin al listado de los camiones.
3. El admin selecciona un camión para modificar.
4. El sistema redirige al admin a un formulario con los datos del camión.
5. El admin modifica los datos y finaliza el proceso.
6. El sistema solicita una confirmación de modificación.
7. El admin confirma la modificación.
8. El sistema verifica que no haya algún campo vacío.
9. El sistema verifica que el camión exista en la base de datos.
10. El sistema actualiza el camión en la base de datos.
11. El sistema redirige al admin al listado de camiones actualizándolo.
12. Fin del caso de uso.

**Curso alternativo:**

7.1. El admin cancela la modificación, dejando el caso de uso sin efecto.

8.1. El sistema detecta un campo vacío e informa al admin.

8.1.1. El admin completa el campo vacío y continua con el flujo norma

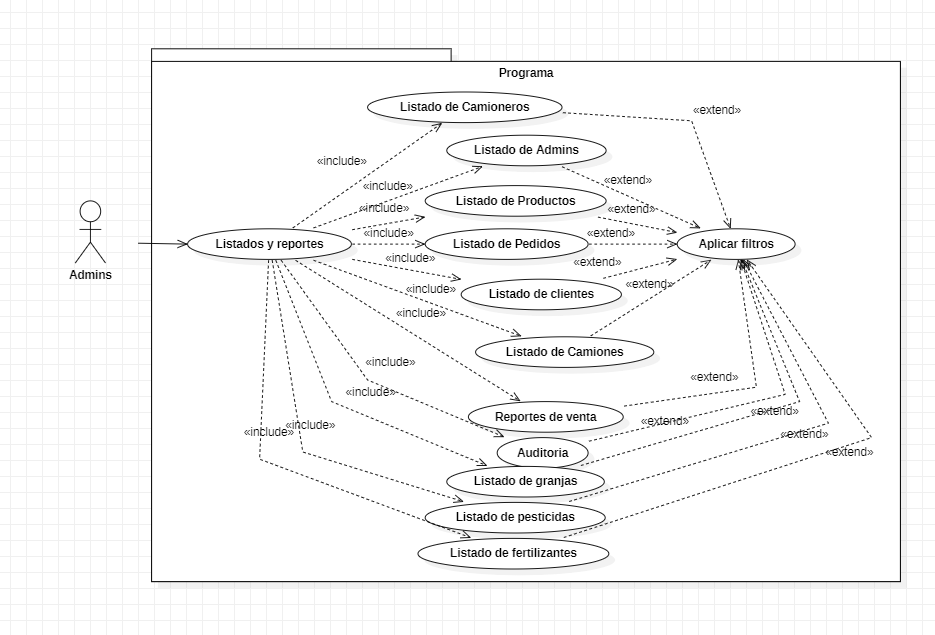
9.1. El sistema detecta que el camión no existe e informa al admin.

**Postcondición:** Se modifica un camión en el sistema.

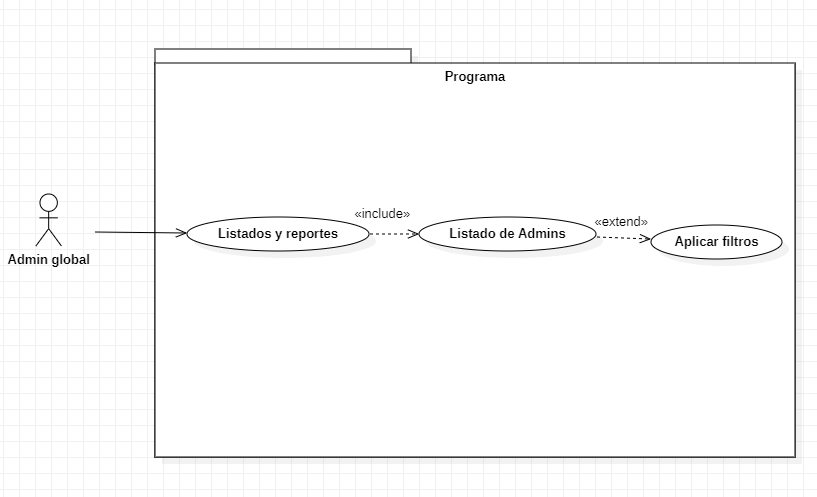
**Validaciones:** El sistema verifica que los campos no estén vacíos y que el camión exista en la base de datos.

**Frecuencia de uso:** Baja.

**Listados y reportes**



**Listado de admins**



**Identificador:** CU 024

**Nombre del CU:** Listado de admins.

**Actores:** Admin global

**Descripción:** Permite consultar el listado de Admins.

**Referencia a Requerimientos Funcionales: 13**

**Precondición:** El admin global debe estar con la sesión iniciada y debe existir algún admin en el sistema.

**Desencadenante:** El admin decide consultar la lista de admins.

**Curso normal:**

1. El admin ingresa a la función que dice “Listados y reportes”.
2. El sistema redirige al admin a la sección de Listados y reportes.
3. El admin selecciona la lista de admins.
4. El sistema redirige al admin al listado de admins.
5. El admin puede buscar uno o varios admins atreves del filtrado y buscador.
6. El admin desde ahí puede agregar, modificar y eliminar uno o varios admins.
7. El admin cierra el listado.
8. Fin caso de uso.

**Curso alternativo:**

4.1 El sistema no tiene registrados admins.

5.1 El admin no encuentra los admins que busca.

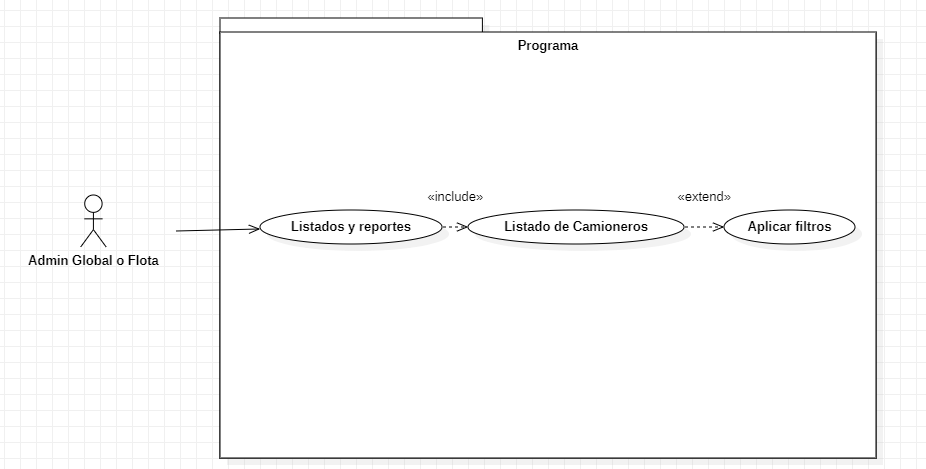
5.1.1 El admin cambia los filtros y encuentra los admins, se continua con el curso normal

**Postcondición:** Se chequea la lista de admins

**Validaciones:** El sistema verifica que el admin sea global y que haya admins registrados en el sistema.

**Frecuencia de uso: Alta**.

**Listado de Camioneros**



**Identificador:** CU 025

**Nombre del CU:** Listado de camioneros.

**Actores:** Admin global/Admin de flota.

**Descripción:** Permite consultar el listado de camioneros.

**Referencia a Requerimientos Funcionales: 14**

**Precondición:** El admin debe estar con la sesión iniciada y debe existir algún camionero en el sistema.

**Desencadenante:** El admin decide consultar la lista de camioneros.

**Curso normal:**

1. El admin ingresa a la función que dice “Listados y reportes”.
2. El sistema redirige al admin a la sección de Listados y reportes.
3. El admin selecciona la lista de camioneros.
4. El sistema redirige al admin al listado de camioneros.
5. El admin puede buscar uno o varios camioneros atreves del filtrado y buscador.
6. El admin desde ahí puede agregar, modificar o eliminar uno o varios camioneros.
7. El admin cierra el listado.
8. Fin caso de uso.

**Curso alternativo:**

4.1 El sistema no tiene registrados camioneros.

5.1 El admin no encuentra los camioneros que busca.

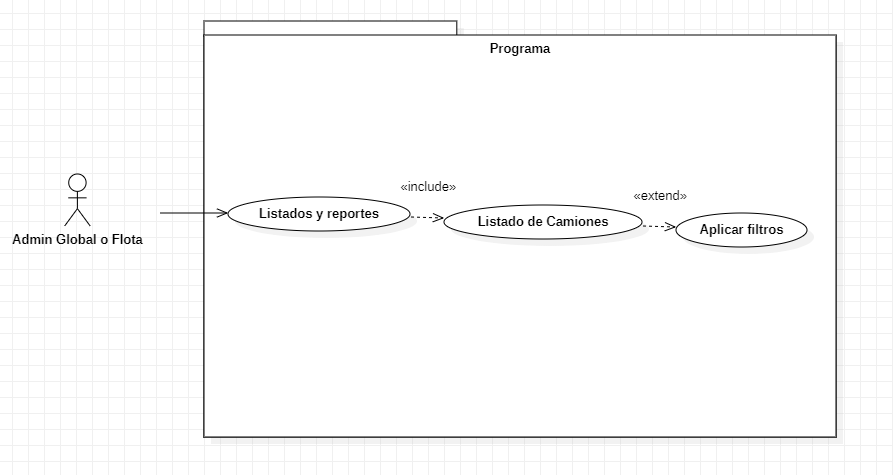
5.1.1 El admin cambia los filtros y encuentra los camioneros, se continua con el curso normal.

**Postcondición:** Se chequea la lista de camioneros.

**Validaciones:** El sistema verifica que el admin sea del tipo global o de flota y que haya camioneros registrados en el sistema.

**Frecuencia de uso: Alta**.

**Listado de camiones**



**Identificador:** CU 026

**Nombre del CU:** Listado de camiones.

**Actores:** Admin global/Admin de flota.

**Descripción:** Permite consultar el listado de camiones.

**Referencia a Requerimientos Funcionales: 14**

**Precondición:** El admin debe estar con la sesión iniciada y debe existir algún camión en el sistema.

**Desencadenante:** El admin decide consultar la lista de camiones.

**Curso normal:**

1. El admin ingresa a la función que dice “Listados y reportes”.
2. El sistema redirige al admin a la sección de Listados y reportes.
3. El admin selecciona la lista de camiones.
4. El sistema redirige al admin al listado de camiones.
5. El admin puede buscar uno o varios camiones atreves del filtrado y buscador.
6. El admin desde ahí puede agregar, modificar o eliminar uno o varios camiones.
7. El admin cierra el listado.
8. Fin caso de uso.

**Curso alternativo:**

4.1 El sistema no tiene registrados camiones.

5.1 El admin no encuentra los camiones que busca

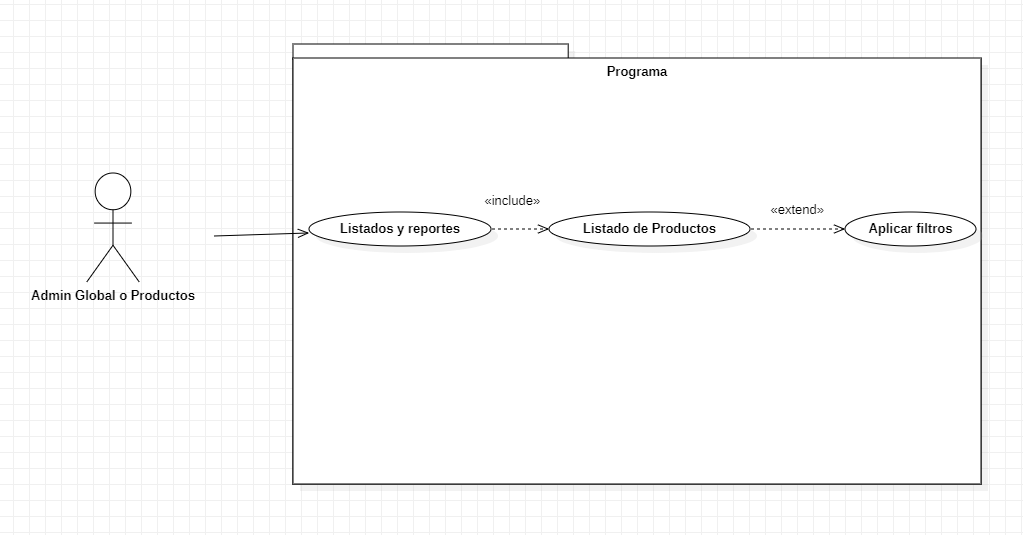
5.1.1 El admin cambia los filtros y encuentra los camiones, se continua con el curso normal

**Postcondición:** Se chequea la lista de camiones.

**Validaciones:** El sistema verifica que el admin sea del tipo global o de flota y que haya camiones registrados en el sistema.

**Frecuencia de uso: Alta**.

**Listado de productos**



**Identificador:** CU 027

**Nombre del CU:** Listado de productos.

**Actores:** Admin global/Admin de productos.

**Descripción:** Permite consultar el listado de productos.

**Referencia a Requerimientos Funcionales: 17**

**Precondición:** El admin debe estar con la sesión iniciada y debe existir algún producto en el sistema.

**Desencadenante:** El admin decide consultar la lista de productos.

**Curso normal:**

1. El admin ingresa a la función que dice “Listados y reportes”.
2. El sistema redirige al admin a la sección de Listados y reportes.
3. El admin selecciona la lista de productos.
4. El sistema redirige al admin al listado de productos.
5. El admin puede buscar uno o varios productos atreves del filtrado y buscador.
6. El admin desde ahí puede agregar, modificar o eliminar uno o varios productos.
7. El admin cierra el listado.
8. Fin caso de uso.

