



Pontificia Universidad Católica de Valparaíso

Facultad de Ingeniería

Escuela de Ingeniería Informática

Ingeniería de Ejecución en Informática

SOFTWARE DE CONTROL DEL PROCESO PRODUCTIVO PARA VIVERO ECOGREEN

Informe final del Proyecto para optar al Título profesional de
Ingeniero de Ejecución en Informática

Autores:

Eugenio Andrés Suarez Maturana - Marcos Américo Baeza Contreras

Profesor Guía:

Iván Mercado Bermúdez

Profesor Co-referente: Aldo Migliaro Osorio

Diciembre 2008

Agradecimientos

Agradezco a mi familia y amigos por brindarme todo el apoyo, comprensión y confianza que mostraron durante esta importante etapa de mi vida.

A toda aquella persona que en su momento a pesar de sus propias dudas y problemas me brindo seguridad y encendió en mí el deseo de ser una mejor persona y mejorar mis destrezas, sembrando así la semilla de mi futuro éxito.

Eternamente Agradecido

Marcos.

Mama gracias por levantarte cada mañana junto a mí y estudiar a mi lado, papa gracias por trabajar inagotablemente para yo poder educarme, hermana gracias por tu apoyo y la confianza que me demostraste cuando pocos lo hacían y gracias a todas las demás personas claves de mi vida que me dieron fuerzas cuando quería caer . Gracias a todos ustedes estoy aquí, a puertas de ser profesional.

Eugenio.

Resumen

El Contenido de las siguientes páginas reflejan el documento de la memoria realizada por los alumnos de la escuela de ingeniería informática de la Pontificia Universidad Católica de Valparaíso, el Sr. Marcos Baeza Contreras y el Sr. Eugenio Suarez Maturana.

El presente informe muestra como se llevo a cabo el software de control del proceso productivo para la empresa Ecogreen, interviniendo las áreas de bodega, reserva, clientes, ventas, empleados y producción que esta tiene.

Se considera el estudio realizado sobre el proceso de producción de un vivero, elecciones sobre el paradigma, tecnología, metodología y herramientas, así como también se muestra la problemática a resolver y como se debería solucionar.

Abstract

The content of the following pages shows the written paper document developed by the students of IT Engineering from the Pontificia Universidad Catolica de Valparaiso, Mr. Marcos Baeza and Mr. Eugenio Suárez.

This report shows the way the productive process control software for Ecogreen enterprise was carried out, interceding the storage, supplies, customers, sales, employees and production areas.

The study on the production process in a nursery, paradigm choices, technology, methodology and tools is considered. Likewise, the report shows the issue to be solved and the way to solve it.

1.- Introducción

El siguiente documento de memoria, contiene el estudio de la problemática actual que presenta el ciclo de producción del vivero Ecogreen. Y el posterior análisis de todos los componentes, áreas de trabajo y sus actividades diarias. Esto junto con el estudio y valiosa colaboración del cliente, darán vida al proyecto denominado “Software de control del proceso productivo para vivero Ecogreen”.

Esta empresa requiere principalmente un sistema que sea capaz de controlar, registrar e informar sobre los procesos de asignación de trabajos diarios a sus empleados y establecer un control de cada uno de ellos, tanto en la forma como se llevarán a cabo sus labores y de los productos necesarios para cada labor, ya que, es de vital importancia tener un registro de los insumos (fertilizantes o materiales de trabajo) que se utilizan diariamente en las faenas. Además se debe asegurar el control y completo registro de las ventas ya que Ecogreen desea implementar un sistema de bono de venta, para esto es de suma importancia tener en cuenta todas las etapas por las cuales paso el producto hasta llegar ser vendida, otro punto importante es el control de insumos y material de trabajo de la bodega, para poder controlar las responsabilidades en caso de cualquier tipo de destrucción o pérdida del insumo.

El sistema deberá permitir la automatización del control de la producción, dándole seguridad a las transacciones, como también una buena organización del vivero. Creando un sistema que controle los trabajos realizados sobre las distintas especies de plantas y los procesos que intervienen en ellos.

El contenido de este informe comienza con la definición de los objetivos generales de este proyecto, para luego llegar a una descripción del funcionamiento actual del vivero Ecogreen y los distintos caminos que se tomaron para poder establecer una solución a la problemática que posteriormente serán la base del “Software de control del proceso productivo para vivero Ecogreen”.

El propósito de este documento es dejar registro de cómo se analizó el problema, se desarrolló la solución a la problemática antes planteada y el desarrollo de las pruebas realizadas para comprobar la consistencia del sistema, detallando las funciones y el uso apropiado de este sistema, restricciones, arquitectura e interfaces. Originando que pueda ser reutilizado por la empresa, para desarrollar cualquier sistema del mismo tipo.

2.- Objetivos

Para todo proyecto informático se deben establecer ciertos puntos y metas que son importantes tanto para los usuarios como para los desarrolladores. Es así como surgen los objetivos generales y específicos del proyecto.

2.1.- Generales

Implementar un Software capaz controlar los principales pasos y flujos en la producción del vivero Ecogreen.

2.2.- Específicos

- Estudiar, modelar y formalizar el funcionamiento actual de la empresa Ecogreen.
- Proponer mejoras al proceso de trabajo actual de la empresa Ecogreen en algunas areas.
- Construir un sistema control de producción que permita mejorar el proceso de trabajo actual de la empresa Ecogreen.
- Realizar una validación del sistema construido.

Mejoras Esperadas:

- Permitir un mejor control de los procesos dentro de la empresa.
- Entregar una mejor administración de la empresa.
- Entregar un flujo de información rápido y ordenado de los datos (informes).
- Dar seguridad a la información de la empresa.
- Disminuir los costos del proceso actual de la empresa.

3.- Estado Del Arte

3.1.- Situación Actual

A continuación se describirá en detalle la empresa para la cual se está desarrollando este proyecto junto con los roles dentro de esta, el flujo actual de trabajo y las mejoras que se esperan realizar a esta.

3.1.1.- Descripción De La Empresa

Este proyecto tiene como cliente directo al Vivero ECOGREEN, el cual como toda empresa de este rubro se dedica a la producción de plantas. Ubicado en Panamericana Norte Km. 97 Hijuelas, Fono-Fax (33) 272333.

Nace en el año 1995 cuando se asocian los Señores Jorge Avalos Maya y Juan Carlos Buvinic con la intención de crear un Vivero rico en calidad tanto humana como productiva, alejándose de la visión existente de las muchas empresas existentes, en este rubro de la misma zona.

Después de varios años la empresa ha logrado posicionarse firmemente en el mercado siendo una de las líderes del sector en lo que respecta a la producción de plantas, algo que sin duda es todo un mérito.

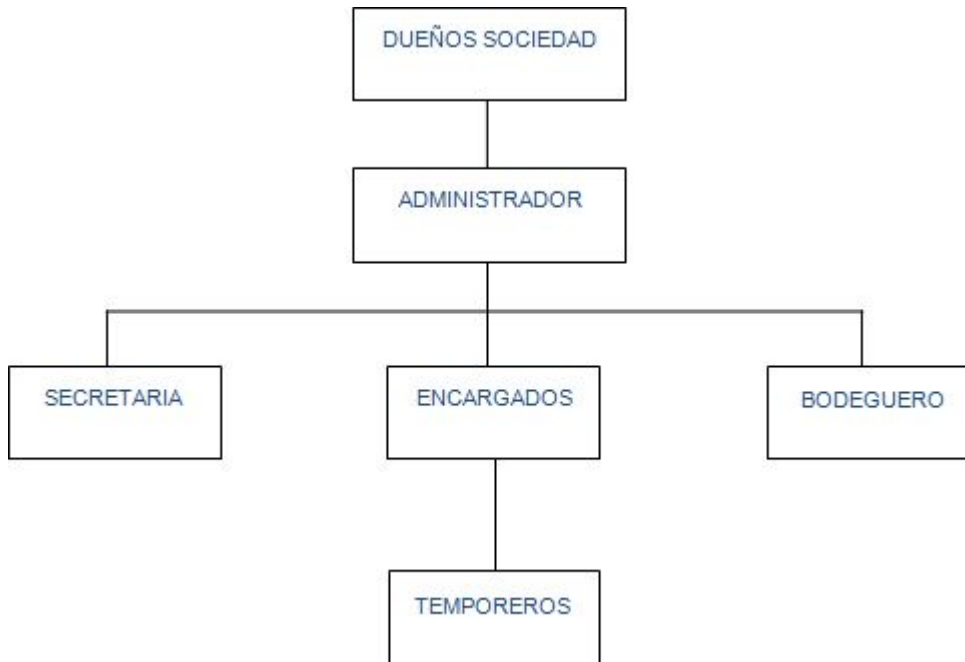
El mercado de clientes, se compone principalmente de supermercados y empresas que se dedican a la función de la reventa de plantas, siendo las personas particulares solo una pequeña parte del total de las ventas. La producción se basa principalmente en:

- Flores
- Arbustos
- Plantas interior y exterior

La empresa cuenta con un capital humano de alrededor de 25 personas, siendo en temporadas altas un número cercano a 32 y temporadas bajas 18 individuos.

3.1.2.- Organigrama

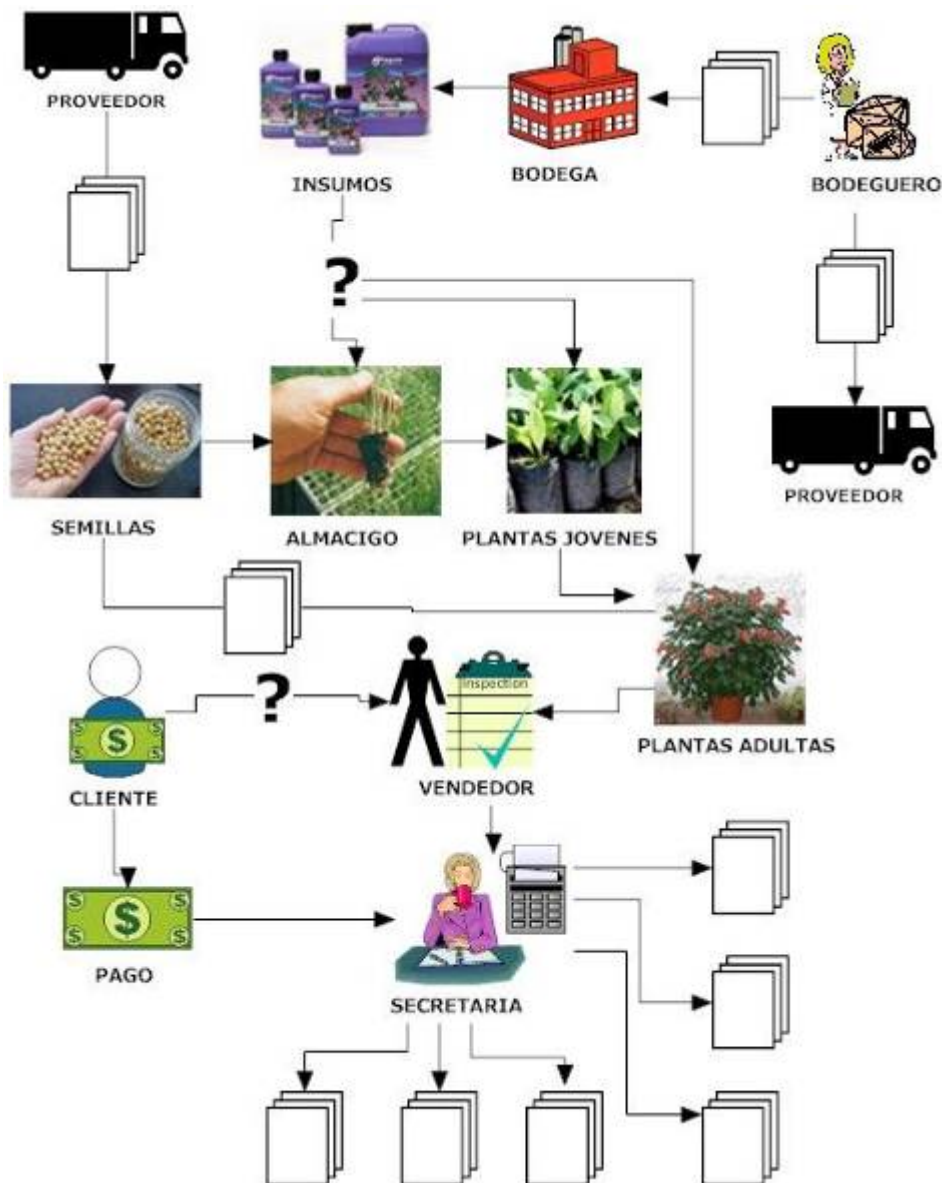
Imagen 01: Organigrama Empresa Ecogreen



Este es el Organigrama del vivero Ecogreen, En nuestro Proyecto se mantuvo un contacto directo con el administrador del vivero el cual por los años en el rubro es el completo experto en todo lo que respecta a procesos productivos.