**Curso alternativo:**

4.1 El sistema no tiene registrados ningún producto.

5.1 El admin no encuentra los productos que busca

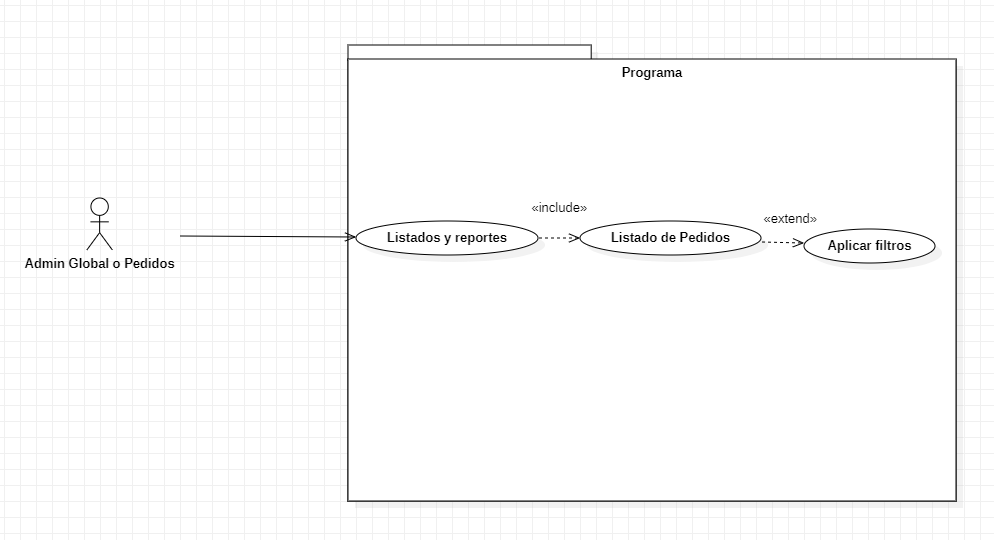
5.1.1 El admin cambia los filtros y encuentra los productos, se continua con el curso normal

**Postcondición:** Se chequea la lista de productos.

**Validaciones:** El sistema verifica que el admin sea del tipo global o de productos y que haya productos registrados en el sistema.

**Frecuencia de uso: Alta**.

**Listado de pedidos**



**Identificador:** CU 028

**Nombre del CU:** Listado de pedidos.

**Actores:** Admin global/Admin de pedidos.

**Descripción:** Permite consultar el listado de pedidos.

**Referencia a Requerimientos Funcionales: 18**

**Precondición:** El admin debe estar con la sesión iniciada y debe existir algún pedido en el sistema.

**Desencadenante:** El admin decide consultar la lista de pedidos.

**Curso normal:**

1. El admin ingresa a la función que dice “Listados y reportes”.
2. El sistema redirige al admin a la sección de Listados y reportes.
3. El admin selecciona la lista de pedidos.
4. El sistema redirige al admin al listado de pedidos.
5. El admin puede buscar uno o varios pedidos atreves del filtrado y buscador.
6. El admin desde ahí puede agregar, modificar o eliminar uno o varios pedidos.
7. El admin cierra el listado.
8. Fin caso de uso.

**Curso alternativo:**

4.1 El sistema no tiene registrado ningún pedido.

5.1 El admin no encuentra los pedidos que busca.

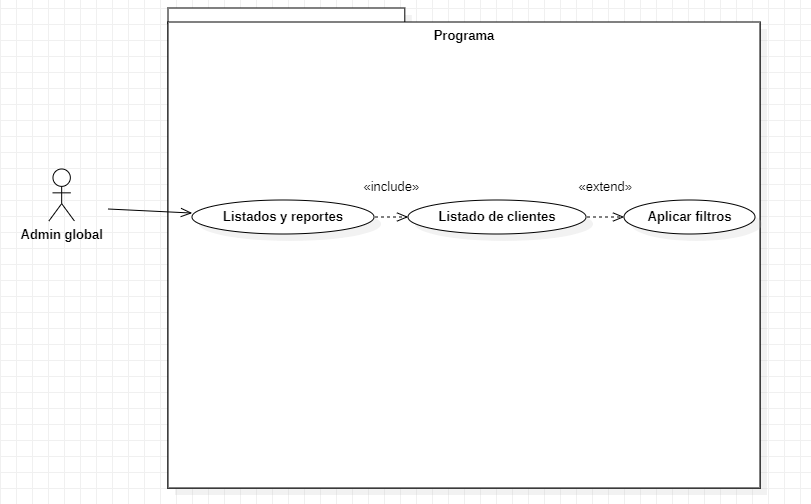
5.1.1 El admin cambia los filtros y encuentra los pedidos, se continua con el curso normal

**Postcondición:** Se chequea la lista de pedidos.

**Validaciones:** El sistema verifica que el admin sea del tipo global o de pedidos y que haya pedidos registrados en el sistema.

**Frecuencia de uso: Alta**.

**Listado de Clientes**



**Identificador:** CU 029

**Nombre del CU:** Listado de clientes.

**Actores:** Admin global.

**Descripción:** Permite consultar el listado de clientes.

**Referencia a Requerimientos Funcionales: 11 a**

**Precondición:** El admin debe estar con la sesión iniciada y debe existir algún cliente en el sistema.

**Desencadenante:** El admin decide consultar la lista de clientes.

**Curso normal:**

1. El admin ingresa a la función que dice “Listados y reportes”.
2. El sistema redirige al admin a la sección de Listados y reportes.
3. El admin selecciona la lista de clientes.
4. El sistema redirige al admin al listado de clientes.
5. El admin puede buscar uno o varios clientes atreves del filtrado y buscador.
6. El admin desde ahí puede eliminar uno o varios clientes.
7. El admin cierra el listado.
8. Fin caso de uso.

**Curso alternativo:**

4.1 El sistema no tiene clientes registrados

5.1 El admin no encuentra los clientes que busca

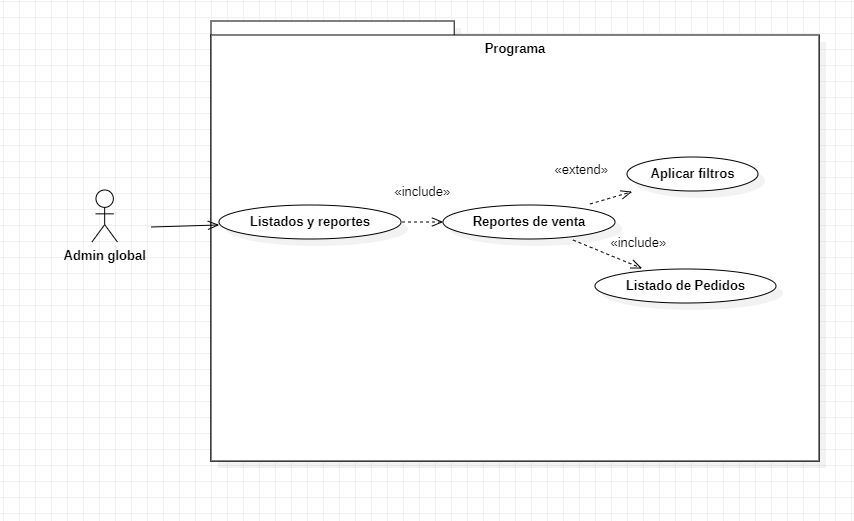
5.1.1 El admin cambia los filtros y encuentra los clientes, se continua con el curso normal

**Postcondición:** Se chequea la lista de clientes.

**Validaciones:** El sistema verifica que el admin sea del tipo global y que haya clientes registrados en el sistema.

**Frecuencia de uso: Alta**.

**Reportes de venta**



**Identificador:** CU 030

**Nombre del CU:** Reportes de venta.

**Actores:** Admin global.

**Descripción:** Permite consultar reportes de venta.

**Referencia a Requerimientos Funcionales: 20**

**Precondición:** El admin debe estar con la sesión iniciada y debe existir algún pedido entregado en el sistema.

**Desencadenante:** El admin decide consultar los Reportes de venta.

**Curso normal:**

1. El admin ingresa a la función que dice “Listados y reportes”.
2. El sistema redirige al admin a la sección de Listados y reportes.
3. El admin selecciona reportes de venta.
4. El sistema redirige al admin a los reportes de venta.
5. El admin puede buscar uno o varios reportes atreves del filtrado y buscador.
6. El admin desde ahí puede consultar los distintos reportes.
7. El admin cierra el listado.
8. Fin caso de uso.

**Curso alternativo:**

4.1 El sistema no tiene registrados pedidos entregados.

5.1 El admin no encuentra a los reportes que busca.

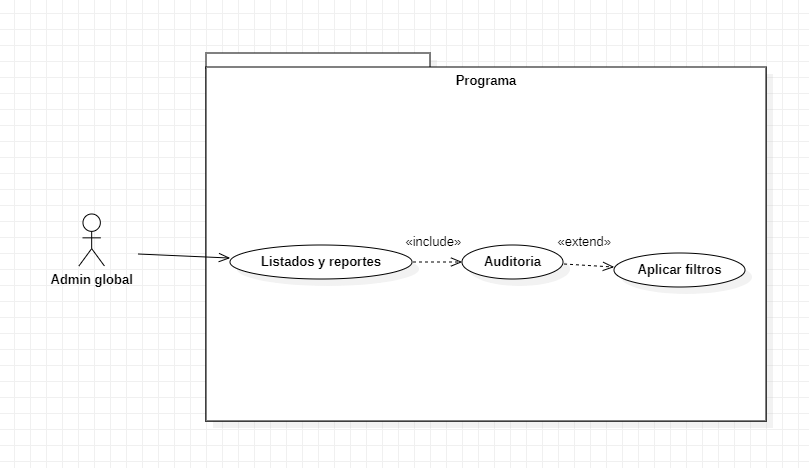
5.1.1 El admin cambia los filtros y encuentra los reportes, se continua con el curso normal.

**Postcondición:** Se chequea la sección de reportes de venta.

**Validaciones:** El sistema verifica que el admin sea del tipo global y que haya pedidos entregados registrados en el sistema.

**Frecuencia de uso:** Baja.

**Auditoria**



**Identificador:** CU 031

**Nombre del CU:** Auditoria.

**Actores:** Admin global.

**Descripción:** Consultar auditoria.

**Referencia a Requerimientos Funcionales: 21**

**Precondición:** El admin debe estar con la sesión iniciada y debe haber movimientos en el sistema.

**Desencadenante:** El admin decide consultar la Auditoria.

**Curso normal:**

1. El admin ingresa a la función que dice “Listados y reportes”.
2. El sistema redirige al admin a la sección de Listados y reportes.
3. El admin selecciona auditoria.
4. El sistema redirige al admin a auditoria.
5. El admin puede buscar uno o varios reportes de auditoria atreves del filtrado y buscador.
6. El admin desde ahí puede consultar los distintos reportes.
7. El admin cierra el listado.
8. Fin caso de uso.

**Curso alternativo:**

4.1 El sistema no tiene registrados ningún movimiento

5.1 El admin no encuentra a los reportes de auditoría que busca.

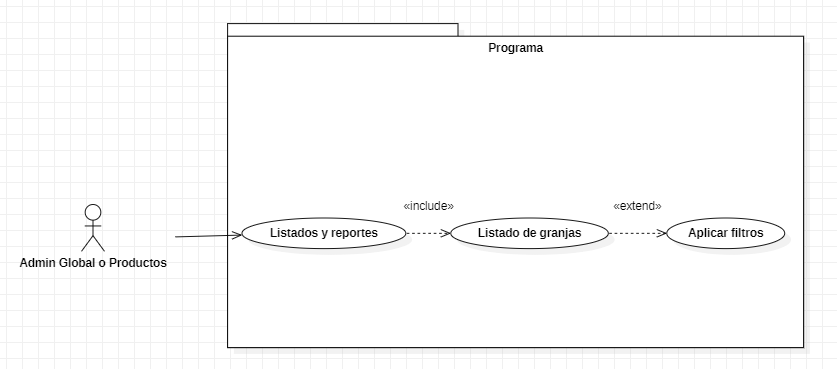
5.1.1 El admin cambia los filtros y encuentra los reportes de auditoría, se continua con el curso normal.

**Postcondición:** Se chequea la sección de auditoría.

**Validaciones:** El sistema verifica que el admin sea del tipo global y que haya movimientos registrados en el sistema.

**Frecuencia de uso:** Baja.

**Listado de granjas**



**Identificador:** CU 032

**Nombre del CU:** Listado de granjas.

**Actores:** Admin global/Admin de productos.

**Descripción:** Permite consultar el listado de granjas.

**Referencia a Requerimientos Funcionales: 15**

**Precondición:** El admin debe estar con la sesión iniciada y debe existir alguna granja en el sistema.

**Desencadenante:** El admin decide consultar la lista de granjas

**Curso normal:**

1. El admin ingresa a la función que dice “Listados y reportes”.
2. El sistema redirige al admin a la sección de Listados y reportes.
3. El admin selecciona la lista de granjas.
4. El sistema redirige al admin al listado de granjas.
5. El admin puede buscar uno o varias granjas atreves del filtrado y buscador.
6. El admin desde ahí puede agregar, modificar o eliminar uno o varias granjas.
7. El admin cierra el listado.
8. Fin caso de uso.

**Curso alternativo:**

4.1 El sistema no tiene registrado ninguna granja.

5.1 El admin no encuentra las granjas que busca.

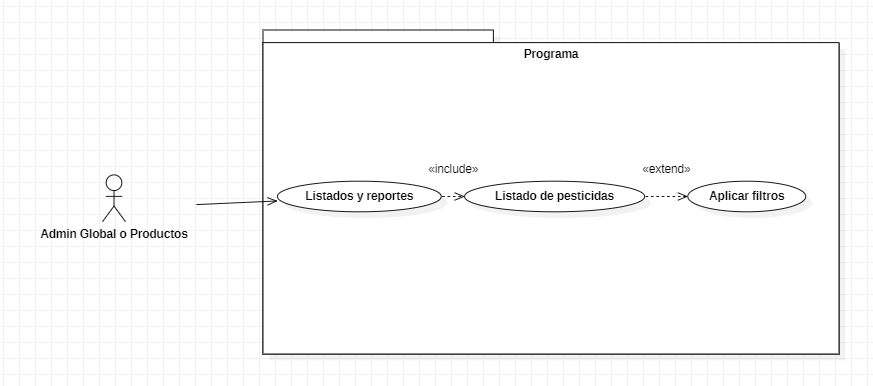
5.1.1 El admin cambia los filtros y encuentra las granjas, se continua con el curso normal

**Postcondición:** Se chequea la lista de granjas.

**Validaciones:** El sistema verifica que el admin sea del tipo global o de productos y que haya granjas registrados en el sistema.