3.1.3.- Descripción Del Flujo Actual

Imagen 02: Flujo Actual Empresa Ecogreen



La planta inicia su proceso de vida como almacigo terminando en planta adulta, esta posteriormente se utilizara para la venta o para la reproducción. Los insumos utilizados sobre los productos se adquieren de los proveedores, y se guardan en bodega. Se realiza un inventario ocasionalmente de lo que se tiene en bodega para solicitarlo a los proveedores.

Cuando se realiza una venta el cliente le pide al vendedor cierto producto, este último no maneja el stock disponible, así ambos tienen que recorrer todo el vivero para ver si la solicitud del cliente se encuentra disponible. De ser este el caso el detalle de venta se le asigna a la secretaria quien realiza finalmente la venta.

La mayoría de los procesos relacionados a la administración del vivero son débiles o bien inexistentes. En lo que se refiere a ventas y a fertilización la confianza en el empleado es fundamental, por nombrar algunos ejemplos.

Por otra parte muchos de los datos importantes son almacenados mediante papel muy fácil de extraviar o modificar.

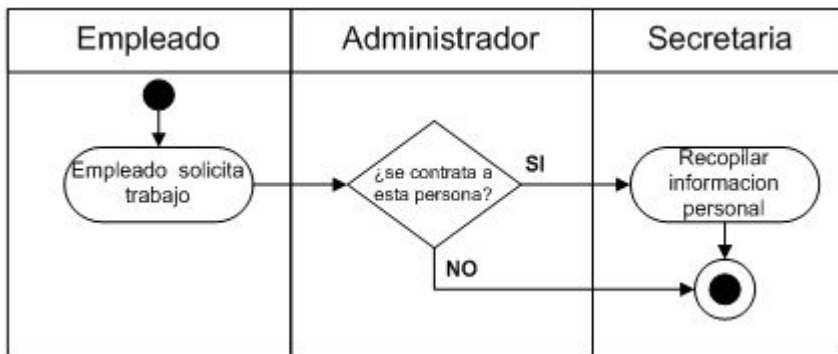
A continuación describimos el flujo actual por medio de las diferentes operaciones que realizan los actores del sistema:

3.1.3.1.- Administrador

3.1.3.1.1.- Gestionar Trabajadores

Los trabajadores o empleados del vivero son gestionados mediante la recopilación de sus respectivos contratos currículum u otros, de forma que son almacenados o guardados por medio de carpetas o archivadores.

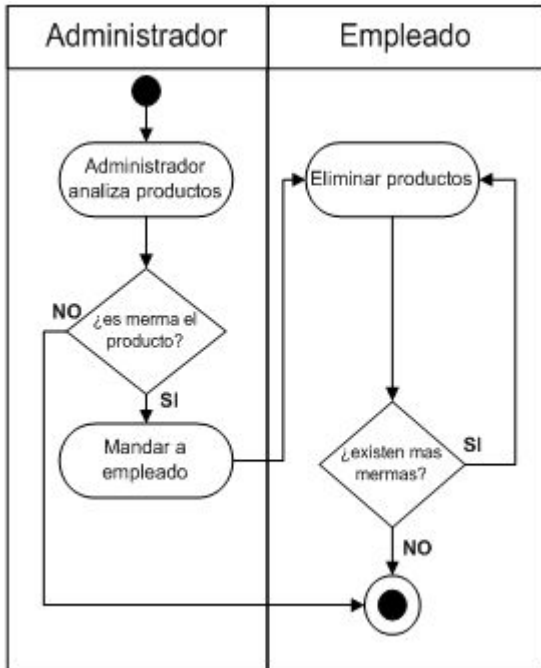
Imagen 03: Proceso control del personal Ecogreen



3.1.3.1.2.- Dar De Baja A Productos

El administrador tiene la experiencia necesaria para dar de baja a una producción específica, el sabe cuando una especie sufrió un problema en el que no puede recuperarse y así pasa a ser merma.

Imagen 04: Proceso actual dar de baja

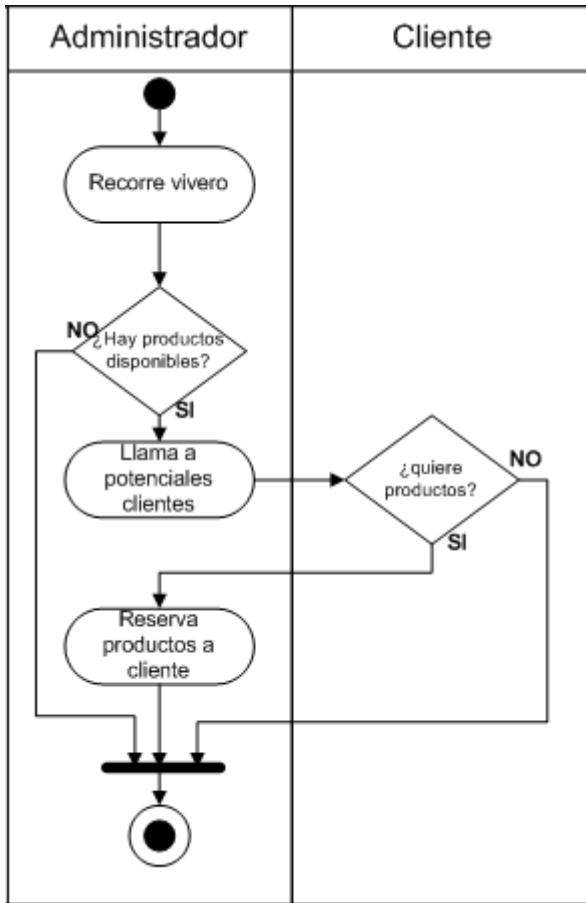


3.1.3.1.3.- Administración De Clientes

Actualmente el registro de clientes se realiza en un cuaderno, los clientes que son registrados en ese cuaderno son conocidos para al administrador, es decir, los que ya han comprado en varias ocasiones anteriormente.

Cuando se realiza una producción de una especie se llama a los clientes que pueden querer dicho producto según lo que se recuerda de su historial de compras.

Imagen 05: Proceso actual registró cliente Ecogreen

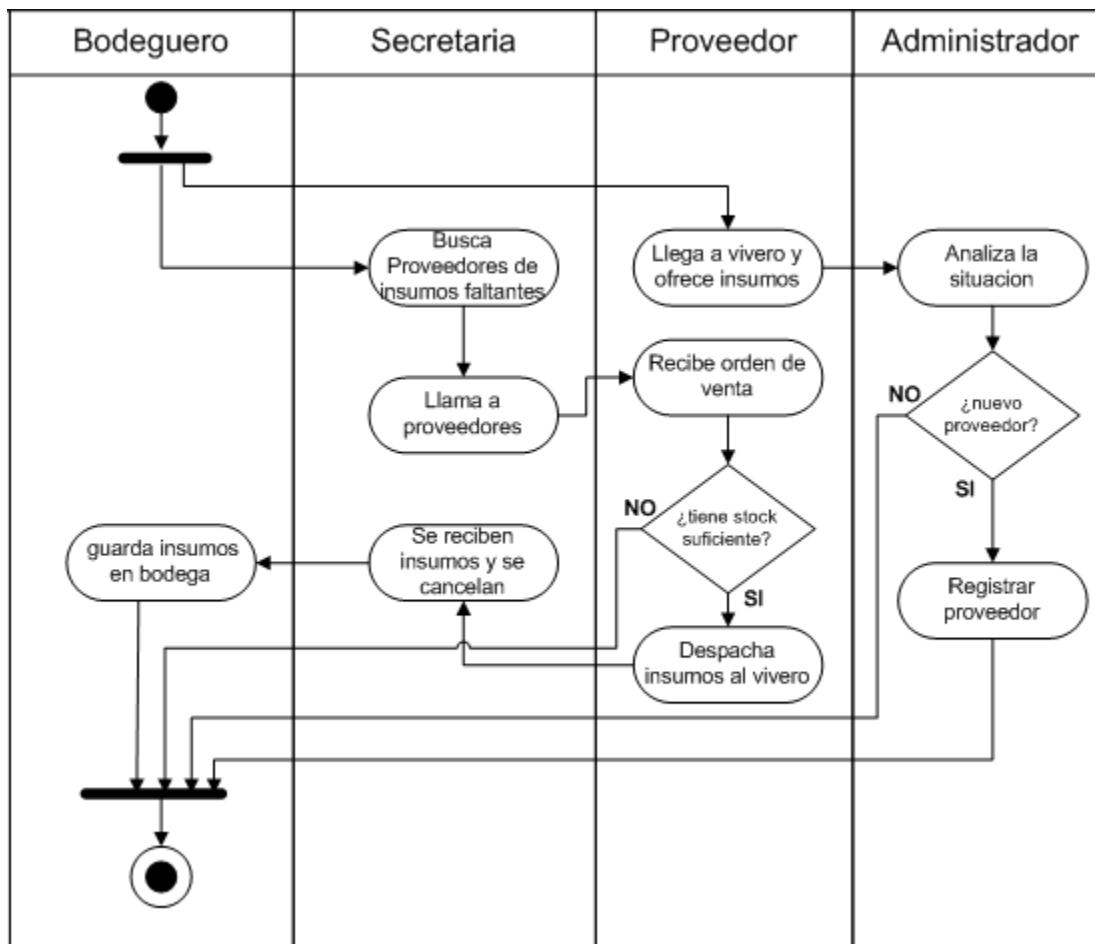


3.1.3.1.4.- Administrar Proveedor

El Proveedor se registra también en cuadernos, se anotan sus datos junto a los productos que este provee. Cuando existe algún insumo se llama a los proveedores correspondientes quienes se irán a dejar los productos al vivero.

En muchas ocasiones proveedores nuevos se acercan al vivero a ofrecer sus productos, los cuales después de ser examinados por el administrador se registran o no, en el cuaderno.

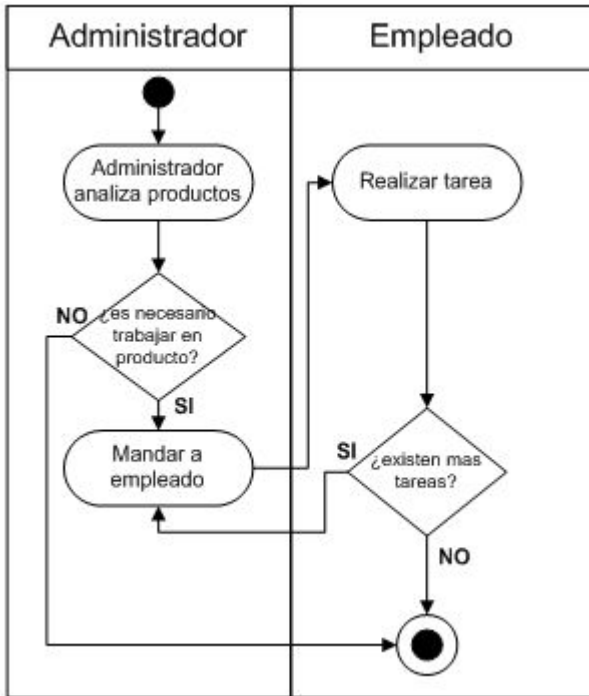
Imagen 06: Proceso actual administración proveedores



3.1.3.1.5.- Administración De La Producción

Los empleados son los que realizan los distintos procesos de producción, esto quiere decir que ejecutan los pasos que necesita la planta según su especie y el estado en que esta se encuentre, todo lo dicho lo hacen a mandato de la persona que realmente sabe y conoce los procesos productivos: el administrador.

Imagen 07: Proceso actual producción Ecogreen

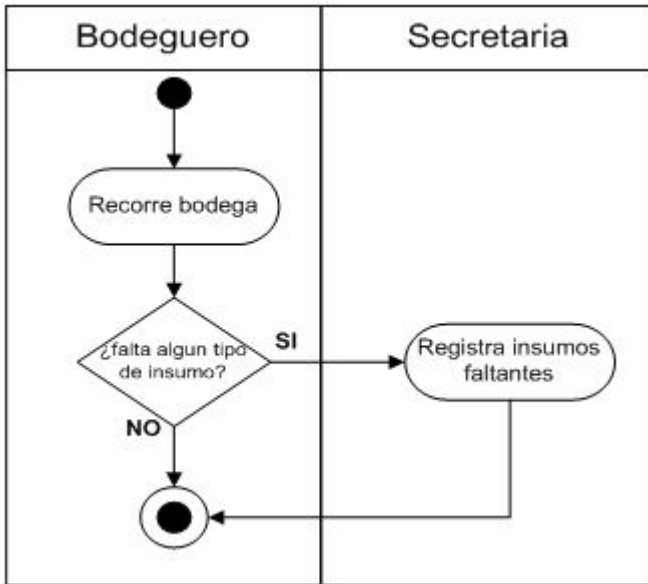


3.1.3.2.- Bodeguero

3.1.3.2.1.- Administración De Bodega

Actualmente no se registra nada relacionado con el inventario de los insumos. La bodega se abre diariamente por la persona que tiene las llaves de esta (bodeguero), este al final del día la cierra. Cuando hay poca cantidad de un insumo es trabajo de este avisar a la secretaria, quien adquirirá de los proveedores los insumos faltantes.

Imagen 08: Proceso actual dar administración bodega



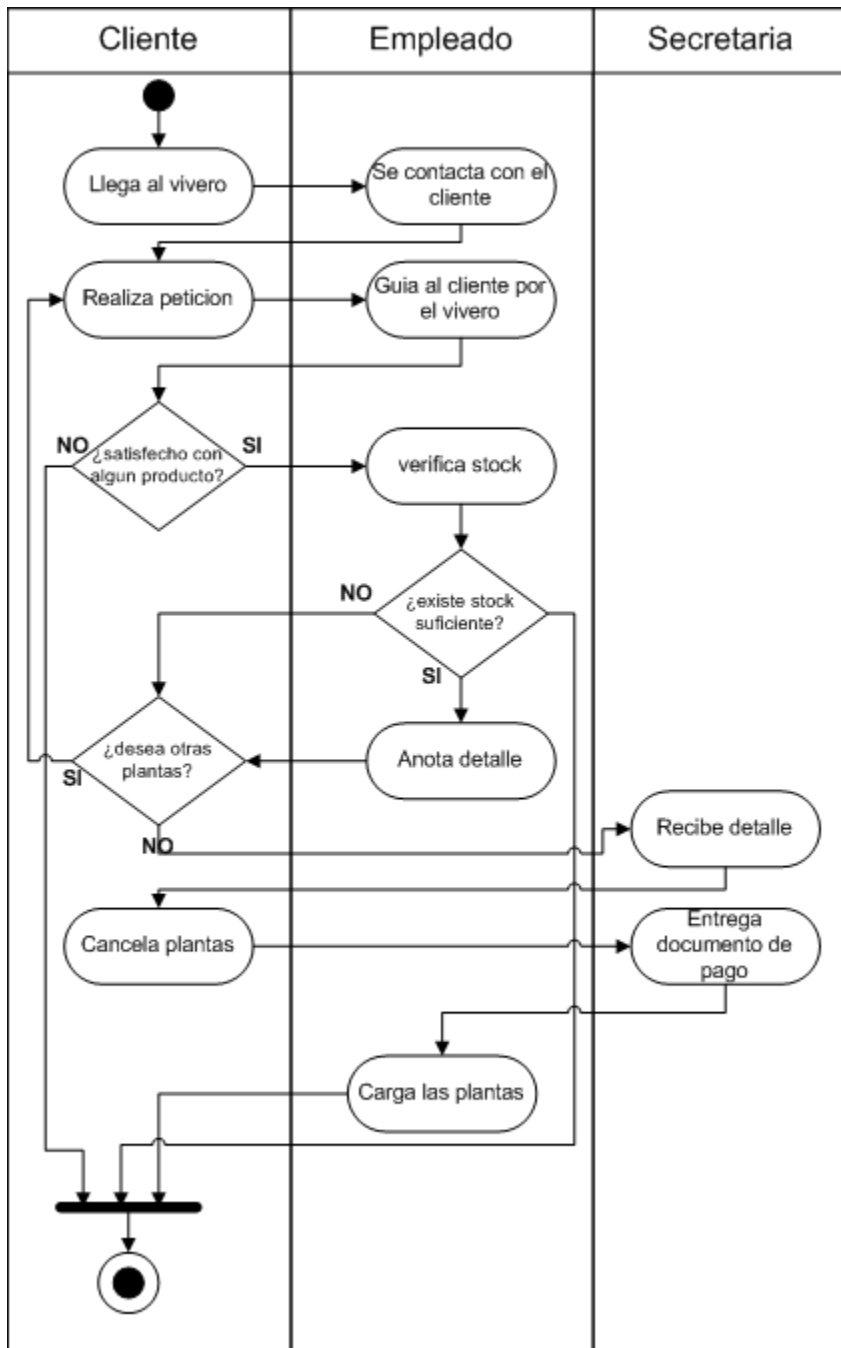
3.1.3.3.- Secretaria

3.1.3.3.1.- Venta

Las ventas las realiza el administrador o la secretaria siendo esta ultima la persona que mas cumple este rol.

Este suceso comienza cuando el cliente llega al vivero con el fin de adquirir plantas, luego se contacta con el vendedor para pedirle una especie en especial o simplemente mirar las disponibles. Después de recorrer el jardín, y comprobar que se tiene el stock necesario para satisfacer las necesidades del cliente, el vendedor entrega el detalle a la secretaria, ahora es ella quien se comunicara directamente con el cliente. El comprador cancela las plantas adquiridas, este paso puede ser de 3 formas: efectivo, cheque, cambio (hablar con el administrador) o debe (solo para clientes conocidos). Posteriormente la secretaria le entrega al cliente una factura, boleta o guía.

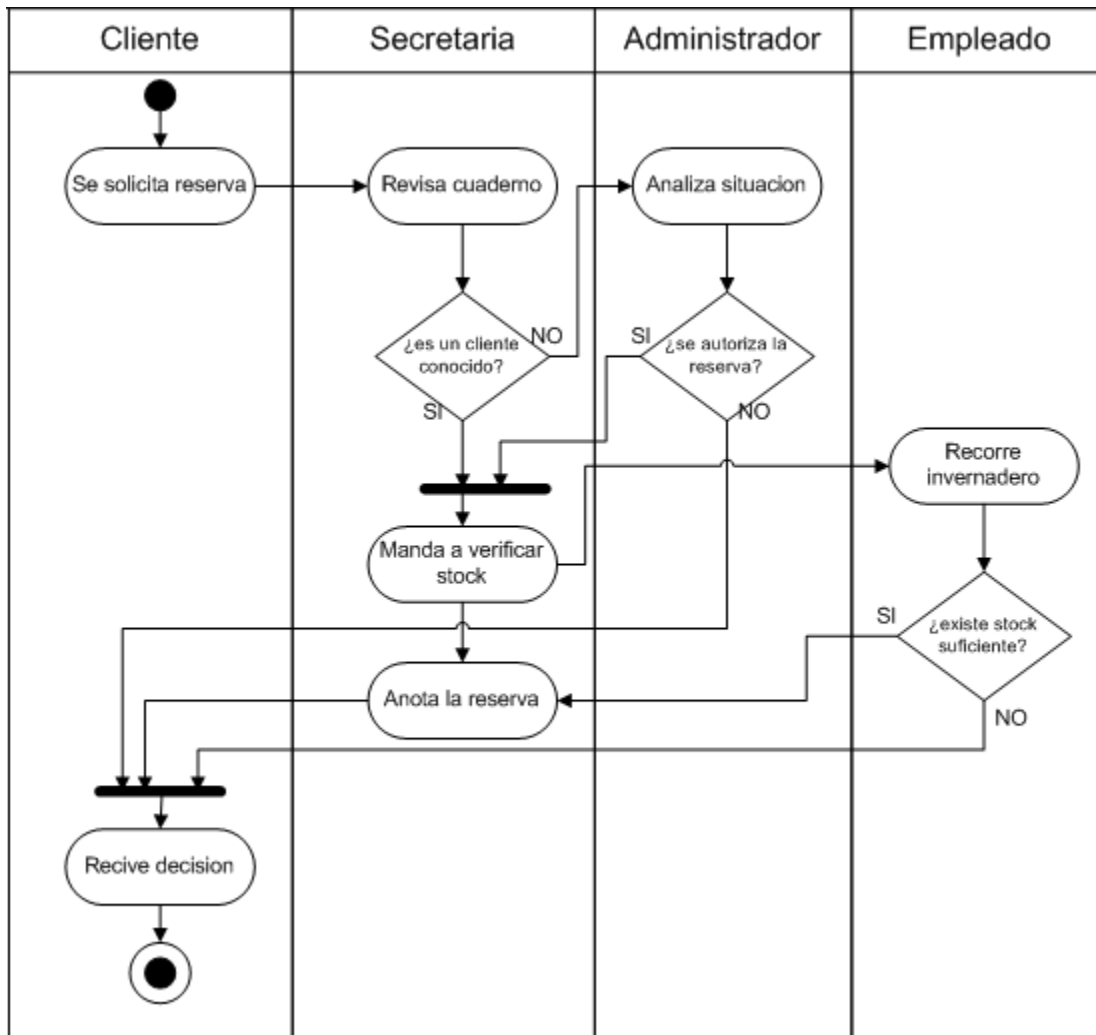
Imagen 09: Proceso actual de venta



3.1.3.3.2.- Administración De Las Reservas

Las reservas actualmente se registran en un libro, vale destacar que en el actual sistema las reservas solo se realizan a clientes que son muy conocidos por el administrador incapacitando a la secretaria el poder registrar a cualquier otro tipo de clientes.

Imagen 10: Proceso actual administración reservas Ecogreen



3.2.- Origen Del Problema

La problemática y la realización de este software surgen debido a la necesidad de establecer una inspección real al proceso productivo del Vivero Ecogreen, dándole un énfasis especial al control del vivero. Esta problemática actualmente se maneja en forma bastante deficitaria y precaria, debido que todo el trabajo se basa en conocimientos del administrador y sus trabajadores, no permitiendo tener algún conocimiento del estado actual por ejemplo el Stock de productos, tipos, personas que realizan el trabajo, etc. Es por esto que es de vital importancia retener esta gran cantidad de datos.

El sistema en general posee las siguientes desventajas:

- Falta de orden y organización dentro del vivero. Debido a que existen muy pocos registros (en cuadernos) que determinen un flujo del trabajo realizado, lo que crea confusión y descoordinación tanto de los trabajadores como del administrador de esta empresa.
- Poca eficacia y eficiencia en la atención de los clientes. Debido a que no se conocen los productos y sus cantidades de estos, lo que hace que el cliente deba visitar la tienda y realizar un paseo por las instalaciones del invernadero para así conocer si se cuenta con la demanda suficiente para satisfacer sus pedidos.
- Mal manejo de la bodega del invernadero. Debido a que no existe un inventario, no se conoce el stock de insumos, lo cual pone en riesgo en cualquier momento el proceso, ya que puede no tener de un insumo que sea vital en el crecimiento de la planta o algún tipo de insecticida que pudiese ayudar en mitigar de una enfermedad a las plantas y arbustos, también se está propenso a algún tipo de pérdida de insumos sin saber el motivo real de estas pérdidas.
- La bodega abierta todo el día la hace propensa a robos.
- No conocimiento de los clientes frecuentes, ni posibles tipos de descuentos o motivaciones a éstos para las eventuales reservas que realicen. Debido a que es fundamental dentro del proceso de reserva asignar un lote en particular de las especies requeridas al cliente frecuente, el cual es registrado en una simple hoja de cuaderno.
- No existe un historial explícito de las compras realizadas por el cliente. Solo se utilizan los “recuerdos” de las ventas realizadas en otros años. Esto trae como consecuencia el que no se puede hacer ningún tipo de estimación futura y mucho menos tratar de predecir cómo se comporta el mercado en fechas similares.
- La cantidad de plantas que se cuenta por especie no se puede saber, solamente se estima. Los inconvenientes directos que tiene este problema, es que de haber un cliente solicitando una gran cantidad de plantas no se puede saber si se lograría cumplir su solicitud.

3.3.- Flujo De Trabajo Futuro

3.3.1.- Análisis De Requerimientos.

Para el análisis y obtención de requerimientos lo que se realizó fue concertar una cantidad de citas con el cliente, en este caso es el administrador, quien nos guió en la forma de trabajar del vivero.