**Frecuencia de uso: Alta**.

**Listado de pesticidas**



**Identificador:** CU 033

**Nombre del CU:** Listado de pesticidas.

**Actores:** Admin global/Admin de productos.

**Descripción:** Permite consultar el listado de pesticidas.

**Referencia a Requerimientos Funcionales: 15**

**Precondición:** El admin debe estar con la sesión iniciada y debe existir algún pesticida en el sistema.

**Desencadenante:** El admin decide consultar la lista de pesticidas.

**Curso normal:**

1. El admin ingresa a la función que dice “Listados y reportes”.
2. El sistema redirige al admin a la sección de Listados y reportes.
3. El admin selecciona la lista de pesticidas.
4. El sistema redirige al admin al listado de pesticidas.
5. El admin puede buscar uno o varios pesticidas atreves del filtrado y buscador.
6. El admin desde ahí puede agregar, modificar o eliminar uno o varias pesticidas.
7. El admin cierra el listado.
8. Fin caso de uso.

**Curso alternativo:**

4.1 El sistema no tiene registrado ningún pesticida.

5.1 El admin no encuentra los pesticidas que busca.

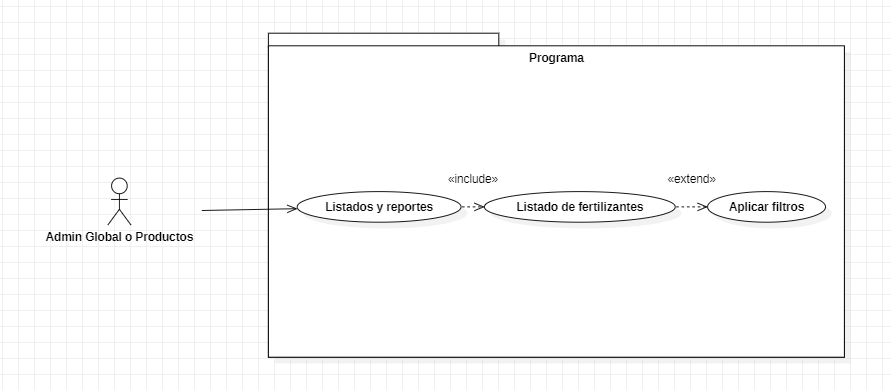
5.1.1 El admin cambia los filtros y encuentra los pesticidas, se continua con el curso normal

**Postcondición:** Se chequea la lista de pesticidas.

**Validaciones:** El sistema verifica que el admin sea del tipo global o de productos y que haya pesticidas registrados en el sistema.

**Frecuencia de uso: Alta**.

**Listado de fertilizantes**



**Identificador:** CU 034

**Nombre del CU:** Listado de fertilizantes.

**Actores:** Admin global/Admin de productos.

**Descripción:** Permite consultar el listado de fertilizantes.

**Referencia a Requerimientos Funcionales: 15**

**Precondición:** El admin debe estar con la sesión iniciada y debe existir algún fertilizante en el sistema.

**Desencadenante:** El admin decide consultar la lista de fertilizantes.

**Curso normal:**

1. El admin ingresa a la función que dice “Listados y reportes”.
2. El sistema redirige al admin a la sección de Listados y reportes.
3. El admin selecciona la lista de fertilizantes.
4. El sistema redirige al admin al listado de fertilizantes.
5. El admin puede buscar uno o varios fertilizantes atreves del filtrado y buscador.
6. El admin desde ahí puede agregar, modificar o eliminar uno o varias fertilizantes.
7. El admin cierra el listado.
8. Fin caso de uso.

**Curso alternativo:**

4.1 El sistema no tiene registrado ningún fertilizante.

5.1 El admin no encuentra los fertilizantes que busca.

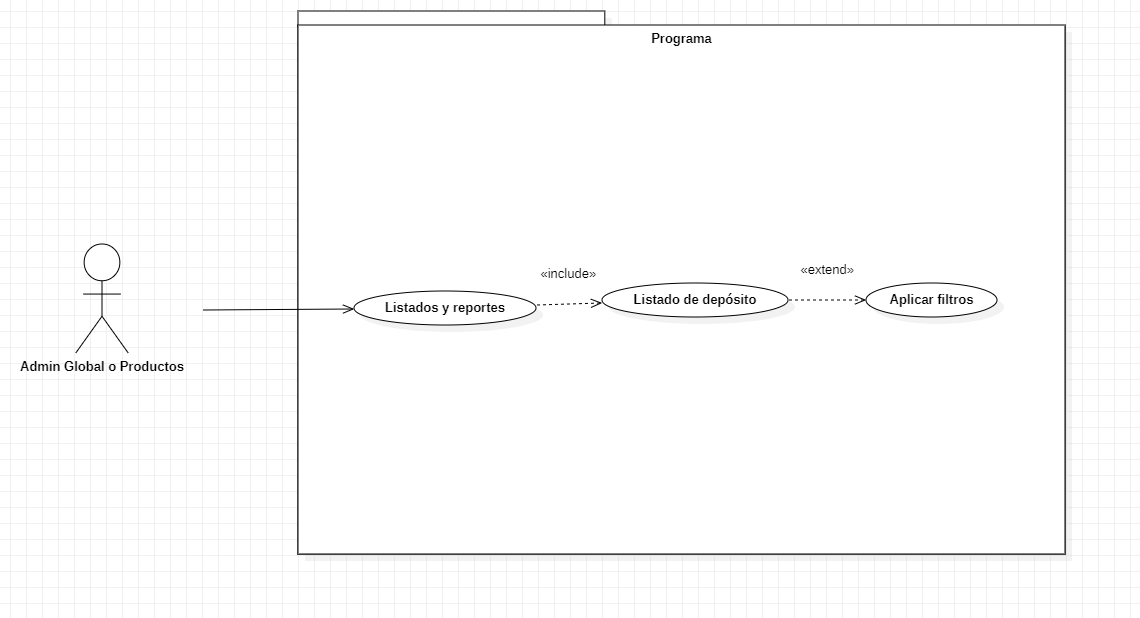
5.1.1 El admin cambia los filtros y encuentra los fertilizantes, se continua con el curso normal

**Postcondición:** Se chequea la lista de fertilizantes.

**Validaciones:** El sistema verifica que el admin sea del tipo global o de productos y que haya fertilizantes registrados en el sistema.

**Frecuencia de uso: Alta**.

**Listado de depósitos**



**Identificador:** CU 035

**Nombre del CU:** Listado de depósitos.

**Actores:** Admin global/Admin de productos.

**Descripción:** Permite consultar el listado de depósitos.

**Referencia a Requerimientos Funcionales: 15**

**Precondición:** El admin debe estar con la sesión iniciada y debe existir algún depósito en el sistema.

**Desencadenante:** El admin decide consultar la lista de depósitos.

**Curso normal:**

1. El admin ingresa a la función que dice “Listados y reportes”.
2. El sistema redirige al admin a la sección de Listados y reportes.
3. El admin selecciona la lista de depósitos.
4. El sistema redirige al admin al listado de depósitos.
5. El admin puede buscar uno o varios depósitos atreves del filtrado y buscador.
6. El admin desde ahí puede agregar, modificar o eliminar uno o varios depósitos.
7. El admin cierra el listado.
8. Fin caso de uso.

**Curso alternativo:**

4.1 El sistema no tiene registrado ningún depósito.

5.1 El admin no encuentra los depósitos que busca.

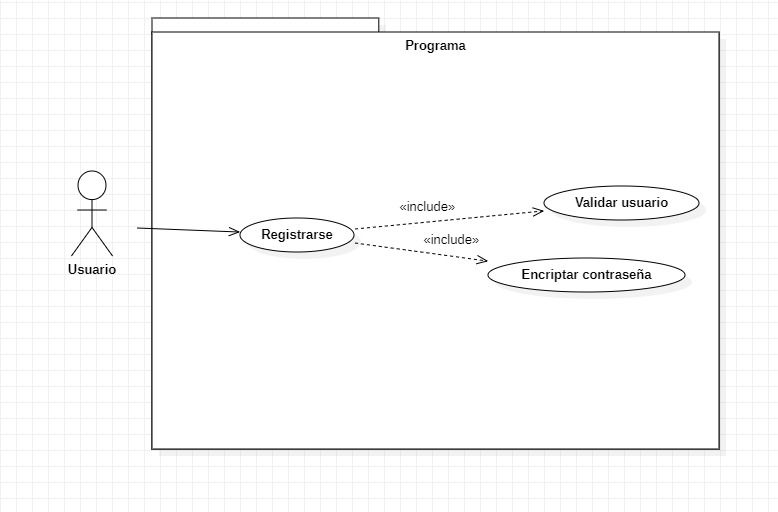
5.1.1 El admin cambia los filtros y encuentra los depósitos, se continua con el curso normal

**Postcondición:** Se chequea la lista de depósitos.

**Validaciones:** El sistema verifica que el admin sea del tipo global o de productos y que haya depósitos registrados en el sistema.

**Frecuencia de uso: Alta**.

**Registrarse**



**Identificador** CU 036

**Nombre del CU:** Registrarse

**Actores:** Usuarios

**Descripción:** El usuario se registra con su nombre de usuario y contraseña, permitiendo acceder a las funcionabilidades del sistema correspondientes.

**Referencia a Requerimientos Funcionales:** 1

**Precondición:** Se debe contar con un mail.

**Desencadenante:** El usuario se registra en el sistema.

**Curso normal:**

1. El usuario accede a la sección de Login.
2. El usuario accede a la sección de registrarse.
3. El usuario completa los campos requeridos.
4. El sistema valida que los campos no estén vacíos.
5. El sistema valida las credenciales.
6. El sistema valida que el usuario no exista.
7. Se encripta la contraseña.
8. El sistema envía el usuario a la página principal.
9. Se registra el usuario
10. Fin del caso de uso.

**Curso alternativo:**

3.1. El sistema detecta un campo vacío e informa al usuario.

3.1.1. El usuario completa los campos vacíos y continua el flujo normal.

6.1. El sistema detecta que las credenciales corresponden a un usuario existente en el sistema y no permite el registro del usuario al sistema.

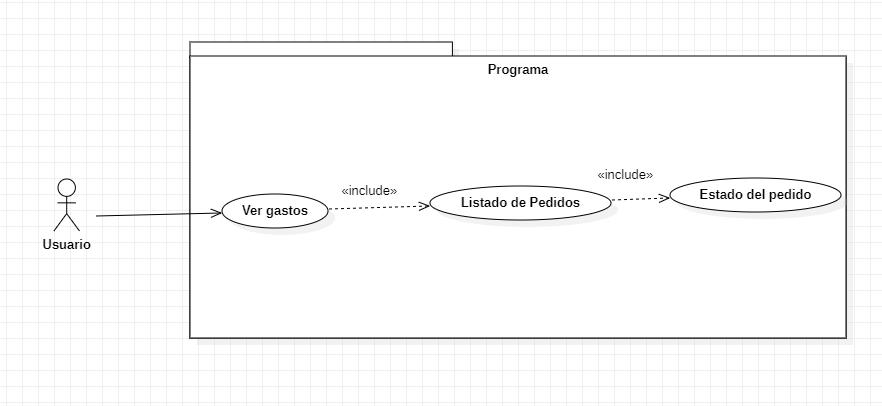
6.1.1. El usuario ingresa unos datos que no estén repetidos en el sistema y se continua el flujo normal.

**Postcondición:** El usuario queda registrado en el sistema y puede acceder a las funcionabilidades correspondientes.

**Validaciones:** El sistema valida que el usuario exista en el sistema y que los campos no estén vacíos.

**Frecuencia de uso:** Alta.

**Ver gastos**



**Identificador** CU 037

**Nombre del CU:** Ver gastos

**Actores:** Usuarios

**Descripción:** Permite al usuario ver sus propios gastos en el sistema y ver pedidos anteriores

**Referencia a Requerimientos Funcionales:** 10

**Precondición:** Se debe tener un usuario registrado y haber realizado pedidos finalizados en el sistema.

**Desencadenante:** El usuario permite ver el historial de pedidos y los gastos en el sistema.

**Curso normal:**

1. El usuario se debe redirigir a su propio perfil y acceder a la sección de “Ver mis gastos”.
2. El sistema redirige al usuario a “Ver mis gastos”
3. El usuario puede ver sus pedidos anteriores además de ver el gasto total de todos sus pedidos.
4. El usuario puede buscar una o varios pedidos atreves del filtrado.
5. El usuario sale de la pantalla.
6. Fin del caso de uso.