Es el administrador quien conoce el flujo de trabajo en detalle, específicamente la manera de desempeñarse de cada empleado dentro del vivero, conociendo ya algunos problemas que tiene el sistema actual. Otra cosa que se refirió el cliente fue como a él le gustaría que fuera el sistema a futuro y su manera de trabajar.

A grandes rasgos se ha determinado que el vivero necesita un sistema que satisfaga las siguientes necesidades:

- Registrar las ventas realizadas en el vivero, así como información importante de las ventas previamente realizadas.
- Controlar a los clientes del vivero y ver su historial de compras.
- Controlar lo que existe en bodega y los insumos faltantes.
- Registrar las fechas de los trabajos productivos así como los insumos utilizados.
- Registrar a los proveedores y los servicios que nos ofrecen estos. También consultar información sobre todos los proveedores.
- Registrar las reservas realizadas por los clientes.

A consecuencia de lo complejo del sistema, es necesario de alguna manera mantener una coordinación en el desarrollo del mismo, para resolver este problema se ha provisto la división del sistema en distintos módulos, los cuales se asocian actividades relacionadas entre sí y son de apoyo para realizar tareas más específicas.

3.3.1.1.- Clasificación De Requerimientos

Requerimientos Funcionales:

Módulo Gestión de Procesos

Ref. #1	Función	Categoría
R1.1	Registrar sembrado en el sistema.	Evidente
R1.2	Registrar abonos realizados en un lote de plantas.	Evidente
R1.4	Llevar un control sobre los eventos incurridas en la actividad.	Evidente
R1.5	Almacenar permanentemente las tareas realizadas en el sistema.	Ocultas

Módulo Venta de plantas

Ref. #2	Función	Categoría
R2.1	Registrar una nueva venta realizada.	Evidente
R2.3	Eliminar permanentemente del sistema una venta previa.	Evidente
R2.4	Verificar Stock antes de realizar una venta	Evidente
R2.5	Actualizar Stock después de una venta	Ocultas
R2.6	Consultar información relevante respecto a las ventas	Evidente
R2.7	Almacenar permanentemente en el sistema las ventas realizadas	Ocultas
R2.8	Asignar bono de venta a cada empleado que esté involucrado en	Ocultas

Módulo Proveedores de Insumo

Ref. #3	Función	Categoría
R3.1	Registrar un Proveedor junto con los insumos que provee.	Evidente
R3.2	Modificar proveedor ingresado previamente.	Evidente
R3.3	Eliminar permanentemente del sistema a un proveedor.	Evidente
R3.4	Verificar Stock antes de ver que Proveedor satisface las	Evidente
R3.5	Almacenar permanentemente en el sistema los proveedores	Ocultas
R3.6	Consultar información relevante respecto a los Proveedores	Evidente

Módulo Informes

Ref. #4	Función	Categoría
R4.1	Generar informes relacionados con la información del sistema	Evidente

Módulo Mantenición de Empleado

Ref. #5	Función	Categoría
R5.1	Registrar un nuevo empleado.	Evidente
R.5.2	Almacenar permanentemente los empleados del sistema	Oculto
R5.3	Registrar modificaciones sobre información de empleados en el	Evidente
R5.4	Eliminar permanentemente a los empleados.	Evidente
R5.5	Consultar información relevante respecto a los empleados	Evidente

Módulo Reserva de Plantas

Ref. #6	Función	Categoría
R6.1	Registrar las reservas de plantas por petición de los clientes.	Evidente
R.6.2	Actualizar el stock de plantas disponibles.	Oculto
R6.3	Anular las reservas previamente realizadas.	Evidente
R6.4	Consultar información relevante de reservas de clientes.	Evidente
R6.5	Consultar información relevante de reservas de productos.	Oculto

Módulo Mantenición de Usuarios

Ref. #7	Función	Categoría
R7.1	Registrar un nuevo Usuario.	Evidente
R.7.2	Almacenar permanentemente los Usuarios del sistema	Oculto
R7.3	Registrar modificaciones sobre información de Usuarios	Evidente
R7.4	Eliminar permanentemente a los Usuarios del sistema.	Evidente

R7.5	Consultar información relevante respecto a los Usuarios	Evidente
------	---	----------

Módulo Gestión de Bodega

Ref. #8	Función	Categoría
R8.1	Registrar un Insumos y Proveedor por el cual se adquirió.	Evidente
R8.2	Modificar Insumos ingresados previamente.	Evidente
R8.3	Eliminar permanentemente del sistema a un Insumo.	Evidente
R8.4	Verificar Stock y Eliminar insumo del sistema de ser el caso	Oculto
R8.5	Almacenar permanentemente en el sistema los Insumos	Oculto
R8.6	Consultar información relevante respecto a los Insumos	Evidente

Módulo Mantenición de Cliente

Ref. #9	Función	Categoría
R9.1	Registrar un nuevo cliente en el sistema.	Evidente
R.9.2	Almacenar permanentemente los clientes	Oculto
R9.3	Registrar modificaciones sobre información de los clientes en el	Evidente
R9.4	Eliminar permanentemente a los Clientes.	Evidente
R9.5	Consultar información relevante respecto a los clientes	Evidente

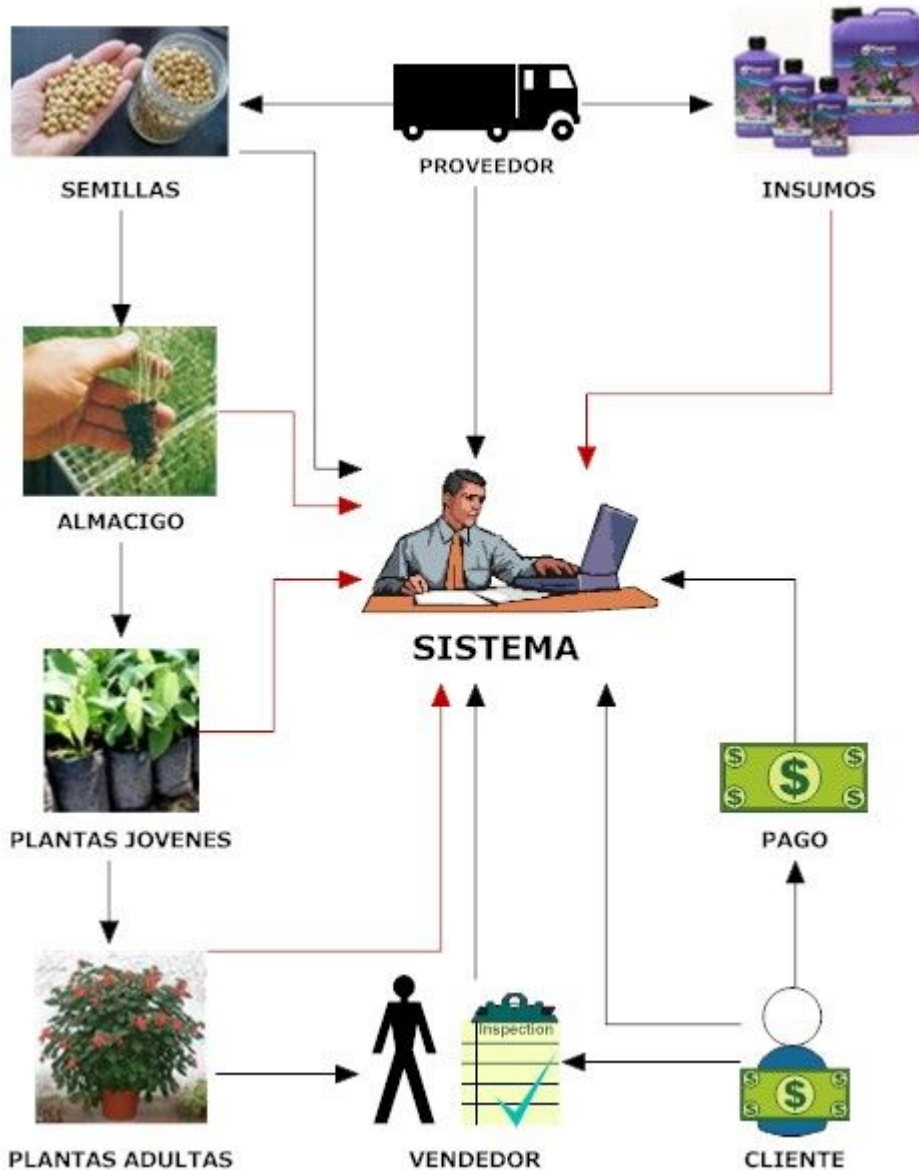
3.3.1.2.- Requerimientos No Funcionales:

- Debe ser fácil de usar y ser amigable para el manejo de sus usuarios.
- Debe tener colores corporativos y que no molesten la vista de estos.
- Compatible con el Sistema Operativo actual.
- Adaptable para posibles inclusiones de nuevos desarrollos y complementos, en otras palabras escalable.
- Informar con mensajes cualquier eventualidad.

- Todo uso del sistema requiere la autenticación de los usuarios.
- Las interfaces son legible, simples de usar, con colores adecuados y ofrece la interactividad con el usuario.
- Tiempos de respuesta rápida para consultas y obtención de información relacionada al negocio y para transacciones.
- Información confiable y exacta respecto a los productos disponibles para el cliente.
- Los datos a ingresar deben ser consecuentes con cada uno de los datos que se procesen en el sistema. Lo datos deben poder manejarse en forma rápida y poder establecer estadísticas con ellos.

3.3.2.- Propuesta De Solución Del Flujo Futuro Para El Proyecto.

Imagen 11: Flujo Futuro Empresa Ecogreen



El sistema funcionara de manera controlada, es decir la mayoría de las funciones del sistema serán efectuadas automáticamente y registradas. Por ejemplo el sistema será capaz de agilizar la interacción de ventas entre cliente y vendedor, dándole seguridad a las diferentes transacciones, incorporando organización de los productos dentro del vivero.

A continuación se describe las diferentes operaciones que realizaran los actores del sistema:

3.3.2.1.- Secretaria

3.3.2.1.1.- Realizar Venta

La venta comienza cuando el cliente llega al vivero con el deseo de adquirir productos. A su recepción se encuentra con el vendedor, si el cliente tiene claro que especie desea se le enviara con la secretaria, de no ser el caso (que es lo normal) se recorrerá el jardín con la intención de entusiasmar al cliente con algún producto, para posteriormente entregarle a la secretaria el detalle de lo que se comprara.

La secretaria deberá loguearse en el sistema e ingresar a la opción de “realizar venta” donde el sistema le entregara la visualización de los campos a llenar. Deberá ingresar los datos correspondientes al detalle y el sistema retornara el total a pagar. Luego se deberá seleccionar el tipo de pago que se llevara a cabo. Se confirma la transacción y la venta estará realizada. En cada venta se actualizara el inventario y el informe de las ventas realizadas.

3.3.2.1.2.- Realizar Reserva

Ya que en el vivero existen plantas que son de temporada, las reservas son un tema muy serio, si existe una reserva para plantas y nunca se lleva a cabo se producen mermas y todo el costo tiene que ser asumido por la empresa. Es por esto, que las reservas se realizaran solo a clientes exclusivos, que se tiene un buen historial de ventas, y que tiene la aprobación del administrador del jardín.

Este comienza cuando el cliente llama o se acerca al vivero con la intención de realizar una reserva, la secretaria se loguea en el sistema y busca al cliente. De encontrarlo le comunica el stock de la especie que requiere, el cliente comunica a la secretaria las fechas que necesita dicho producto. Con todos estos datos la secretaria llena los campos necesarios en la opción “realizar reserva” y confirma.

3.3.2.2.- Encargado

3.3.2.2.1.- Gestión De Bodega

A diferencia de antes, ahora el encargado de bodega será el responsable de registrar en el sistema todo lo que sucede con los insumos.

Las herramientas tendrán un serial único para el vivero, el encargado tendrá que loguearse en el sistema y registrar los empleados que estén utilizando las herramientas, para que en un posible futuro que estas desaparezcan o se estropeen muy luego, se tenga a los responsables de estos hechos. Si se producen perdidas en la bodega y no existen culpables será el encargado quien tendrá que correr con las responsabilidades, por esto ahora la bodega estara cerrada todo el día y el encargado la abrirá cuando le soliciten algún implemento.

3.3.2.2.2.- Gestión De Procesos

El encargado será responsable también de ingresar en el sistema otras tareas:

- Ingresos de plantas sembradas.
- Fechas de cuando los plantas subieron de nivel
- Insumos utilizados en las plantas

Todo esto lo hará a indicación de los empleados que están trabajando en dichas plantas, el caso de los insumos, el encargado es el mismo quien maneja la bodega, aquí se tiene un control directo.

3.3.2.3.- Administrador

3.3.2.3.1.- Gestionar Empleados

Los empleados de la tienda serán manejados por el sistema, el cual tendrá todos los datos de cada uno, serán ingresados, eliminados o actualizados por el administrador.

3.3.2.3.2.- Gestionar Cuentas

Cada usuario que ingrese al sistema deberá poseer una clave de acceso para darle seguridad al sistema. La cual deberá ingresar al momento de hacer alguna operación. Los empleados comunes y vendedores no podrán y no deberán acceder al sistema, la secretaria y el encargado deberán hacerlo para realizar sus tareas diarias.

3.3.2.3.3.- Revisión De Informes

El sistema entregara informes sobre los datos que se hayan ingresados anteriormente en los distintos procesos que este tiene.

3.3.2.3.4.- Gestión Proveedores

Los datos de los proveedores se registraran en el sistema junto a los insumos que estos proveen. El administrador dispondrá de la información necesaria para cuando los insumos estén escasos.

Módulos del Sistema

En este punto presentamos los módulos del sistema junto a la descripción de los mismos.

Modulo	Descripción
Gestión de Trabajos	Corresponde al historial que se refiere a los procesos productivos aplicados en las plantas desde sus primeros días de vida hasta que se realiza la venta.
Venta de Plantas	Trata de la gestión de las ventas llevadas a cabo, junto con todas las elecciones que nos permite esta opción.
Proveedores de Insumos	Corresponde a la gestión de la información relacionada a los Proveedores de insumos para la posterior producción de las distintas especies de plantas.
Mantención de Empleado	Nos permite trabajar con la funcionalidad de tener la información relacionada a los empleados que trabajan en el vivero (Ingreso, Modificación, Eliminación y Consulta).
Informes	Corresponde a la gestión de la información relacionada a datos relevantes generados en el negocio orientados para el uso administrativo.
Reservas de Plantas	Es todo lo relacionado con las reservas que pueden realizar los clientes al vivero
Mantención	Corresponde a la gestión de la información relacionada a los Usuarios que utilizan el Sistema (Ingreso, Modificación, Eliminación y Consulta), para

de Usuarios	controlar su acceso.
Gestión de Bodega	Trata de la gestión de los insumos en la bodega ya seas los pertinentes a la producción de las plantas(abonos, fertilizantes etc.) o insumos tales como palas, martillos, carretillas etc.
Mantención de Clientes	Corresponde a la gestión de la información relacionada a los Clientes del vivero (Ingreso, Modificación, Eliminación y Consulta).
Modulo Productos	Corresponde a la gestión de la información relacionada con los productos y los lotes (Ingreso, Modificación, Eliminación).

4.- Metodología

En este punto se describen las actividades desarrolladas basadas en RUP para este proyecto. Por las características del proyecto, se han incluido los siguientes artefactos, roles y actividades de la metodología, manteniendo los más esenciales. A continuación se describen brevemente cada uno de los artefactos que se generarán y usarán durante el proyecto.

1. Flujos de Trabajo

Se utilizarán Diagramas de Actividad para modelar los Flujos de Trabajo (workflows) del área problema, tanto los actuales (previos a la implantación de nuevo sistema) como los propuestos, que serán soportados por el sistema desarrollado.

2. Características del Producto Software

Es una lista de las características principales del producto, deseables desde una perspectiva de las necesidades del cliente (requerimientos obtenidos).

3. Modelo de Casos de Uso

El modelo de Casos de Uso presenta la funcionalidad del sistema y los actores que hacen uso de ella. Se representa mediante Diagramas de Casos de Uso.

4. Especificaciones de Casos de Uso

Para los casos de uso que lo requieran (cuya funcionalidad no sea evidente o que no baste con una simple descripción narrativa) se realiza una descripción detallada utilizando una plantilla de documento, donde se incluyen: precondiciones, pos condiciones, flujo de eventos, requisitos no-funcionales asociados.

5. Modelo de Diseño

Este modelo establece la realización de los casos de uso en clases y pasando desde una representación en términos de análisis (sin incluir aspectos de implementación) hacia una de diseño (incluyendo una orientación hacia el entorno de implementación). Está constituido esencialmente por un Diagrama de Clases y algunos Diagramas de Estados para las clases que lo requieran.

6. Modelo de Pruebas

Para cada Caso de Uso se establecen pruebas de Aceptación que validarán la correcta implementación del Caso de Uso. Cada prueba es especificada mediante un documento que establece las condiciones de ejecución, las entradas de la prueba, y los resultados esperados.

7. Manual de Instalación

Este documento incluye las instrucciones para realizar la instalación del producto.

8. Material de Usuario

Corresponde a un conjunto de documentos y facilidades de uso del sistema.

9. Producto

Todos los ficheros fuente y ejecutable del producto.

5.- Plan De Trabajo

5.1.- Descripción Trabajo Realizado (Carta Gantt)

5.1.1.- Inicio

- **Definir Objetivos:** Definición de objetivos para el inicio del proyecto
- **Entrevista con el cliente (Levantamiento de requerimientos):** Cita con el administrador del vivero para conocer su forma de trabajo, ya sea documentos que ocupa para trabajar y actividades que realiza. A su vez nos explica el funcionamiento general de la empresa.
- **Análisis de requerimientos:** Análisis de los primeros datos entregados por el administrador del vivero.
- **Elección del Paradigma y Metodología:** Estudio y elección de paradigma y metodología de desarrollo más apta para el proyecto, tomando en cuenta la solución que se le quiere dar a este.
- **Entrevista con el cliente:** Levantamiento de requerimientos: Reunión con el administrador para explicar más en detalle el funcionamiento interno del vivero (proceso de producción).
- **Análisis de requerimientos:** Análisis y refinamiento de los requerimientos obtenidos en la reunión.

- **Definir Características del Sistema:** definición de las características del sistema basados en los requerimientos y solicitudes del cliente.
- **Estudio de Factibilidad:** Estudio de viabilidad del proyecto.
- **Entrevista con el cliente (Levantamiento de requerimientos):** Reunión con el cliente para ver más detalladamente procesos de la empresa (venta y bodega).
- **Análisis de requerimientos:** Análisis de los datos obtenidos en la última reunión.
- **Análisis de riesgos:** Estudio de los riesgos y creación de planes de mitigación y contingencia.
- **Elección de Herramientas:** Selección de las herramienta a utilizar en el desarrollo del proyecto.
- **Entrevista con el cliente (Levantamiento de requerimientos):** Entrevista con el cliente para refinar detalles pendientes en el proceso productivo e informar posibles cambios en este con el nuevo sistema.
- **Análisis de requerimientos:** Análisis de los datos obtenidos en la última reunión con el cliente.

- **WorkFlows:** Diseño de los workFlows actuales y futuros de los procesos.
- **Modelos Casos de Uso:** Diseño del modelo general de casos de usos.

5.1.2.- Elaboración

- **Refinamiento Casos de Uso:** Revisión y refinamiento detallado todos los casos de uso.
- **Especificaciones Casos de uso:** Especificación de los casos de uso más importante para los procesos relevantes del vivero.
- **Refinamiento Casos de Uso:** Revisión de los casos de uso más relevantes.
- **Diagrama de secuencia:** Creación de los diagramas de secuencia de importancia para un mejor entendimiento en caso de posibles cambios.
- **Diagrama entidad-relación:** Diseño del diagrama entidad relación.
- **Diagrama de clases:** Creación del modelo de clases.
- **Modelo Análisis y Diseño:** Creación del modelo de diseño.
- **Modelo Implementación:** Construcción del modelo de implementación.

- **Modelo de Pruebas:** Creación del modelo de pruebas basado en el modelo de pruebas caja negra.
- **Prototipo Modulo:** Codificación de un pequeño modulo del sistema.

5.1.3.- Construcción

- **Entrevista con el Cliente:** Cita con el administrador del vivero para ratificar su forma de trabajo y con esto poder aclarar algunas dudas en la generación de documentos que necesita que el sistema sea capaz de generar para mejorar la toma de decisiones
- **Análisis:** Análisis de los datos entregados en la entrevista previa.
- **Refinamiento Modelo Entidad Relación:** Mejoras que se le deben realizar al modelo relacional para poder satisfacer las necesidades del cliente, basados en la entrega de datos facilitados por el administrador del vivero
- **Entrevista con el Cliente:** Cita con el cliente para informar el avance el proyecto y refinar detalles de este
- **Refinamiento Modelo Entidad Relación:** Mejoras que se le deben realizar al modelo relacional para poder satisfacer las necesidades del cliente, basados en la entrega de datos facilitados por el administrador del vivero

- **Refinamiento Modelo de Clases:** Mejoras realizadas al modelo de clases basadas en el refinamiento del modelo entidad relación
- **Interfaz Principal:** diseño y construcción de la interfaz principal del sistema.
- **Modulo Venta de Plantas:** Construcción del modulo encargado de las gestión de las ventas de los productos
- **Modulo Mantenión Empleados:** Construcción del modulo encargado de las gestión de las mantención y manipulación de los datos de los empleados
- **Modulo Reserva Plantas:** Construcción del modulo encargado de manejar y controlar los datos referentes a las reservas de los productos por parte de los clientes VIP
- **Modulo Gestión Bodega:** Construcción del modulo encargado de la gestión de la entrega de utensilios a los empleados de la empresa para desarrollar las labores asignadas.
- **Modulo Mantenión Clientes:** Construcción del modulo encargado de la gestión de la mantención y manipulación de los datos de los clientes
- **Informe Avance:** Entrega del informe de avance
- **Presentación de Avance:** Presentación de los avances del proyecto hasta la fecha

- **Modulo Mantención Productos:** Construcción del modulo encargado de las gestión de las mantención y manipulación de los datos de los productos.
- **Modulo Proveedores Insumo:** Construcción del modulo encargado de las gestión de las mantención y manipulación de los datos de los proveedores de insumos del vivero
- **Modulo Procesos:** Construcción del modulo encargado de las gestión los procesos y trabajos realizados por lo empleados sobres los distintos lotes con los productos distribuidos en el vivero.
- **Modulo Mantencin Usuarios:** Construcción del modulo encargado de la gestión de la mantención y manipulación de los datos de los usuario del sistema.

5.1.4.- Transición

- **Integración:** Proceso Integración de todos los módulos del sistema.
- **Pruebas:** Periodo de testeo del sistema integrado y con todos las funcionalidades operativas.
- **Mejoras:** Refinamiento del sistema dependiendo de los resultados de las pruebas previamente realizadas.
- **Entrega Software Terminado:** Entrega del sistema terminado.

- **Revisión Software:** revisión final del sistema.
- **Mejoras:** Refinamiento del sistema dependiendo de la revisión previamente realizada.
- **Informe Final:** Entrega del informe final.

6.- Desarrollo

6.1.- Estudio De Factibilidad

El estudio de factibilidad es fundamental a la hora de enfrentarse a un proyecto, ya que el estudio de factibilidad nos permite determinar si el proyecto al cual nos enfrentamos es realmente viable. Es claro que para sacar adelante un proyecto de cualquier índole se requiere cumplir con una planificación y además adecuarse a los recursos disponibles para el desarrollo, debemos tener presente que todo proyecto es posible sacar adelante si se cuenta con tiempo y recursos ilimitados, esto hace necesario para nosotros realizar un estudio de factibilidad, ya que no contamos con recursos ilimitados y además estamos sujetos a plazos con los cuales nos hemos comprometido cumplir. El estudio de factibilidad nos permitirá evitar problemas futuros, ya que estimaremos en cierta medida el esfuerzo necesario para lograr tener éxito en el desafío al cual nos enfrentamos, la información obtenida de este estudio nos permitirá determinar si las condiciones bajo las cuales enfrentamos la problemática anteriormente señalada son las apropiadas para realizar el proyecto en su totalidad, y estar atentos a posibles situaciones adversas que nos obliguen a tomar cursos alternativos para apegarnos a los recursos e hitos propuestos.

- **Factibilidad Técnica:** Consiste en determinar si es posible dar solución a la problemática presentada tomando en cuenta los recursos computacionales disponibles en la organización.
- **Factibilidad Económica:** Consiste en ver si el proyecto es viable económicamente.
- **Factibilidad Legal:** El estudio de la factibilidad legal, tiene como objetivo verificar para cada escenario, si el sistema a desarrollar no vulnera las leyes vigentes o la reglamentación propia de la organización. Es decir, pretende observar si no se incurre en infracciones, violaciones u otros que podrían provocar la imposibilidad de poner en ejecución el sistema, o su interrupción en algún momento de su operación rutinaria.