**Curso alternativo:**

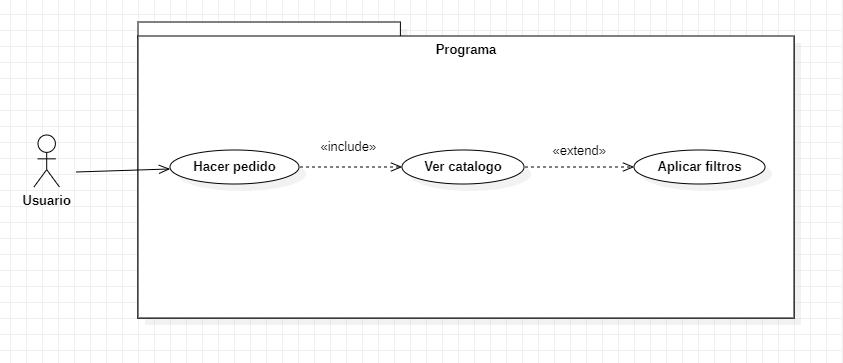
3.1. El sistema no tiene registrado pedidos terminados de este usuario, se finaliza el caso de uso

**Postcondición:** El usuario puede observar sus anteriores pedidos y ver sus gastos totales.

**Validaciones:** El sistema valida que el usuario exista en el sistema y que haya pedidos terminados asociados a este cliente

**Frecuencia de uso: Media**.

**Hacer pedido**



**Identificador** CU 038

**Nombre del CU:** Hacer pedido

**Actores:** Usuarios

**Descripción:** Permite hacer pedidos al sistema

**Referencia a Requerimientos Funcionales:** 4,5,6

**Precondición:** Se debe tener un usuario registrado.

**Desencadenante:** Permite al usuario realizar pedidos al sistema.

**Curso normal:**

1. El usuario se dirige al catálogo.
2. El usuario usa los filtros y buscadores para buscar los productos.
3. El usuario selecciona los productos a pedir.
4. El sistema verifica que haya una cantidad mayor registrada de producto de la que se pidió.
5. El sistema pide confirmar el pedido.
6. El usuario confirma el pedido.
7. Fin de caso de uso.

**Curso alternativo:**

3.1. El usuario no selecciona productos a pedir, finaliza el caso de uso.

4.1 No hay la suficiente cantidad de producto registrado, comparado con la cantidad de producto pedido, se notifica tanto al admin como al usuario, finaliza el caso de uso.

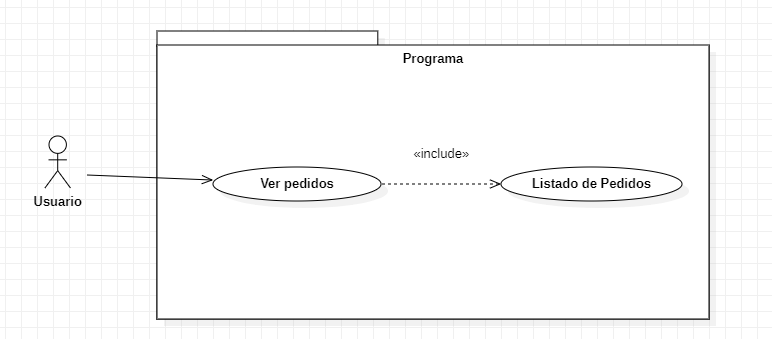
5.1 El usuario no confirma el pedido, finaliza el caso de uso.

**Postcondición:** El usuario realiza el pedido

**Validaciones:** El sistema valida que el usuario exista en el sistema y que la cantidad de producto pedido sea menor a la cantidad de producto registrado en el sistema

**Frecuencia de uso: Alta**.

**Ver estado del pedido**



**Identificador** CU 39

**Nombre del CU:** Ver estado del pedido

**Actores:** Usuarios

**Descripción:** Permite ver el estado del pedido

**Referencia a Requerimientos Funcionales:** 7

**Precondición:** Se debe haber realizado un pedido al sistema.

**Desencadenante:** Se consulta el estado del pedido del cliente.

**Curso normal:**

1. El usuario se debe redirigir a su propio perfil y acceder a la sección de “Ver mis pedidos”.
2. El sistema redirige al usuario a “Ver mis pedidos”
3. El usuario puede ver sus pedidos actuales y ver el estado actual en el que se encuentran.
4. El usuario puede buscar una o varios pedidos atreves del filtrado.
5. El usuario sale de la pantalla.
6. Fin del caso de uso.

**Curso alternativo:**

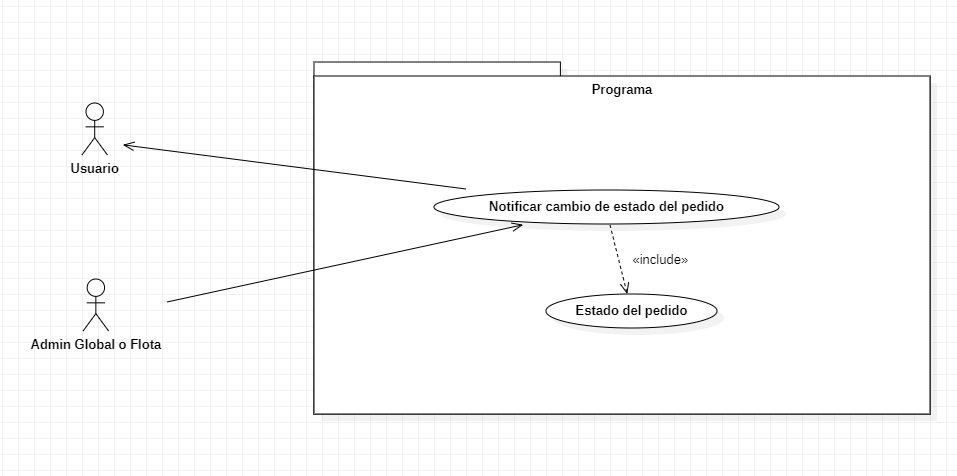
3.1. El sistema no tiene registrado pedidos actuales de este usuario, se finaliza el caso de uso.

**Postcondición:** El usuario puede observar sus pedidos actuales y ver el estado de este mismo.

**Validaciones:** El sistema valida que el usuario exista en el sistema y que haya pedidos sin terminar asociados a este cliente

**Frecuencia de uso: Media**.

**Notificar por cambio de estado del pedido**



**Identificador** CU 040

**Nombre del CU:** Notificar por cambio de estado del pedido

**Actores:** Usuarios y admin global o flota

**Descripción:** Notifica al usuario por cambio de estado del pedido

**Referencia a Requerimientos Funcionales:** 8

**Precondición:** Se debe haber realizado un pedido confirmado al sistema.

**Desencadenante:** Se notifica al usuario del cambio de estado de su pedido.

**Curso normal:**

1. El admin se dirige al listado de pedido confirmados.
2. El sistema redirige al admin al listado de pedidos confirmados.
3. El admin selecciona un pedido.
4. El admin selecciona el estado del pedido.
5. El admin confirma el estado.
6. Cierra la pantalla.
7. Le llega la notificación de actualización de pedido al cliente.
8. Se termina el caso de uso.

**Curso alternativo:**

2.1. El sistema no tiene registrado confirmados, se finaliza el caso de uso.

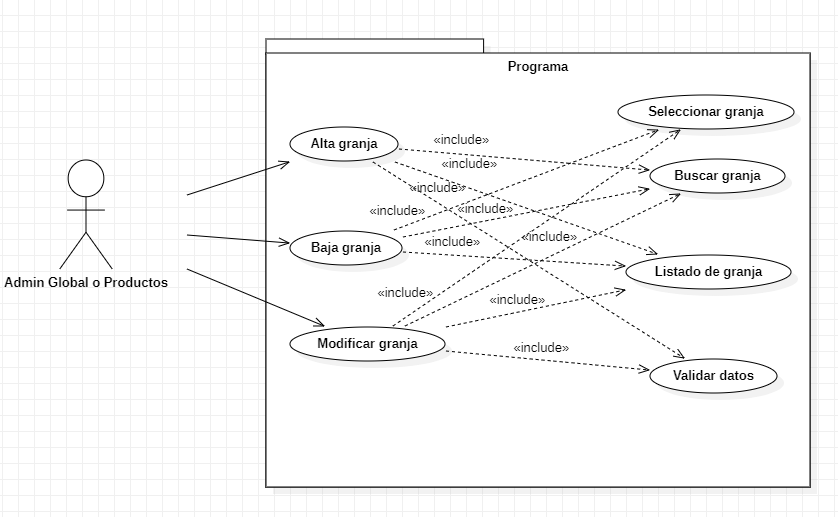
5.1 El admin no confirma el estado del pedido, se finaliza el caso de uso.

**Postcondición:** El usuario recibe una notificación por cambio de estado de su pedido.

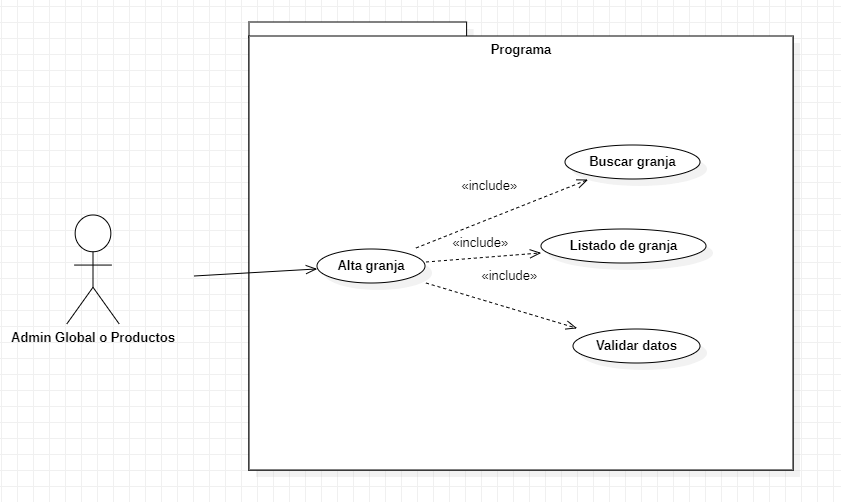
**Validaciones:** Se debe haber realizado pedidos confirmados en el sistema.

**Frecuencia de uso: Alta**.

**ABM de granja**



**Alta granja**



**Identificador:** CU 041

**Nombre del CU:** Alta granja

**Actores:** Admin global/Admin de productos

**Descripción:** Permite dar de alta una granja.

**Referencia a Requerimientos Funcionales: 15**

**Precondición:** El admin debe estar con la sesión iniciada.

**Desencadenante:** El admin decide agregar una granja.

**Curso normal:**

1. El admin ingresa al listado de granja.
2. El sistema redirige al admin al listado de granjas.
3. El admin ingresa a la función que dice “Agregar una granja”.
4. El sistema redirige al admin a un formulario para agregar una granja.
5. El admin ingresa los datos.
6. El sistema verifica que los campos no estén vacíos.
7. El sistema verifica que la nueva granja no exista en la base de datos.
8. El sistema ingresa la nueva granja a la base de datos.
9. El sistema redirige a la página de gestión de granjas y se actualiza el listado.
10. Fin del caso de uso.

**Curso alternativo:**

6.1. El sistema detecta un campo vacío e informa al admin.

6.1.1. El admin completa el campo vacío y continua con el flujo normal.

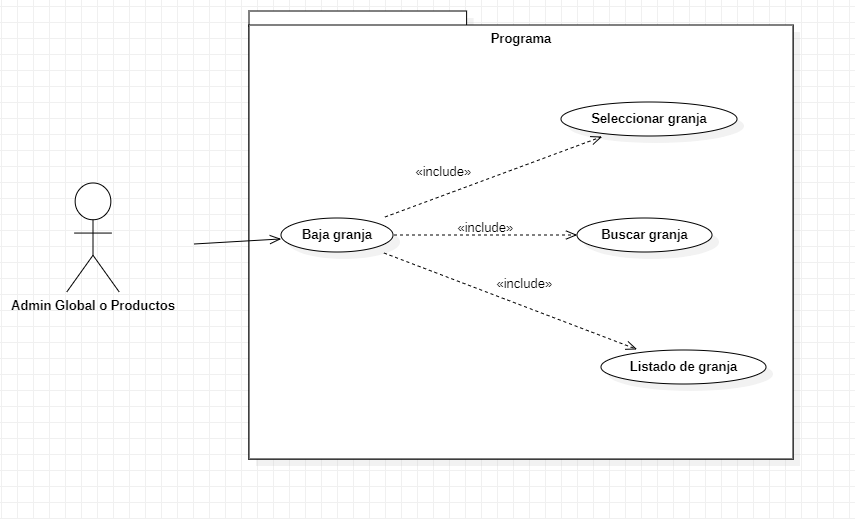
7.1. El sistema detecta que la nueva granja ya existe en la base de datos y avisa a través de un mensaje al admin.

**Postcondición:**  Se agrega una nueva granja al sistema.

**Validaciones:** El sistema verifica que los campos no estén vacíos y que la granja no exista en la base de datos.

**Frecuencia de uso: Baja**.

**Baja granja**



**Identificador:** CU 042

**Nombre del CU:** Baja granja

**Actores:** Admin global/Admin de productos

**Descripción:** Permite eliminar una granja del sistema.

**Referencia a Requerimientos Funcionales: 15**

**Precondición:** El admin debe estar con la sesión iniciada y debe existir alguna granja en el sistema.

**Desencadenante:** El admin decide eliminar una granja.

**Curso normal:**

1. El admin ingresa al listado de granjas.
2. El sistema redirige al admin al listado de granjas.
3. El admin selecciona una granja para eliminar.
4. El sistema solicita una confirmación de baja.
5. El admin confirma la baja.
6. El sistema verifica que la granja exista en la base de datos.
7. El software verifica que la granja no este asociado a ningún producto.
8. Si existe y no está asociado a ningún producto, la granja es eliminada.
9. El sistema recarga la página actualizando el listado.
10. Fin del caso de uso.

**Curso alternativo:**

5.1. El admin cancela la baja, dejando el caso de uso sin efecto.

6.1. El sistema detecta que la granja no existe e informa al admin.

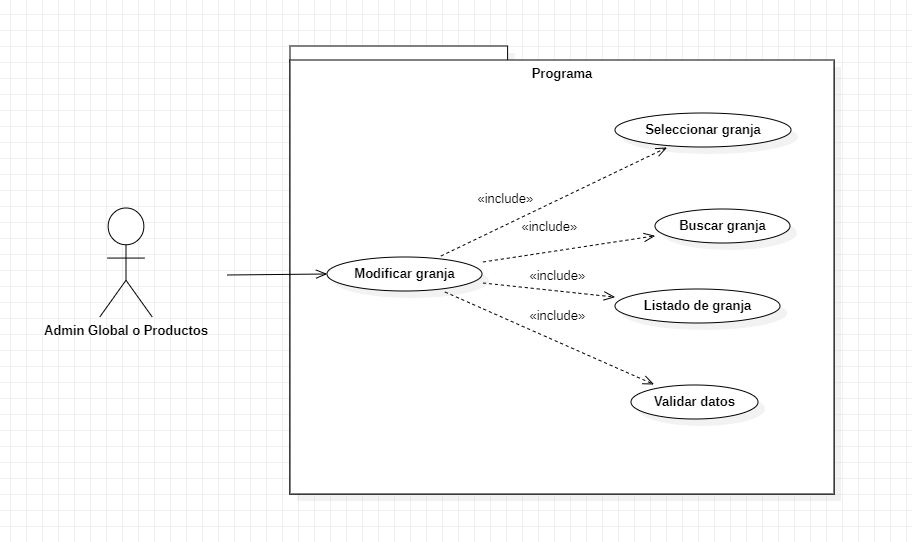
7.1. El sistema detecta que la granja está asociado a unos productos, por lo que el sistema avisa que se requiere eliminar estos para continuar con la eliminación de la granja.

**Postcondición:** Se elimina una granja del sistema.

**Validaciones:** El sistema verifica que la granja exista en la base de datos y que no esté asociado a ningún producto para eliminarlo.

**Frecuencia de uso:** Baja.

**Modificar granja**



**Identificador:** CU 043

**Nombre del CU:** Modificar granja

**Actores:** Admin global/Admin de productos

**Descripción:** Permite modificar una granja del sistema.

**Referencia a Requerimientos Funcionales: 15**

**Precondición:** El admin debe estar con la sesión iniciada y debe existir alguna granja en el sistema.

**Desencadenante:** El admin decide modificar una granja.

**Curso normal:**

1. El admin ingresa al listado de granjas.
2. El sistema redirige al admin al listado de las granjas.
3. El admin selecciona una granja para modificar.
4. El sistema redirige al admin a un formulario con los datos de la granja.
5. El admin modifica los datos y finaliza el proceso.
6. El sistema solicita una confirmación de modificación.
7. El admin confirma la modificación.
8. El sistema verifica que no haya algún campo vacío.
9. El sistema verifica que la granja exista en la base de datos.
10. El sistema actualiza la granja en la base de datos.
11. El sistema redirige al admin al listado de camiones actualizándolo.
12. Fin del caso de uso.

**Curso alternativo:**

7.1. El admin cancela la modificación, dejando el caso de uso sin efecto.

8.1. El sistema detecta un campo vacío e informa al admin.

8.1.1. El admin completa el campo vacío y continua con el flujo norma

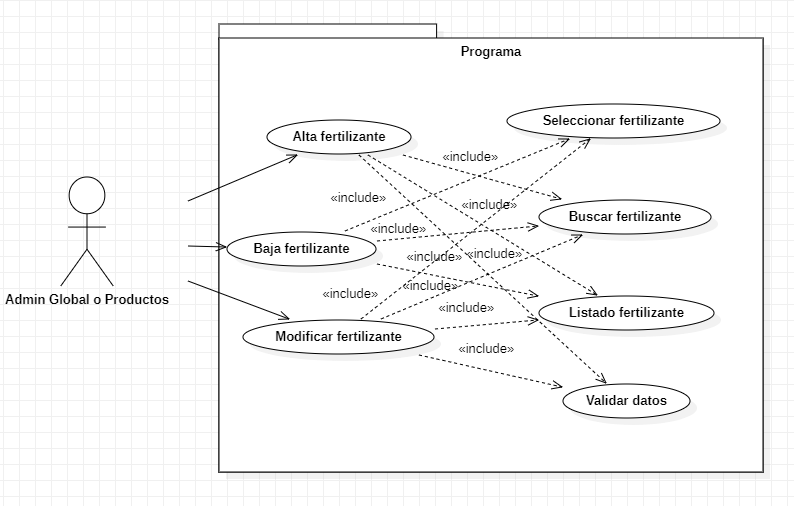
9.1. El sistema detecta que la granja no existe e informa al admin.

**Postcondición:** Se modifica una granja en el sistema.

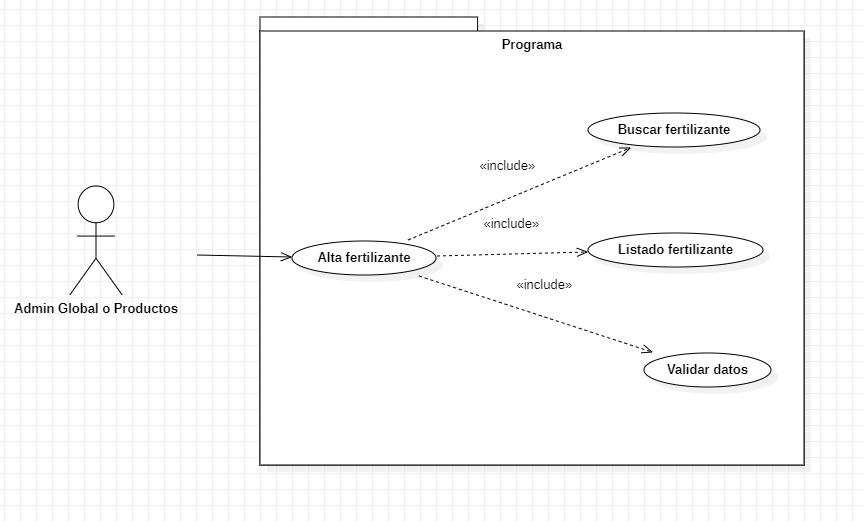
**Validaciones:** El sistema verifica que los campos no estén vacíos y que la granja exista en la base de datos.

**Frecuencia de uso:** Baja.

**ABM Fertilizante**



**Alta fertilizante**



**Identificador:** CU 044

**Nombre del CU:** Alta fertilizante

**Actores:** Admin global/Admin de productos

**Descripción:** Permite dar de alta un fertilizante.

**Referencia a Requerimientos Funcionales: 15**

**Precondición:** El admin debe estar con la sesión iniciada.

**Desencadenante:** El admin decide agregar un fertilizante.

**Curso normal:**

1. El admin ingresa al listado de fertilizantes.
2. El sistema redirige al admin al listado de fertilizantes.
3. El admin ingresa a la función que dice “Agregar un fertilizante”.
4. El sistema redirige al admin a un formulario para agregar un fertilizante.
5. El admin ingresa los datos.
6. El sistema verifica que los campos no estén vacíos.
7. El sistema verifica que el nuevo fertilizante no exista en la base de datos.
8. El sistema ingresa el nuevo fertilizante a la base de datos.
9. El sistema redirige a la página de gestión de fertilizantes y se actualiza el listado.
10. Fin del caso de uso.

**Curso alternativo:**

6.1. El sistema detecta un campo vacío e informa al admin.

6.1.1. El admin completa el campo vacío y continua con el flujo normal.

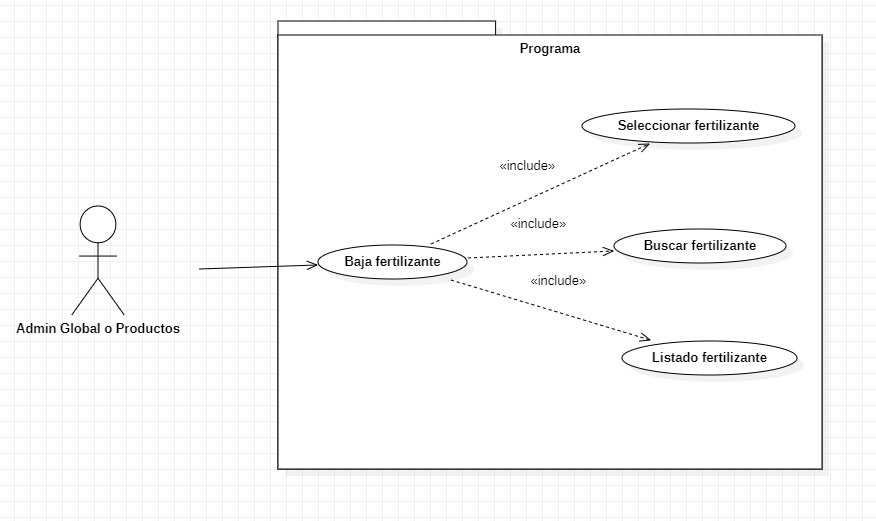
7.1. El sistema detecta que el nuevo fertilizante ya existe en la base de datos y avisa a través de un mensaje al admin.

**Postcondición:**  Se agrega un nuevo fertilizante al sistema.

**Validaciones:** El sistema verifica que los campos no estén vacíos y que el fertilizante no exista en la base de datos.

**Frecuencia de uso: Baja**.

**Baja fertilizante**



**Identificador:** CU 045

**Nombre del CU:** Baja fertilizante

**Actores:** Admin global/Admin de productos

**Descripción:** Permite eliminar un fertilizante del sistema.

**Referencia a Requerimientos Funcionales: 15**

**Precondición:** El admin debe estar con la sesión iniciada y debe existir algún fertilizante en el sistema.

**Desencadenante:** El admin decide eliminar un fertilizante.

**Curso normal:**

1. El admin ingresa al listado de fertilizantes.
2. El sistema redirige al admin al listado de fertilizantes.
3. El admin selecciona un fertilizante para eliminar.
4. El sistema solicita una confirmación de baja.
5. El admin confirma la baja.
6. El sistema verifica que el fertilizante exista en la base de datos.
7. El software verifica que el fertilizante no este asociado a ningún producto.
8. Si existe y no está asociado a ningún producto, el fertilizante es eliminado.
9. El sistema recarga la página actualizando el listado.
10. Fin del caso de uso.

**Curso alternativo:**

5.1. El admin cancela la baja, dejando el caso de uso sin efecto.

6.1. El sistema detecta que el fertilizante no existe e informa al admin.

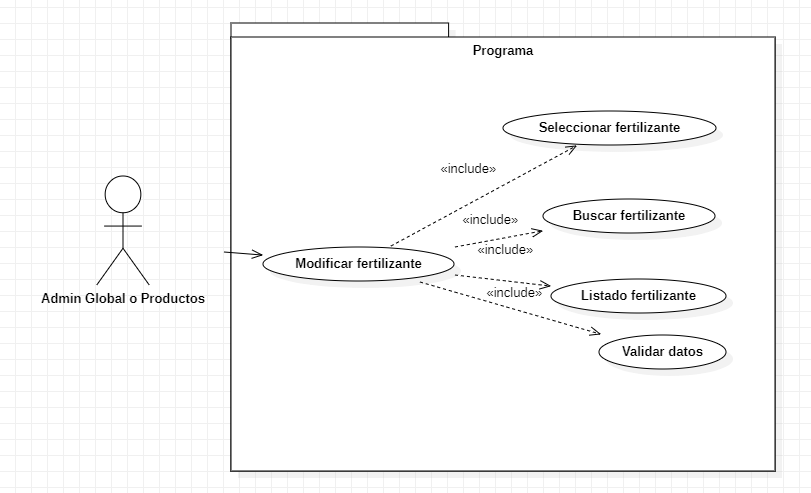
7.1. El sistema detecta que el fertilizante está asociado a unos productos, por lo que el sistema avisa que se requiere eliminar estos para continuar con la eliminación del fertilizante.

**Postcondición:** Se elimina un fertilizante del sistema.

**Validaciones:** El sistema verifica que el fertilizante exista en la base de datos y que no esté asociado a ningún producto para eliminarlo.

**Frecuencia de uso:** Baja.

**Modificar fertilizante**



**Identificador:** CU 046

**Nombre del CU:** Modificar fertilizante

**Actores:** Admin global/Admin de productos

**Descripción:** Permite modificar un fertilizante del sistema.

**Referencia a Requerimientos Funcionales: 15**

**Precondición:** El admin debe estar con la sesión iniciada y debe existir algún fertilizante en el sistema.

**Desencadenante:** El admin decide modificar un fertilizante

**Curso normal:**

1. El admin ingresa al listado de fertilizantes.
2. El sistema redirige al admin al listado de los fertilizantes.
3. El admin selecciona un fertilizante para modificar.
4. El sistema redirige al admin a un formulario con los datos del fertilizante.
5. El admin modifica los datos y finaliza el proceso.
6. El sistema solicita una confirmación de modificación.
7. El admin confirma la modificación.
8. El sistema verifica que no haya algún campo vacío.
9. El sistema verifica que el fertilizante exista en la base de datos.
10. El sistema actualiza el fertilizante en la base de datos.
11. El sistema redirige al admin al listado de fertilizantes actualizándolo.
12. Fin del caso de uso.

**Curso alternativo:**

7.1. El admin cancela la modificación, dejando el caso de uso sin efecto.

8.1. El sistema detecta un campo vacío e informa al admin.

8.1.1. El admin completa el campo vacío y continua con el flujo norma

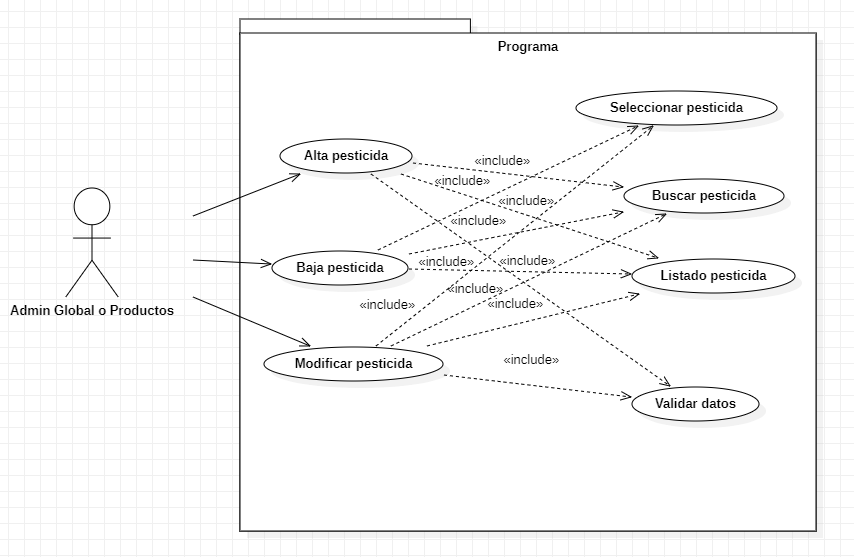
9.1. El sistema detecta que el fertilizante no existe e informa al admin.