- **Factibilidad Operacional:** Este estudio pretende determinar si el sistema será usado y aceptado por los usuarios finales, cumpliendo realmente los objetivos propuestos y solucionando los problemas para los que será diseñado.

A continuación llevamos a cabo el estudio de factibilidad dividido en los cuatro aspectos antes mencionados:

6.1.1.- Factibilidad Técnica

El estudio de factibilidad técnica se utiliza para determinar si el proyecto es viable con las tecnologías y recursos existentes además de las restricciones impuestas.

Las limitaciones técnicas están dadas porque el cliente que ya cuenta con el siguiente hardware en sus instalaciones:

- Athlon xp 2.6 GHz, 256 Mb de RAM, disco duro de 80 Gigas.
- Athlon xp 2.6 GHz, 512 Mb de RAM, disco duro de 60 Gigas.
- Athlon xp 2.6 GHz, 512 Mb de RAM, disco duro de 60 Gigas.

Se recomienda la utilización de un switch para poder realizar la conexión entre el hardware antes mencionado.

El proyecto es completamente viable técnicamente ya que con las limitaciones técnicas actuales no son restrictivas y el sistema funcionará perfectamente en el hardware mencionado (Athlon xp 2.6 GHz) el cual actualmente funciona con Windows XP con SP2, el hardware de mayor capacidad será utilizado como servidor del sistema.

6.1.2.- Factibilidad Económica

El estudio de factibilidad económica se basara en análisis de costos y beneficios asociados directamente a cada alternativa del presente proyecto. La idea del análisis costo/beneficio es identificar todos los costos y beneficios de adquirir el nuevo sistema y hacer una comparación de ellos. Se comparan los costos de cada alternativa con los beneficios esperados para asegurar que los beneficios excedan a los costos, y estos resultados son

comparados entre las diversas alternativas probables, por ejemplo comparar los costos/beneficios del sistema a desarrollar por nosotros con los costos/beneficios de continuar operando con los sistemas antiguos.

En el estudio de factibilidad económica se deben tener en cuenta los costos de implementación en este tipo de costos podemos mencionar: el costo de llevar a cabo la investigación de sistemas, el costo de hardware y software necesarios para desarrollar la aplicación, costos de operación y de capacitación para operar el nuevo sistema versus los beneficios que se pueden cuantificar más fácilmente y se encasillan en dos tipos principalmente: Primero podemos hablar de ahorros en costos, tales como disminución de costos de operación y aumentos de utilidades directas.

Con este proyecto nuestro cliente se verá beneficiado ya que podrá tener el control de las existencias de sus productos, entregar una mejor atención y beneficios a sus clientes frecuentes lo que aumentará sus ingresos, responsabilizar a los empleados en caso de mermas o extravíos de insumos, por lo anteriormente expuesto el costo de llevar a cabo el proyecto es mucho menor que los beneficios, para poder demostrar esto existe una serie de herramientas, equivalentes a diferentes métodos de evaluación de proyectos de inversión, los cuales consisten en comparar de alguna forma el flujo de ingresos con el flujo de desembolsos del proyecto. De estos métodos los más utilizados son: VAN y TIR.

6.1.2.1.- Inversión Inicial

Costo de dos desarrolladores para implementar el sistema y realizar el análisis del sistema por cada uno serían \$450.000 en un periodo de 10 meses que en total serían \$4.500.000, Además de un switch necesario para conectar el hardware en la empresa (DLink Switch 5 bocas DES-1005D 10/100) que tiene un precio aproximado a la fecha de \$11000 y que cumple satisfactoriamente todos los requisitos.

6.1.2.2.- Costos Operacionales

Costos operacionales estimados en \$20.000 por conceptos de insumos por 12 meses, agregando a la inversión inicial es de \$240.000 anual por el tiempo que se utilice el sistema.

6.1.2.3.- Costos De Capacitación

Costo de capacitación del nuevo sistema y de las funcionalidades más importantes es de \$120.000, las que se desglosan en 4 charlas explicativas con un costo de \$30.000 c/u y el manual de usuario incluido en el costo asociado al desarrollo.

6.1.2.4.- Costos De Mantención

Costos de mantención al equipo que funcionará en el vivero y al servidor que tiene el sistema, será aproximadamente de \$800.000.

6.1.2.5.- Beneficios Anuales

Beneficio del uso del sistema, tomando en cuenta el ahorro de tiempo en los procesos tales como ventas o reservas, mejoras considerables en el manejo de insumos en las bodegas y las sanciones correspondiente en caso de extravíos de estos (palas, martillos, carretillas etc.), mejoras en ofertas y oportunidades de pagos a los clientes habituales que llevarían a un aumento notable de las ventas de especies específicas en determinadas fechas, además de optimizar considerablemente el proceso de producción de las especies en donde el sistema además de llevar el control de los procesos servirá para comparar y verificar si los procesos aplicados son los correctos o si están siendo mal y bien aplicados mediante la entrega de informe detallados al administrador para una mejora en la toma de decisiones todos estos beneficios tienen un valor mensual de \$450.000 aproximadamente (estimación realizada en una reunión en conjunto con el cliente).

6.1.2.6.- Total Beneficios De Operación

Corresponde al ahorro menos los gastos operacionales, y tendríamos nuestro flujo de caja, $\$4.511.000 - \$1.160.000 = \$3.351.000$, la tasa de interés aplicada por el banco Central es de 5.20% a mayo del 2008, y el proyecto se evalúa a 4 años.

$$VAN = \sum_{n=1}^n F_n / (1+r)^n$$

$$VAN = -4.511.000 + \frac{3.351.000}{(1+0.052)} + \frac{3.351.000}{(1+0.052)^2} + \frac{3.351.000}{(1+0.052)^3} + \frac{3.351.000}{(1+0.052)^4}$$

$$VAN = -4.511.000 + 3.185361,2 + 3.027.909,9 + 2.878.241,3 + 2.735.970,8$$

$$VAN = 7316483,2$$

$$\text{Período de Recuperación PR} = \frac{I(\text{Inversión inicial})}{\text{Beneficio Neto}}$$

$$PR = \frac{4.511.000}{3.351.000} = 1,35 \approx 2 \text{ Se recupera la inversión aproximada en 2 años.}$$

6.1.3.- Factibilidad Legal

Es fundamental para el éxito del proyecto no infringir ninguna ley[1]., para esto nos debemos preocupar de contar con todas las licencias correspondientes a las tecnologías y herramientas a utilizar en el desarrollo, Con respecto al software, no se tendrán que adquirir licencias, ya que la Escuela de Ingeniería Informática y el Desarrollador ya las poseen, además, la versión de la herramienta MS Visual Studio Express 2005, luego de registrarla gratuitamente es totalmente funcional, por lo tanto su uso es legal para fines de uso Académico.

Otro punto importante a tener en cuenta las normas, leyes y sanciones vigentes en el país. Para lo anterior se consideró la legislación actual respecto a delitos informáticos en el país, Ley N° 19223 [1]. Los datos facilitados por al empresa Ecogreen han sido utilizados solo para el buen desarrollo del sistema y respetando todos los puntos importantes en la ley antes mencionada.

Por lo tanto cumpliendo con las normas y leyes antes mencionadas nuestro proyecto no debiera tener ningún tipo de problema legal.

6.1.4.- Factibilidad Operacional

Para verificar la factibilidad operacional la participación de trabajadores de la empresa es esencial para así disminuir la resistencia al cambio por parte de los futuros usuarios del sistema, los usuarios finales que se relacionaran directamente con el sistema forman parte importante ya que son ellos quienes fueron tomados en cuenta para la toma de requerimiento y las respectivas funciones a realizar por el sistema.

Las siguientes preguntas fueron resueltas en conjunto con los entrevistados (administrador, secretaria y encargado).

¿El sistema producirá inconvenientes en el flujo de trabajo actual de la empresa?

El sistema en si no debiese causar algún impacto en forma negativa, ya que la forma de trabajo actual se estará mejorando e incorporando una serie de elementos que antes simplemente no se tomaban en cuenta como lo son el uso de la información y el detalle de las cosas realizadas en los productos, además de tener el conocimiento de quien y con que lo realizo; Todo esto será registrado para así tener una base de referencia en cada labor a realizar.

¿El sistema dificultará el actuar de los distintos empleados de la empresa?

El sistema en si, puede alterar un poco las actividades de los empleados ya que establecerá responsabilidades para cada uno de los trabajos a realizarse y quizás forme un poco de confusión en un principio, pero una vez aprendido la forma de ir registrando y basándose en el sistema será de gran ayuda a nuestra labor, además se podrá mejorar el sistema de bonos de producción que es muy relevante para los empleados.

¿El sistema aumentara la productividad de los empleados de la empresa?

Sin duda la razón de este sistema es establecer control y patrones para nuestro futuro trabajo y dejar de depender de la experiencia de los empleados para cumplir nuestras metas, más aun, como este nuevo sistema ayudara a establecer con exactitud los bonos de producción obviamente mejorara la productividad, Así cada trabajador será más eficaz en lo que realiza para mejorar su beneficio propio y así también el de la empresa.

De acuerdo a lo anterior, se concluye que el sistema cuenta con un amplio apoyo por parte de la empresa Ecogreen y que el sistema se usará como se prevé.

Como conclusión y viendo la positiva aceptación en todos los estudios de factibilidad antes mencionados se puede deducir que el proyecto es absolutamente viable en todas las áreas.

6.2.- Análisis De Riesgo

Una tarea importante, es anticipar los riesgos que podrían afectar el desarrollo y buen diseño del proyecto, además de la calidad del software a implementar, es sumamente necesario emprender acciones para evitar esos riesgos. Por lo tanto se estableció estrategias de riesgo proactivas. Para esto se realizó estudios a los riesgos potenciales, valorando su probabilidad y su impacto para establecer una prioridad según su importancia.

El primer objetivo del equipo de desarrollo es evitar el riesgo, pero como no se pueden evitar todos los riesgos[2], el equipo trabajó en el desarrollo de planes de contingencia que permitirán responder de una manera eficaz y así poder controlar y minimizar la aparición de un riesgo el cual podría en un caso extremo hacer del sistema un fracaso.

Para esto se identificaron en un listado los posibles riesgos, luego de un análisis previo se han priorizados por impacto y probabilidad de ocurrencia, con esto se pudo crear un plan de riesgo para mitigar o eliminar los riesgos más relevantes encontrados en el estudio.

A continuación se presenta una lista con los riesgos identificados, incluyendo un análisis de su impacto considerando 4 niveles: catastrófico, crítico, marginal, despreciable, y considerando la probabilidad de que ocurra.

Riesgo	Impacto	Probabilidad
El sistema resulta ser más grande de lo que se tenía previsto	Crítico	50%
Grado de estimación de la seguridad muy bajo debido a la	Crítico	30%

inexperiencia del grupo desarrollador.		
Base de datos muy grande, con muchas tablas.	Marginal	80%
Cambios Drásticos en el proceso productivo del vivero	Crítico	15%
Conflictos en elección de herramientas.	Marginal	90%
Falta de entusiasmo por parte de los usuarios finales con el proyecto.	Crítico	30 %
Los clientes no comprenden el impacto de los cambios en los requerimientos	Crítico	25 %
Inasistencia a reuniones por parte de los miembros del equipo debido a trabajo.	Marginal	15 %
Se proponen cambios en los requerimientos que requieren rehacer el diseño.	Despreciable	10 %
Falta de comprensión de los requerimientos por parte de los desarrolladores.	Despreciable	8 %

Priorización inadecuada de requerimientos de poco interés para el cliente	Marginal	20%
---	----------	-----

Dado que los riesgos están identificados y priorizados se establecerán planes de mitigación y planes de contingencia para ellos.

Riesgo	Plan de mitigación	Plan de contingencia
El sistema resulta ser más grande de lo que se tenía previsto	Hacer estimaciones más pequeñas a los distintos módulos para obtener una estimación más exacta.	Entrega evolutiva de módulos menos importantes para el sistema.
Grado de estimación de la seguridad muy bajo debido a la inexperiencia del grupo desarrollador.	Verificar las estimaciones más pequeñas para aumentar la precisión y experiencia.	Ninguno.
Base de datos muy grande, con muchas tablas.	Documentar y establecer normas para una fácil comprensión del modelo.	Ordenar lo obtenido.
Cambios Drásticos en el proceso productivo del vivero	Estar siempre informado de todos los nuevos cambios que afectan al proceso productivo	Reanalizar los módulos que se ven afectados para así realizar la manutención correspondiente
Conflictos en elección de herramientas.	Elegir herramientas de trabajo en conjunto.	Establecer herramientas conocidas.