**Postcondición:** Se modifica un fertilizante en el sistema.

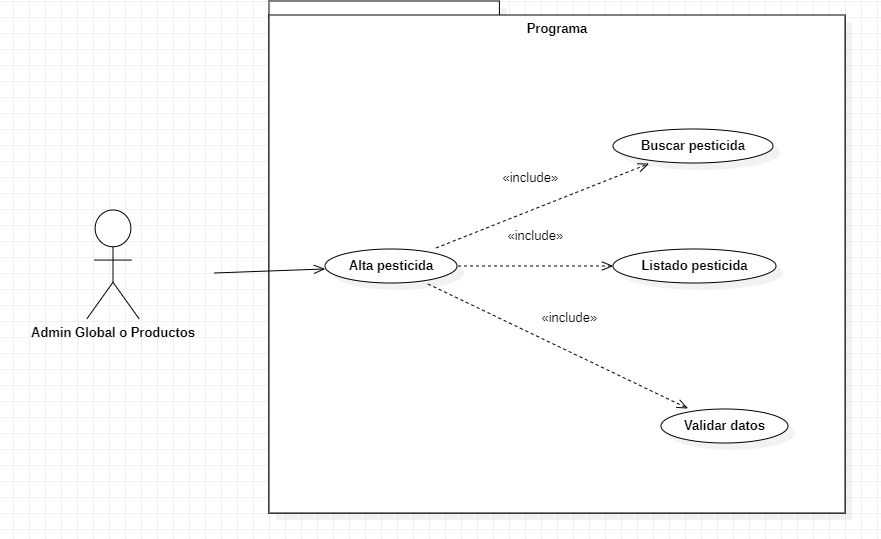
**Validaciones:** El sistema verifica que los campos no estén vacíos y que el fertilizante exista en la base de datos.

**Frecuencia de uso:** Baja.

**ABM pesticida**



**Alta pesticida**



**Identificador:** CU 047

**Nombre del CU:** Alta pesticida

**Actores:** Admin global/Admin de productos

**Descripción:** Permite dar de alta un pesticida.

**Referencia a Requerimientos Funcionales: 15**

**Precondición:** El admin debe estar con la sesión iniciada.

**Desencadenante:** El admin decide agregar un pesticida.

**Curso normal:**

1. El admin ingresa al listado de pesticidas.
2. El sistema redirige al admin al listado de pesticidas.
3. El admin ingresa a la función que dice “Agregar un pesticida”.
4. El sistema redirige al admin a un formulario para agregar un pesticida.
5. El admin ingresa los datos.
6. El sistema verifica que los campos no estén vacíos.
7. El sistema verifica que el nuevo pesticida no exista en la base de datos.
8. El sistema ingresa el nuevo pesticida a la base de datos.
9. El sistema redirige a la página de gestión de pesticidas y se actualiza el listado.
10. Fin del caso de uso.

**Curso alternativo:**

6.1. El sistema detecta un campo vacío e informa al admin.

6.1.1. El admin completa el campo vacío y continua con el flujo normal.

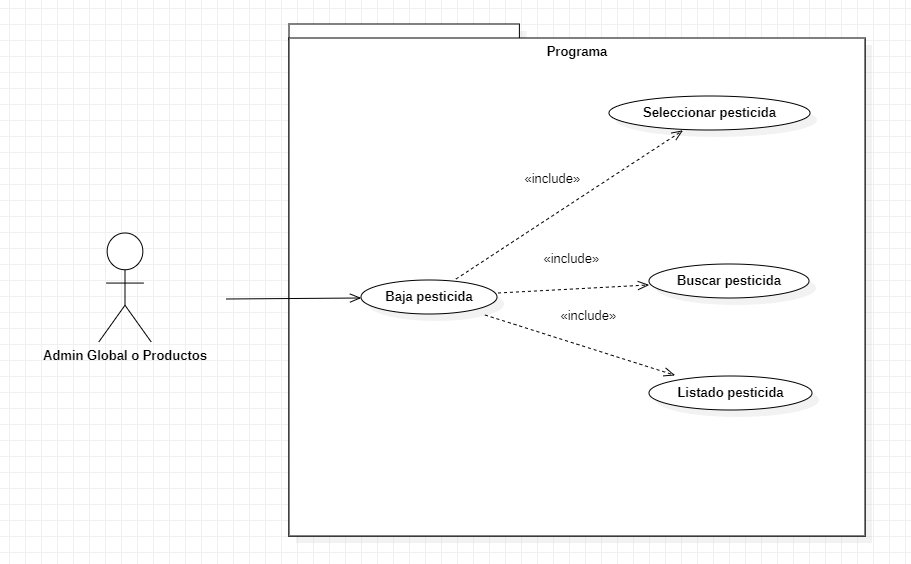
7.1. El sistema detecta que el nuevo pesticida ya existe en la base de datos y avisa a través de un mensaje al admin.

**Postcondición:**  Se agrega un nuevo pesticida al sistema.

**Validaciones:** El sistema verifica que los campos no estén vacíos y que el pesticida no exista en la base de datos.

**Frecuencia de uso: Baja**.

**Baja pesticida**



**Identificador:** CU 048

**Nombre del CU:** Baja pesticida

**Actores:** Admin global/Admin de productos

**Descripción:** Permite eliminar un pesticida del sistema.

**Referencia a Requerimientos Funcionales: 15**

**Precondición:** El admin debe estar con la sesión iniciada y debe existir algún pesticida en el sistema.

**Desencadenante:** El admin decide eliminar un pesticida.

**Curso normal:**

1. El admin ingresa al listado de pesticidas.
2. El sistema redirige al admin al listado de pesticidas.
3. El admin selecciona un pesticida para eliminar.
4. El sistema solicita una confirmación de baja.
5. El admin confirma la baja.
6. El sistema verifica que el pesticida exista en la base de datos.
7. El software verifica que el pesticida no este asociado a ningún producto.
8. Si existe y no está asociado a ningún producto, el pesticida es eliminado.
9. El sistema recarga la página actualizando el listado.
10. Fin del caso de uso.

**Curso alternativo:**

5.1. El admin cancela la baja, dejando el caso de uso sin efecto.

6.1. El sistema detecta que el pesticida no existe e informa al admin.

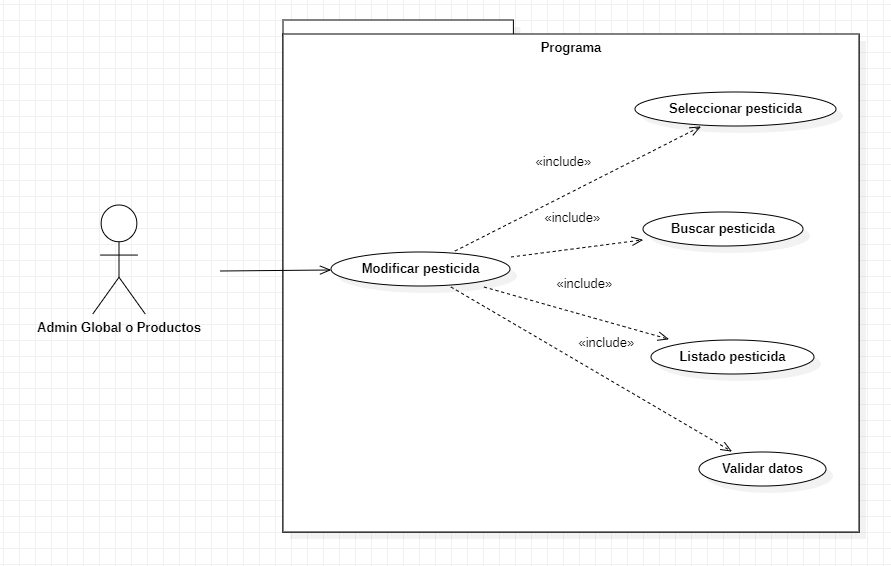
7.1. El sistema detecta que el pesticida está asociado a unos productos, por lo que el sistema avisa que se requiere eliminar estos para continuar con la eliminación del pesticida.

**Postcondición:** Se elimina un pesticida del sistema.

**Validaciones:** El sistema verifica que el pesticida exista en la base de datos y que no esté asociado a ningún producto para eliminarlo.

**Frecuencia de uso:** Baja.

**Modificar pesticida**



**Identificador:** CU 049

**Nombre del CU:** Modificar pesticida

**Actores:** Admin global/Admin de productos

**Descripción:** Permite modificar un pesticida del sistema.

**Referencia a Requerimientos Funcionales: 15**

**Precondición:** El admin debe estar con la sesión iniciada y debe existir algún pesticida en el sistema.

**Desencadenante:** El admin decide modificar un pesticida

**Curso normal:**

1. El admin ingresa al listado de pesticidas.
2. El sistema redirige al admin al listado de los pesticidas.
3. El admin selecciona un pesticida para modificar.
4. El sistema redirige al admin a un formulario con los datos del pesticida.
5. El admin modifica los datos y finaliza el proceso.
6. El sistema solicita una confirmación de modificación.
7. El admin confirma la modificación.
8. El sistema verifica que no haya algún campo vacío.
9. El sistema verifica que el pesticida exista en la base de datos.
10. El sistema actualiza el pesticida en la base de datos.
11. El sistema redirige al admin al listado de pesticidas actualizándolo.
12. Fin del caso de uso.

**Curso alternativo:**

7.1. El admin cancela la modificación, dejando el caso de uso sin efecto.

8.1. El sistema detecta un campo vacío e informa al admin.

8.1.1. El admin completa el campo vacío y continua con el flujo normal.

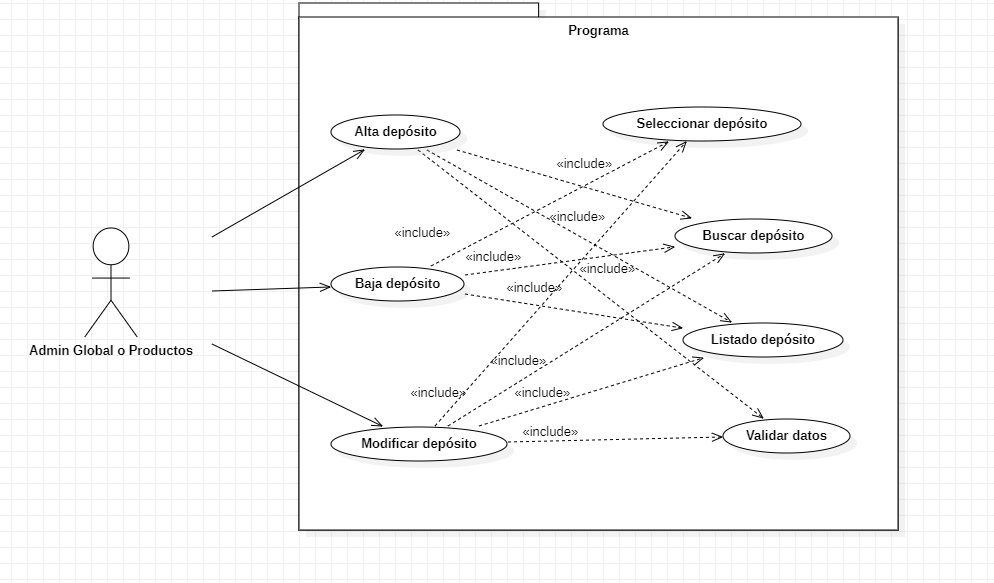
9.1. El sistema detecta que el pesticida no existe e informa al admin.

**Postcondición:** Se modifica un pesticida en el sistema.

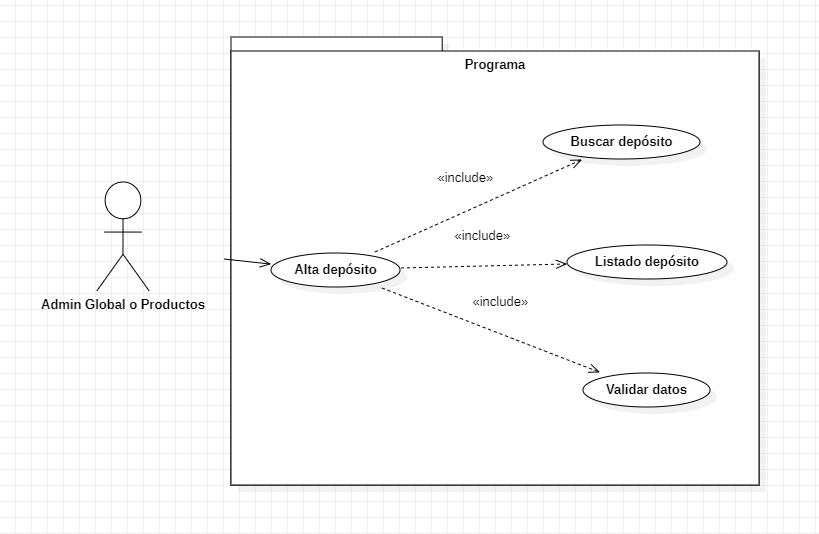
**Validaciones:** El sistema verifica que los campos no estén vacíos y que el pesticida exista en la base de datos.

**Frecuencia de uso:** Baja.

**ABM depósito**



**Alta Deposito**



**Identificador:** CU 50

**Nombre del CU:** Alta depósito

**Actores:** Admin global/Admin de productos

**Descripción:** Permite dar de alta un depósito.

**Referencia a Requerimientos Funcionales: 15**

**Precondición:** El admin debe estar con la sesión iniciada.

**Desencadenante:** El admin decide agregar un depósito.

**Curso normal:**

1. El admin ingresa al listado de depósitos.
2. El sistema redirige al admin al listado de depósitos.
3. El admin ingresa a la función que dice “Agregar un depósito”.
4. El sistema redirige al admin a un formulario para agregar un depósito.
5. El admin ingresa los datos.
6. El sistema verifica que los campos no estén vacíos.
7. El sistema verifica que el nuevo depósito no exista en la base de datos.
8. El sistema ingresa el nuevo depósito a la base de datos.
9. El sistema redirige a la página de gestión de depósitos y se actualiza el listado.
10. Fin del caso de uso.

**Curso alternativo:**

6.1. El sistema detecta un campo vacío e informa al admin.

6.1.1. El admin completa el campo vacío y continua con el flujo normal.

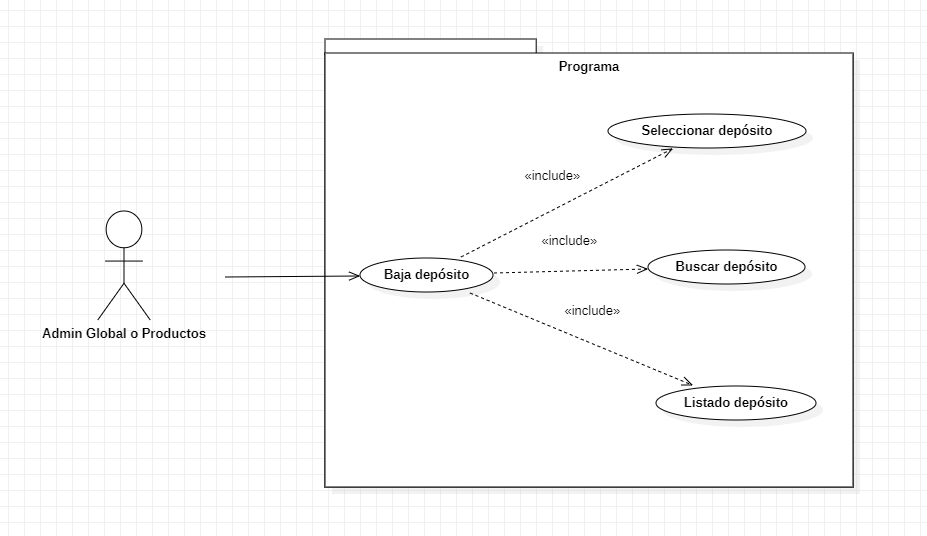
7.1. El sistema detecta que el nuevo depósito ya existe en la base de datos y avisa a través de un mensaje al admin.

**Postcondición:**  Se agrega un nuevo depósito al sistema.

**Validaciones:** El sistema verifica que los campos no estén vacíos y que el depósito no exista en la base de datos.

**Frecuencia de uso: Baja**.

**Baja depósito**



**Identificador:** CU 051

**Nombre del CU:** Baja depósito

**Actores:** Admin global/Admin de productos

**Descripción:** Permite eliminar un depósito del sistema.

**Referencia a Requerimientos Funcionales: 15**

**Precondición:** El admin debe estar con la sesión iniciada y debe existir algún depósito en el sistema.

**Desencadenante:** El admin decide eliminar un depósito.

**Curso normal:**

1. El admin ingresa al listado de depósitos.
2. El sistema redirige al admin al listado de depósitos.
3. El admin selecciona un depósito para eliminar.
4. El sistema solicita una confirmación de baja.
5. El admin confirma la baja.
6. El sistema verifica que el depósito exista en la base de datos.
7. El software verifica que el depósito no este asociado a ningún producto.
8. Si existe y no está asociado a ningún producto, el depósito es eliminado.
9. El sistema recarga la página actualizando el listado.
10. Fin del caso de uso.

**Curso alternativo:**

5.1. El admin cancela la baja, dejando el caso de uso sin efecto.

6.1. El sistema detecta que el depósito no existe e informa al admin.

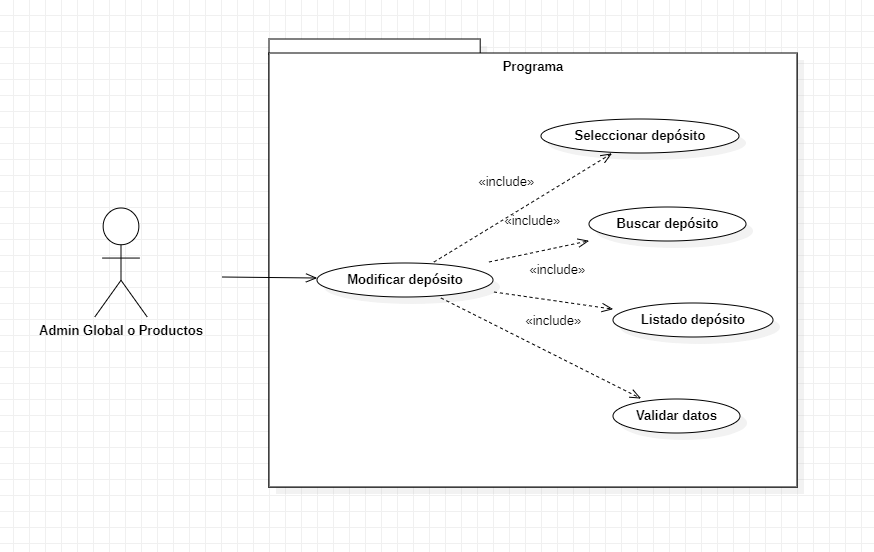
7.1. El sistema detecta que el depósito está asociado a unos productos, por lo que el sistema avisa que se requiere eliminar estos para continuar con la eliminación del depósito.

**Postcondición:** Se elimina un depósito del sistema.

**Validaciones:** El sistema verifica que el depósito exista en la base de datos y que no esté asociado a ningún producto para eliminarlo.

**Frecuencia de uso:** Baja.

**Modificar depósito**



**Identificador:** CU 052

**Nombre del CU:** Modificar depósito.

**Actores:** Admin global/Admin de productos

**Descripción:** Permite modificar un depósito del sistema.

**Referencia a Requerimientos Funcionales: 15**

**Precondición:** El admin debe estar con la sesión iniciada y debe existir algún depósito en el sistema.

**Desencadenante:** El admin decide modificar un depósito.

**Curso normal:**

1. El admin ingresa al listado de depósitos.
2. El sistema redirige al admin al listado de los depósitos.
3. El admin selecciona un depósito para modificar.
4. El sistema redirige al admin a un formulario con los datos del depósito.
5. El admin modifica los datos y finaliza el proceso.
6. El sistema solicita una confirmación de modificación.
7. El admin confirma la modificación.
8. El sistema verifica que no haya algún campo vacío.
9. El sistema verifica que el depósito exista en la base de datos.
10. El sistema actualiza el depósito en la base de datos.
11. El sistema redirige al admin al listado de depósitos actualizándolo.
12. Fin del caso de uso.

**Curso alternativo:**

7.1. El admin cancela la modificación, dejando el caso de uso sin efecto.

8.1. El sistema detecta un campo vacío e informa al admin.

8.1.1. El admin completa el campo vacío y continua con el flujo normal.

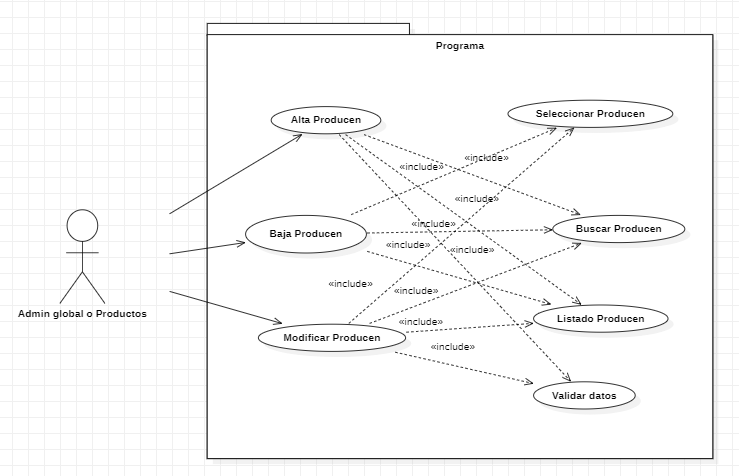
9.1. El sistema detecta que el depósito no existe e informa al admin.

**Postcondición:** Se modifica un depósito en el sistema.

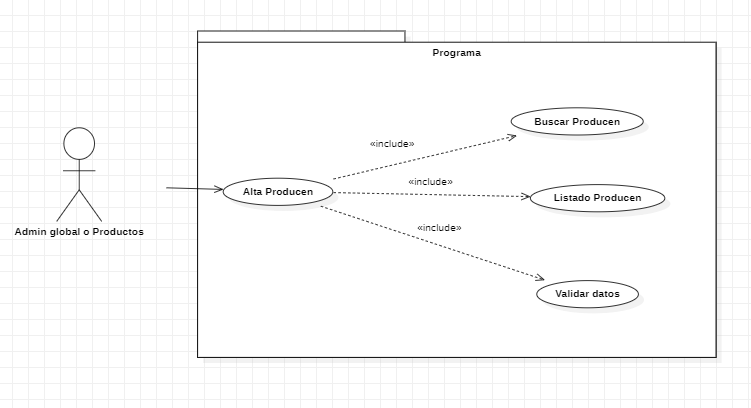
**Validaciones:** El sistema verifica que los campos no estén vacíos y que el depósito exista en la base de datos.

**Frecuencia de uso:** Baja.

ABM Producen



Alta Producen



**Identificador:** CU 53

**Nombre del CU:** Alta Producen.

**Actores:** Admin global/Admin de productos.

**Descripción:** Permite dar de alta a producen (relación N-N entre granjas y productos).

**Referencia a Requerimientos Funcionales: 16**

**Precondición:** El admin debe estar con la sesión iniciada y debe existir alguna granja y productos registrados en el sistema.

**Desencadenante:** El admin decide agregar un producen.

**Curso normal:**

1. El admin ingresa al listado de producen.
2. El sistema redirige al admin al listado de producen.
3. El admin ingresa a la función que dice “Agregar producen”.
4. El sistema redirige al admin a un formulario para agregar un producen.
5. El admin ingresa los datos requeridos.
6. El sistema verifica que los campos no estén vacíos.
7. El sistema verifica que el nuevo producen no exista, y que la granja y producto que seran asignados, esten registrados en la base de datos.
8. El sistema ingresa el nuevo producen a la base de datos.
9. El sistema redirige a la página de gestión de producen y se actualiza el listado.
10. Fin del caso de uso.

**Curso alternativo:**

6.1. El sistema detecta un campo vacío e informa al admin.

6.1.1. El admin completa el campo vacío y continua con el flujo normal.

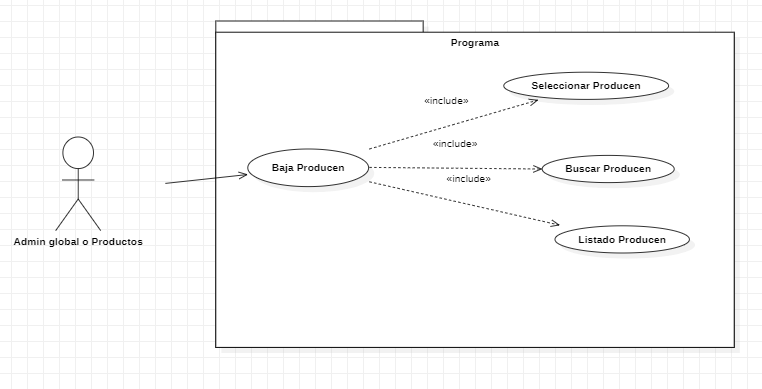
7.1. El sistema detecta que el nuevo producen ya existe en la base de datos o que la granja o producto no existen en la base de datos y avisa a través de un mensaje al admin.

**Postcondición:**  Se agrega un nuevo producen al sistema.

**Validaciones:** El sistema verifica que los campos no estén vacíos y que el producen no exista , y la granja y productos esten registrados en la base de datos.

**Frecuencia de uso: Baja**.

Baja Producen



**Identificador:** CU 054

**Nombre del CU:** Baja Producen

**Actores:** Admin global/Admin de productos

**Descripción:** Permite eliminar a producen (relación N-N entre granjas y productos) del sistema.

**Referencia a Requerimientos Funcionales: 16**

**Precondición:** El admin debe estar con la sesión iniciada y debe existir algun producen

**Desencadenante:** El admin decide eliminar un producen.

**Curso normal:**

1. El admin ingresa al listado de producen.
2. El sistema redirige al admin al listado de producen.
3. El admin selecciona un producen para eliminar.
4. El sistema solicita una confirmación de baja.
5. El admin confirma la baja.
6. El sistema verifica que el producen exista en la base de datos.
7. Si existe, todas las referencias donde esta el producen son eliminadas.
8. El sistema recarga la página actualizando el listado.
9. Fin del caso de uso.

**Curso alternativo:**

5.1. El admin cancela la baja, dejando el caso de uso sin efecto.

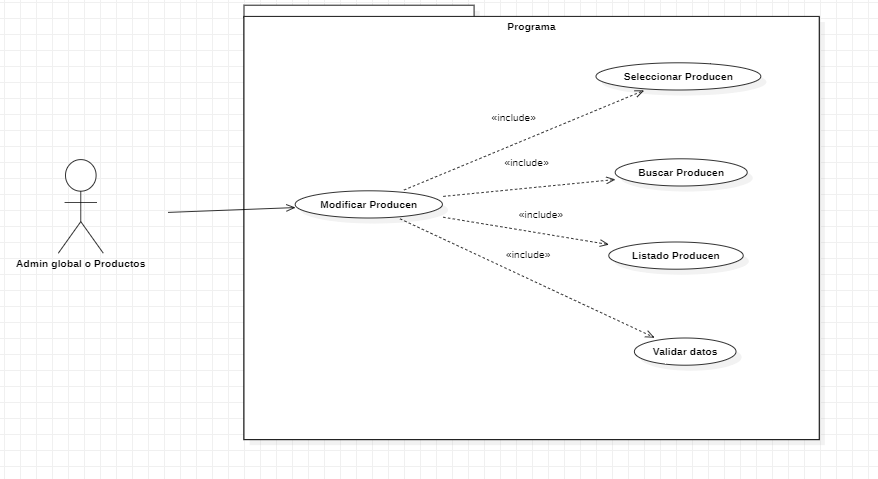
6.1. El sistema detecta que el producen no existe e informa al admin.