Los clientes no comprenden el impacto de los cambios en los requerimientos	Preparar un documento que muestre al cliente el impacto en la planificación y el coste	Ninguno.
Falta de entusiasmo por parte de los usuarios finales con el proyecto.	Realizar los requerimientos de usuario con la mínima jerga técnica posible, proveer de un contacto que le permita comunicación con el supervisor de los subsistemas. Esto hace que los usuarios sientan que sus opiniones son tomadas en cuenta.	Ninguno.
Inasistencia a reuniones por parte de los miembros del equipo debido a trabajo	Reorganizar el equipo de forma que haya traslape en el trabajo y las personas comprendan el de los demás.	Ninguno.
Se proponen cambios en los requerimientos que requieren rehacer el diseño	Valorar el impacto de los requerimientos nuevos y maximizar la información oculta en ellos.	Ninguno.
Falta de comprensión de los requerimientos por parte de los desarrolladores	Listar los requerimientos con la mínima abstracción posible para evitar ambigüedades.	Ninguno.
Priorización inadecuada de requerimientos de poco interés	Listar los requerimientos en forma conjunta con el cliente	Ninguno

para el cliente	acotando bien el problema	
-----------------	---------------------------	--

6.3.- Paradigma

A continuación se explicaran los paradigmas con sus técnicas, ventajas y desventajas entre los cuales se podría elegir el desarrollo de este proyecto para luego dar a conocer la elección nuestro equipo y el porqué.

6.3.1.- Programación Extrema (Metodologías Agiles XP)

¿En qué consiste XP y cuáles son sus objetivos?

La programación extrema se basa en la simplicidad, la comunicación y el reciclado continuo de código, para algunos no es más que aplicar una pura lógica.

Los objetivos de XP son muy simples: la satisfacción del cliente. Esta metodología trata de dar al cliente el software que él necesita y cuando lo necesita. Por tanto, debemos responder muy rápido a las necesidades del cliente, incluso cuando los cambios sean al final de ciclo de la programación.

El segundo objetivo es potenciar al máximo el trabajo en grupo. Tanto los jefes de proyecto, los clientes y desarrolladores, son parte del equipo y están involucrados en el desarrollo del software.

6.3.1.1.- Fases de la Programación Extrema XP

Existen diversas prácticas inherentes al desarrollo de software a continuación de nombran y explican las referentes a xp.

6.3.1.1.1.- Planificación.

XP plantea la planificación como un permanente dialogo entre las partes la empresarial (Deseable) y la técnica (posible). Las personas del negocio necesitan determinar:

- **Ámbito:** ¿Qué es lo que el software debe de resolver para que este genere valor?

- **Prioridad:** ¿Qué debe ser hecho en primer lugar?
- **Composición de versiones:** ¿Cuánto es necesario hacer para saber si el negocio va mejor con software que sin el? En cuanto el software aporte algo al negocio debemos de tener lista las primeras versiones.
- **Fechas de versiones:** ¿Cuáles son las fechas en la presencia del software o parte del mismo pudiese marcar la diferencia ? El personal del negocio no puede tomar en vacío estas decisiones, y el personal técnico tomará las decisiones técnicas que proporcionan la materia prima para las decisiones del negocio.
- **Estimaciones:** ¿Cuánto tiempo lleva implementar una característica?
- **Consecuencias:** Informar sobre las consecuencias de la toma de decisiones por parte del negocio. Por ejemplo el cambiar las bases de datos a Oracle.
- **Procesos:** ¿Cómo se organiza el trabajo y el equipo?
- **Programación detallada:** Dentro de una versión ¿Qué problemas se resolverán primero?

6.3.1.1.2.- Pequeñas Versiones.

Cada versión debe de ser tan pequeña como fuera posible, conteniendo los requisitos de negocios más importantes, las versiones tiene que tener sentido como un todo, no puedes implementar media característica y lanzar la versión.

Es mucho mejor planificar para 1 mes o 2 que para seis meses y un año, las compañías que entregan software muy voluminoso no son capaces de hacerlo con mucha frecuencia.

6.3.1.1.3.- Hacer pruebas.

No debe existir ninguna característica en el programa que no haya sido probada, los programadores escriben pruebas para chequear el correcto funcionamiento del programa,

los clientes realizan pruebas funcionales. El resultado un programa más seguro que conforme pasa el tiempo es capaz de aceptar nuevos cambios.

6.3.1.1.4.- Re-codificación.

Cuando se implementan nuevas características en los programas se plantea la manera de hacerlo lo más simple posible, después de implementar esta característica, nos preguntamos cómo hacer el programa más simple sin perder funcionalidad, este proceso se le denomina recodificar o refactorizar (refactoring). Esto a veces nos puede llevar a hacer más trabajo del necesario, pero a la vez estaremos preparando nuestro sistema para que en un futuro acepte nuevos cambios y pueda albergar nuevas características. No debemos de recodificar ante especulaciones si no solo cuándo el sistema te lo pida.

6.3.1.1.5.- Programación Por Parejas.

Todo el código de producción lo escriben dos personas frente al ordenador, con un sólo ratón y un sólo teclado. Cada miembro de la pareja juega su papel: uno codifica en el ordenador y piensa la mejor manera de hacerlo, el otro piensa mas estratégicamente, ¿Va a funcionar?, ¿Puede haber pruebas donde no funcione?, ¿Hay forma de simplificar el sistema global para que el problema desaparezca ?. El emparejamiento es dinámico, puedo estar emparejado por la mañana con una persona y por la tarde con otra, si tienes un trabajo sobre un área que no conoces muy bien puedes emparejarte con otra persona que si conozca ese área. Cualquier miembro del equipo se puede emparejar con cualquiera.

6.3.1.1.6.- Integración Continúa.

El código se debe integrar como mínimo una vez al día, y realizar las pruebas sobre la totalidad del sistema. Una pareja de programadores se encargara de integrar todo el código en una maquina y realizar todas las pruebas hasta que estas funcionen al 100%.

6.3.1.1.7.- Cliente In-Situ.

Un cliente real debe sentarse con el equipo de programadores, estar disponible para responder a sus preguntas, resolver discusiones y fijar las prioridades. Lo difícil es que el cliente nos ceda una persona que conozca el negocio para que se integre en el equipo normalmente estos elementos son muy valiosos, pero debemos de hacerles ver que será

mejor para su negocio tener un software pronto en funcionamiento, y esto no implica que el cliente no pueda realizar cualquier otro trabajo.

6.3.1.1.8.- En Definitiva XP Supone:

- Las personas son claves en los procesos de desarrollo.
- Los programadores son profesionales no necesitan supervisión.
- Los procesos se aceptan y se acuerdan, no se imponen.
- Desarrolladores y gerentes comparten el liderazgo del proyecto.
- El trabajo de los desarrolladores con las personas que conocen el negocio es regular, no puntual.

6.3.2.- Rational Unified Process (RUP)

El Proceso Unificado de Rational (RUP, el original inglés Rational Unified Process) es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. RUP es en realidad un refinamiento realizado por Rational Software del más genérico Proceso Unificado.

RUP es un proceso para el desarrollo de un proyecto de un software que define claramente quien, cómo, cuándo y qué debe hacerse en el proyecto. Como 3 características esenciales está dirigido por los Casos de Uso: que orientan el proyecto a la importancia para el usuario y lo que este quiere, está centrado en la arquitectura: que Relaciona la toma de decisiones que indican cómo tiene que ser construido el sistema y en qué orden, y es iterativo e incremental: donde divide el proyecto en mini proyectos donde los casos de uso y la arquitectura cumplen sus objetivos de manera más depurada.

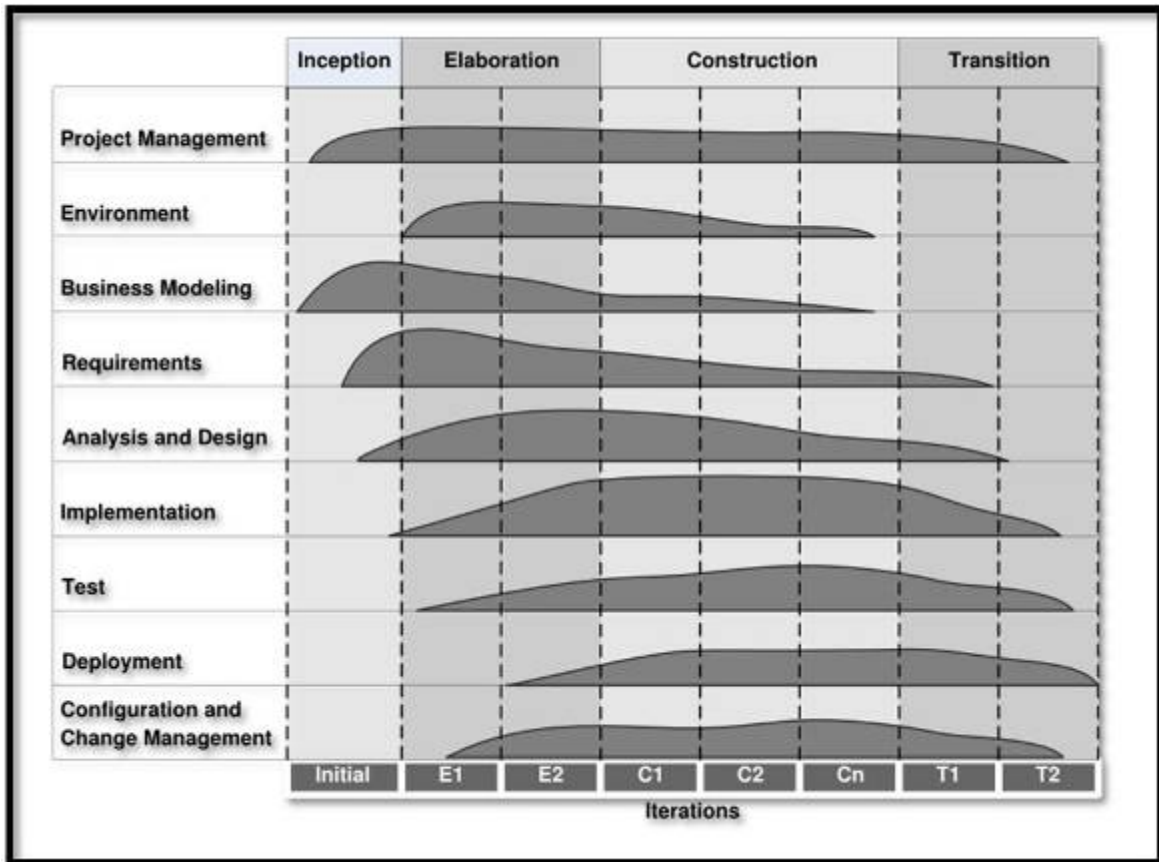
El RUP está basado en 6 principios claves:

1. **Adaptar el proceso:** El proceso deberá adaptarse a las características propias del proyecto u organización. El tamaño del mismo, así como su tipo o las regulaciones que lo condicionen, influirán en su diseño específico.

2. **Balancear prioridades:** Los requerimientos de los diversos inversores pueden ser diferentes, contradictorios o disputarse recursos limitados. Debe encontrarse un balance que satisfaga los deseos de todos.
3. **Colaboración entre equipos:** El desarrollo de software no lo hace una única persona sino múltiples equipos. Debe haber una comunicación fluida para coordinar requerimientos, desarrollo, evaluaciones, planes, resultados, etc.
4. **Demostrar valor iterativamente:** Los proyectos se entregan, aunque sea de un modo interno, en etapas iteradas. En cada iteración se analiza la opinión de los inversores, la estabilidad y calidad del producto, y se refina la dirección del proyecto.
5. **Elevar el nivel de abstracción:** Este enfoque ha ofrecido una alternativa muy productiva para construir aplicaciones.
6. **Enfocarse en la calidad:** El control de calidad no debe realizarse al final de cada iteración, sino en todos los aspectos de la producción.