7**Postcondición:** Se elimina un producen del sistema.

**Validaciones:** El sistema verifica que el producen exista en la base de datos.

**Frecuencia de uso:** Baja.

Modificar Producen



**Identificador:** CU 055

**Nombre del CU:** Modificar Producen.

**Actores:** Admin global/Admin de productos

**Descripción:** Permite modificar un producen (relación N-N entre granjas y productos) del sistema.

**Referencia a Requerimientos Funcionales: 16**

**Precondición:** El admin debe estar con la sesión iniciada y debe existir algún producen en el sistema.

**Desencadenante:** El admin decide modificar un producen.

**Curso normal:**

1. El admin ingresa al listado de producen.
2. El sistema redirige al admin al listado de producen.
3. El admin selecciona un producen para modificar.
4. El sistema redirige al admin a un formulario con los datos del producen.
5. El admin modifica los datos y finaliza el proceso.
6. El sistema solicita una confirmación de modificación.
7. El admin confirma la modificación.
8. El sistema verifica que no haya algún campo vacío.
9. El sistema verifica que el producen exista en la base de datos.
10. El sistema actualiza todas las referencias a ese produce en la base de datos.
11. El sistema redirige al admin al listado de producen actualizándolo.
12. Fin del caso de uso.

**Curso alternativo:**

7.1. El admin cancela la modificación, dejando el caso de uso sin efecto.

8.1. El sistema detecta un campo vacío e informa al admin.

8.1.1. El admin completa el campo vacío y continua con el flujo normal.

9.1. El sistema detecta que el producen no existe e informa al admin.

**Postcondición:** Se modifica un producen en el sistema.

**Validaciones:** El sistema verifica que los campos no estén vacíos y que el producen exista en la base de datos.

**Frecuencia de uso:** Baja.

## Diccionario de clases

|  |  |  |
| --- | --- | --- |
| **Nombre** | Persona | |
| **Descripción** | Clase base | |
| **Atributos** | | |
| **Nombre** | **Tipo de dato** | **Visibilidad** |
| idPersona | Int | private |
| nombre | string | private |
| apellido | string | private |
| email | string | private |
| telefono | string | private |
| fchNacimiento | string | private |

|  |  |  |
| --- | --- | --- |
| **Nombre** | Persona: Cliente | |
| **Descripción** | Administra clientes heredando los atributos de la clase persona | |
| **Atributos** | | |
| **Nombre** | **Tipo de dato** | **Visibilidad** |
| usuario | string | private |
| contrasena | string | private |
| direccion | string | private |

|  |  |  |
| --- | --- | --- |
| **Nombre** | Persona: Admin | |
| **Descripción** | Administra admins heredando los atributos de la clase persona | |
| **Atributos** | | |
| **Nombre** | **Tipo de dato** | **Visibilidad** |
| usuario | string | private |
| contraseña | string | private |
| tipoDeAdmin | string | private |

|  |  |  |
| --- | --- | --- |
| **Nombre** | Persona: Camionero | |
| **Descripción** | Administra camioneros heredando los atributos de la clase persona | |
| **Atributos** | | |
| **Nombre** | **Tipo de dato** | **Visibilidad** |
| cedula | string | Private |
| disponible | string | Private |
| fchManejo | string | Private |

|  |  |  |
| --- | --- | --- |
| **Nombre** | Camión | |
| **Descripción** | Administra camiones heredando los atributos de la clase persona | |
| **Atributos** | | |
| **Nombre** | **Tipo de dato** | **Visibilidad** |
| idCamion | int | private |
| marca | string | private |
| modelo | string | private |
| carga | double | private |
| disponible | boolean | private |

|  |  |  |
| --- | --- | --- |
| **Nombre** | Granja | |
| **Descripción** | Administra Granjas | |
| **Atributos** | | |
| **Nombre** | **Tipo de dato** | **Visibilidad** |
| idGranja | int | private |
| nombre | string | private |
| ubicación | string | private |
| idCliente | int | private |

|  |  |  |
| --- | --- | --- |
| **Nombre** | Fertilizante | |
| **Descripción** | Administra fertilizantes | |
| **Atributos** | | |
| **Nombre** | **Tipo de dato** | **Visibilidad** |
| idFertilizante | int | private |
| nombre | string | private |
| origen | string | private |
| compQuimica | string | private |
| ph | short | private |
| impacto | string | private |

|  |  |  |
| --- | --- | --- |
| **Nombre** | Pesticida | |
| **Descripción** | Administra pesticidas | |
| **Atributos** | | |
| **Nombre** | **Tipo de dato** | **Visibilidad** |
| idPesticida | int | private |
| nombre | string | private |
| tipo | string | private |
| toxicidad | string | private |
| resistencia | string | private |
| impacto | string |  |

|  |  |  |
| --- | --- | --- |
| **Nombre** | Producto | |
| **Descripción** | Administra productos | |
| **Atributos** | | |
| **Nombre** | **Tipo de dato** | **Visibilidad** |
| idProducto | int | private |
| nombre | string | private |
| tipo | string | private |
| tipoVenta | string | private |

|  |  |  |
| --- | --- | --- |
| **Nombre** | Produce | |
| **Descripción** | Relacion entre Producto y Granja | |
| **Atributos** | | |
| **Nombre** | **Tipo de dato** | **Visibilidad** |
| idGranja | int | private |
| idProducto | int | private |
| fchProduccion | string | private |
| stock | int | private |
| idDeposito | int | private |
| precio | double | private |

|  |  |  |
| --- | --- | --- |
| **Nombre** | Pedido | |
| **Descripción** | Administra pedidos | |
| **Atributos** | | |
| **Nombre** | **Tipo de dato** | **Visibilidad** |
| idPedido | int | private |
| idCliente | int | private |
| estado | string | private |
| fechaEntre | DateTime | private |
| fechaEspe | DateTime | private |
| retraso | string | private |
| costo | double | private |
| idCamion | int | private |
| idCamionero | Int | private |

|  |  |  |
| --- | --- | --- |
| **Nombre** | Produce\_Pedido | |
| **Descripción** | Relación entre Produce y Pedido | |
| **Atributos** | | |
| **Nombre** | **Tipo de dato** | **Visibilidad** |
| idPedido | int | private |
| idProducto | int | private |
| idGranja | int | private |
| fchProduccion | string | private |

|  |  |  |
| --- | --- | --- |
| **Nombre** | Produce\_Ferti | |
| **Descripción** | Relación entre Produce y Fertilizante | |
| **Atributos** | | |
| **Nombre** | **Tipo de dato** | **Visibilidad** |
| idFertilizante | int | private |
| idProducto | int | private |
| idGranja | int | private |
| fchProduccion | string | private |
| cantidad | string | private |

|  |  |  |
| --- | --- | --- |
| **Nombre** | Produce\_Pesti | |
| **Descripción** | Relación entre Produce y Pesticida | |
| **Atributos** | | |
| **Nombre** | **Tipo de dato** | **Visibilidad** |
| idPesticida | int | private |
| idProducto | int | private |
| idGranja | int | private |
| fchProduccion | string | private |
| cantidad | string | private |

|  |  |  |
| --- | --- | --- |
| **Nombre** | Deposito | |
| **Descripción** | Administra depositos | |
| **Atributos** | | |
| **Nombre** | **Tipo de dato** | **Visibilidad** |
| idDeposito | Int | private |
| capacidad | string | private |
| ubicación | string | private |
| temperatura | short | private |
| condiciones | string | private |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre** | Controladora | | |
| **Descripción** | Controladora principal | | |
|  | | | |
| **Nombre** | **Tipo de dato** | | **Visibilidad** |
| controladoraPersona | ControladoraPersona | | private |
| controladoraItem | ControladoraItem | | private |
| instancia | Controladora | | private |
| **Métodos** | | | |
| **Nombre** | **Tipo de dato** | **Descripción** | **Visibilidad** |
| obtenerInstancia | Controladora | Permite solo instanciar una vez la clase Controladora | public |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre** | ControladoraPersona | | |
| **Descripción** | Controladora de personas | | |
| **Atributos** | | | |
| **Nombre** | **Tipo de dato** | **Visibilidad** | |
| instancia | ControladoraPersona | private | |
| **Métodos** | | | |
| **Nombre** | **Tipo de dato** | **Descripción** | **Visibilidad** |
| obtenerInstancia | ControladoraPersona | Permite solo instanciar una vez la clase ControladoraPersona | public |
| bajaCliente | bool | Eliminar un cliente | public |
| login | bool | Permite ingresar al sistema | public |
| buscarCliente | Cliente | Permite buscar un cliente | public |
| listCliente | List <Cliente> | Devuelve la lista de clientes | public |
| cambiarCon | bool | Permite cambiar la contraseña | Public |
| encriptarCont | bool | Permite encriptar la contraseña | public |
| chatLin | String | Permite el chat entre usuarios y admins | public |
| ntfChat | bool | Notifica un nuevo mensaje del chat entre admin y usuario | public |
| registrarse | bool | Permite a un usuario registrarse | public |
| altaAdmin | bool | Permite dar de alta un nuevo Admin | public |
| bajaAdmin | bool | Permite eliminar un Admin | public |
| modAdmin | bool | Permite modificar un Admin | public |
| buscarAdmin | Admin | Permite buscar un Admin | public |
| listAdmin | List<Admin> | Devuelve la lista de Admins | public |
| altaCamionero | bool | Permite añadir un nuevo camionero | public |
| bajaCamionero | bool | Permite eliminar un camionero | public |
| modCamionero | bool | Permite modificar un camionero | public |
| buscarCamionero | Camionero | Permite buscar un camionero | public |
| listCamionero | List<Camionero> | Devuelve la lista de Camioneros | public |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre** | ControladoraItem | | |
| **Descripción** | Controladora de items | | |
| **Atributos** | | | |
| **Nombre** | **Tipo de dato** | **Visibilidad** | |
| instancia | ControladoraItem | private | |
| **Métodos** | | | |
| **Nombre** | **Tipo de dato** | **Descripción** | **Visibilidad** |
| obtenerInstancia | ControladoraItem | Permite solo instanciar una vez la clase ControladoraItem | public |
| hacerPedido | bool | Permite hacer un pedido al cliente | public |
| bajaPedido | bool | Permite dar de baja un pedido | public |
| modPedido | bool | Permite modificar un pedido | public |
| buscarPedido | Pedido | Permite buscar un pedido | public |
| listPedido | List<Pedido> | Devuelve la lista de pedidos | public |
| finPedido | bool | Permite marcar un pedido como finalizado | public |
| confPedido | bool | Permite confirmar un pedido | public |
| asigPedido | bool | Permite asignar un camión y camionero al pedido |  |
| verPedidos | List<pedido> | Permite ver los pedidos actuales al cliente | public |
| verPedidosAnt | List<pedido> | Permite ver los pedios anteriores al cliente | public |
| verGastos | Double | Permite ver los gastos totales de los pedidos anteriores | public |
| notificarCamb | string | Notifica cambio de estado a un cliente | public |
| altaProducto | bool | Permite añadir un producto | public |
| bajaProducto | bool | Permite eliminar un producto | public |
| modProducto | bool | Permite modificar un producto | public |
| buscarProducto | Producto | Permite buscar un producto | public |
| listProducto | List<Producto> | Devuelve la lista de productos | public |
| ntfPorStock | string | Notifica a los admins globales y de productos por falta de stock de un producto | public |
| ntfPorCad | string | Notifica a los admins globales y de productos por que un producto esta caducado | public |
| ntfPorRet | string | Permite a los admins globales y de pedidos notificar al cliente un retraso de su pedido | public |
| altaCamion | bool | Permite añadir un nuevo camión | public |
| bajaCamion | bool | Permite eliminar un camión | public |
| modCamion | bool | Permite modificar un camión | public |
| buscarCamion | Camion | Permite buscar un camión | public |
| listCamion | List<Camion> | Devuelve la lista de camiones | public |
| altaGranja | bool | Permite añadir una nueva granja | public |
| bajaGranja | bool | Permite eliminar una granja | public |
| modGranja | bool | Permite modificar una granja | public |
| buscarGranja | Granja | Permite buscar una granja | public |
| listGranja | List<Granja> | Devuelve la lista de granjas | public |
| altaFert | bool | Permite añadir un nuevo fertilizante | public |
| bajaFert | bool | Permite eliminar un fertilizante | public |
| modFert | bool | Permite modificar un fertilizante | public |
| buscarFert | Fertilizante | Permite buscar un fertilizante | public |
| listFert | <Fertilizante> | Devuelve la lista de fertilizantes | public |
| altaPest | bool | Permite añadir un nuevo pesticida | public |
| bajaPest | bool | Permite eliminar un pesticida | public |
| modPest | bool | Permite modificar un pesticida | public |
| buscarPest | Pesticida | Permite buscar un pesticida | public |
| listPest | List<Pesticida> | Devuelve la lista de pesticidas | public |
| altaDesp | bool | Permite añadir un depósito | public |
| bajaDesp | bool | Permite eliminar un depósito | public |
| modDeps | bool | Permite modificar un depósito | public |
| buscarDeps | Deposito | Permite buscar un depósito | public |
| listDeps | List<deposito> | Devuelve la lista de depósitos | public |