Ciclo de vida

Imagen 12: Un típico perfil de proyecto mostrando el tamaño relativo de las cuatro fases



El RUP divide el proceso de desarrollo en ciclos, dentro de los cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en los distintas actividades. Teniendo un producto final al final de cada ciclo, cada ciclo se divide en fases que finalizan con un hito donde se debe tomar una decisión importante:

- **Inicio:** se hace un plan de fases, se identifican los principales casos de uso y se identifican los riesgos.
- **Elaboración:** se hace un plan de proyecto, se completan los casos de uso y se eliminan los riesgos.
- **Construcción:** se concentra en la elaboración de un producto totalmente operativo y eficiente y el manual de usuario.

- **Transición:** se implementa el producto en el cliente y se entrena a los usuarios. Como consecuencia de esto suelen surgir nuevos requisitos a ser analizados.

6.3.2.1.- Descripción De Las Actividades

Dependiendo de las iteraciones del proceso el equipo de desarrollo puede realizar 7 tipos de actividades en este:

6.3.2.1.1.- Fase De Inicio

Durante la fase de inicio las iteraciones hacen poner mayor énfasis en actividades modelado del negocio y de requisitos.

6.3.2.1.2.- Modelado Del Negocio

En esta fase el equipo se familiarizará más al funcionamiento de la empresa, sobre conocer sus procesos.

- Entender la estructura y la dinámica de la organización para la cual el sistema va ser desarrollado.
- Entender el problema actual en la organización objetivo e identificar potenciales mejoras.
- Asegurar que clientes, usuarios finales y desarrolladores tengan un entendimiento común de la organización objetivo.

6.3.2.1.3.- Requisitos

En esta línea los requisitos son el contrato que se debe cumplir, de modo que los usuarios finales tienen que comprender y aceptar los requisitos que especifiquemos.

- Establecer y mantener un acuerdo entre clientes y otros stakeholders sobre lo que el sistema podría hacer.
- Proveer a los desarrolladores un mejor entendimiento de los requisitos del sistema.
- Definir el ámbito del sistema.
- Proveer una base para estimar costos y tiempo de desarrollo del sistema.

- Definir una interfaz de usuarios para el sistema, enfocada a las necesidades y metas del usuario.

6.3.2.1.4.- Fase De Elaboración

En la fase de elaboración, las iteraciones se orientan al desarrollo de la baseline de la arquitectura, abarcan más los flujos de trabajo de requerimientos, modelo de negocios (refinamiento), análisis, diseño y una parte de implementación orientado a la baseline de la arquitectura.

6.3.2.1.5.- Análisis y Diseño

En esta actividad se especifican los requerimientos y se describen sobre cómo se van a implementar en el sistemas

- Transformar los requisitos al diseño del sistema.
- Desarrollar una arquitectura para el sistema.
- Adaptar el diseño para que sea consistente con el entorno de implementación

6.3.2.1.6.- Fase De Construcción

6.3.2.1.6.1 Implementación

Se implementan las clases y objetos en ficheros fuente, binarios, ejecutables y demás. El resultado final es un sistema ejecutable.

- Planificar qué subsistemas deben ser implementados y en qué orden deben ser integrados, formando el Plan de Integración.
- Cada implementador decide en qué orden implementa los elementos del subsistema.
- Si encuentra errores de diseño, los notifica.
- Se integra el sistema siguiendo el plan.

6.3.2.1.6.2 Pruebas

Este flujo de trabajo es el encargado de evaluar la calidad del producto que estamos desarrollando, pero no para aceptar o rechazar el producto al final del proceso de desarrollo, sino que debe ir integrado en todo el ciclo de vida.

- Encontrar y documentar defectos en la calidad del software.
- Generalmente asesora sobre la calidad del software percibida.
- Provee la validación de los supuestos realizados en el diseño y especificación de requisitos por medio de demostraciones concretas.
- Verificar las funciones del producto de software según lo diseñado.
- Verificar que los requisitos tengan su apropiada implementación.

6.3.2.1.6.3 Despliegue

Esta actividad tiene como objetivo producir con éxito distribuciones del producto y distribuirlo a los usuarios. Las actividades implicadas incluyen:

- Probar el producto en su entorno de ejecución final.
- Empaquetar el software para su distribución.
- Distribuir el software.
- Instalar el software.
- Proveer asistencia y ayuda a los usuarios.
- Formar a los usuarios y al cuerpo de ventas.
- Migrar el software existente o convertir bases de datos.

Principales características

- Forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo)

- Pretende implementar las mejores prácticas en Ingeniería de Software
- Desarrollo iterativo
- Administración de requisitos
- Uso de arquitectura basada en componentes
- Control de cambios
- Modelado visual del software
- Verificación de la calidad del software

El RUP es un producto de Rational (IBM). Se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. Incluye artefactos (que son los productos tangibles del proceso como por ejemplo, el modelo de casos de uso, el código fuente, etc.) y roles (papel que desempeña una persona en un determinado momento, una persona puede desempeñar distintos roles a lo largo del proceso).

6.3.3.- Elección De Un Paradigma

Aunque xp no es un mal paradigma en este caso se encontraron muchas trabas para poder utilizarlo por ejemplo el tiempo de trabajo semanal y diario, que el cliente esté disponible al 100% para el equipo de desarrollo o que facilite a un empleado que conozca el flujo de trabajo es imposible , En base a lo expuesto anteriormente se optado por el se opta por usar el Rational Unified Process (RUP) [2]., rescatando sus mejores y más valiosas ventajas por ejemplo la modularización del problema en sí, y por su desarrollo incremental e iterativo.

6.4.- Metodología

Es correcto afirmar que el paradigma establece las actividades necesarias para la obtención de software, sin embargo el paradigma no es el encargado de decidir como efectuar éstas actividades, esto es responsabilidad de la metodología. Las principales estrategias o metodologías para enfrentar cada tarea del proceso de análisis de sistemas, se pueden agrupar en dos grandes categorías:

6.4.1.- Análisis Estructurado

Es un método para el análisis de sistemas manuales o automatizados, que conduce al desarrollo de especificaciones para sistemas nuevos o para efectuar modificaciones a los ya existentes[2]., Éste análisis permite al analista conocer un sistema o proceso en una forma lógica y manejable al mismo tiempo que proporciona la base para asegurar que no se omite ningún detalle pertinente.

Componentes:

Símbolos gráficos: Iconos y convenciones para identificar y describir los componentes de un sistema junto con las relaciones entre estos componentes.

Diccionario de datos: descripción de todos los datos usados en el sistema. Puede ser manual o automatizado.

Descripciones de procesos y procedimientos: declaraciones formales que usan técnicas y lenguajes que permiten a los analistas describir actividades importantes que forman parte del sistema.

Reglas: estándares para describir y documentar el sistema en forma correcta y completa.

Ventajas del análisis estructurado: Los analistas deben trabajar con los usuarios para hacerles comprender el funcionamiento del sistema actual y el sistema futuro, para ello se hace aconsejable utilizar un lenguaje común, sencillo y fiable, estas son las características de los diagramas de flujo de datos. Los usuarios pueden hacer sugerencias para modificar los diagramas con la finalidad de describir las actividades con mayor exactitud, y permitirá evitar los errores desde el inicio pudiendo prevenir una posible falla del sistema.

6.4.2.- Análisis Orientado A Objetos

Define los programas en términos de "clases de objetos", objetos que son entidades que combinan estado (es decir, datos), comportamiento (esto es, procedimientos o métodos) e identidad (propiedad del objeto que lo diferencia del resto). La programación orientada a objetos expresa un programa como un conjunto de estos objetos, que colaboran entre ellos

para realizar tareas. Esto permite hacer los programas y módulos más fáciles de escribir, mantener y reutilizar.

Algunas de las mayores ventajas de la metodología de orientación a objetos son:

- Documenta, sin importar su implementación
- Abstrae el mundo real, por lo que es más fácilmente entendible
- Está más orientada a la interacción con el usuario, que a procesos con entrada y salida
- Facilidad de dividir el sistema en varios subsistemas
- Fomenta la reutilización de componentes

6.4.3.- Elección De Una Metodología

En base a lo expuesto anteriormente y el poder de UM. L con la orientación a objetos la metodología elegida para ocupar es la metodología de orientación a objetos.

6.5.- Herramientas

A continuación presentamos todas las herramientas utilizadas en este proyecto separadas en las categorías en las cuales fueron empleadas

6.5.1.- Documentación

Todos los documentos que deben ser realizados en el desarrollo de este proyecto serán creados a través del procesador de textos MS WORD 2007.

6.5.2.- Diagramas

Para los diagramas se utilizaran las Herramientas MS Visio 2007 y SmartDraw 2007.

6.5.3.- Planificación de Proyecto

Para realizar la planificación temporal se utilizara MS Project 2007.

6.5.4.- Desarrollo e Implementación del Software

Como herramienta principal de programación se usara MS Visual Basic Express 2005

6.6.- Tecnologías.

6.6.1.- Servidor De Datos Sql Server

SQL Server es un sistema de gestión de bases de datos relacionales basado en el lenguaje SQL, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea.

Entre sus características figuran:

- Soporte de transacciones.
- Gran estabilidad.
- Gran seguridad.
- Escalabilidad.
- Soporta procedimientos almacenados.

Incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.

Permite trabajar en modo cliente-servidor donde la información y datos se alojan en el servidor y las terminales o que clientes de la red sólo accedan a la información. Además permite administrar información de otros servidores de datos.

Microsoft SQL Server constituye la alternativa de Microsoft a otros potentes sistemas gestores de bases de datos como son Oracle o Sybase.

Nuestra elección fue SQL Server 2005 Express es una versión diseñada para ayudar a los desarrolladores a construir aplicaciones robustas y fiables ofreciendo una sencilla pero

potente base de datos que es además gratuita. SQL Server Express ya está incluida como opción al instalar cualquier herramienta Visual Studio Express

6.6.1.1.- Compatibilidad De Las Características De SQL Server Express

SQL Server Express es compatible con la mayoría de las características y funciones de SQL Server 2005. A continuación se muestra una lista con las principales características y componentes que admite.

- Procedimientos almacenados
- Vistas
- Desencadenadores
- Cursores
- Utilidades **sqlcmd** y **osql**.
- Niveles de aislamiento de instantánea
- Compatibilidad con XML nativo, incluidos XQuery y esquemas XML
- Compatibilidad con lenguajes de Transact-SQL
- Conexión de administrador dedicada
- Administrador de configuración de SQL Server
- Réplica (sólo como suscriptor)
- Optimizador de consultas avanzado
- SMO/RMO
- Integración con Visual Studio 2005
- Service Broker (únicamente como cliente)

- CLR de SQL
- Conjuntos de resultados activos múltiples (MARS)

6.6.2.- Características De MS Visual Basic Express

Microsoft Visual Basic 2005. es una evolución del lenguaje Visual Basic que continúa la tradición de ofrecer una manera rápida y fácil de crear aplicaciones basadas en .NET Framework, permitiendo a los desarrolladores generar de manera productiva aplicaciones Windows, para la Web y para dispositivos móviles, con seguridad de tipos y orientados a objetos.

Esta versión contiene nuevas características para el desarrollo rápido de aplicaciones. Con ellas se permite: la eliminación garantizada de recursos, la sobrecarga de operadores, los tipos genéricos y los eventos personalizados, y el acceso rápido, gracias a la opción My, a las tareas frecuentes de .NET Framework relacionadas, por ejemplo, con la aplicación y su entorno durante la ejecución. Además proporciona interoperabilidad de lenguajes, recolección de objetos no referenciados, seguridad mejorada y control de versiones.

Las características del editor de Visual Basic reducen drásticamente los errores de programación en la fase de diseño, tanto para desarrolladores principiantes como avanzados. Al proporcionar una funcionalidad similar a la corrección ortográfica y gramatical de Microsoft Word, por el cual Visual Basic sugiere correcciones para los errores de sintaxis más habituales. Además el compilador advierte a los desarrolladores de la existencia de código semánticamente incorrecto que podría producir errores en tiempo de ejecución, como por ejemplo intentos de obtener acceso a elementos de código ante de la inicialización.

Visual Basic proporciona una depuración mejorada. Al incorporar “Editar y continuar”, los desarrolladores pueden modificar y comprobar el código fuente sin necesidad de detener y reiniciar la sesión de depuración. Este ciclo iterativo de desarrollo y depuración, junto con las funciones avanzadas de corrección de errores y análisis de código en modo de interrupción, proporcionan a los desarrolladores que utilizan Visual Basic la experiencia de depuración más eficaz y flexible que hayan conocido jamás.

Visual Basic mejora en gran medida la manipulación y recuperación de datos. Está disponible un diseño de datos simplificado desde el entorno de desarrollo para los datos locales y remotos. También incorpora la capacidad de crear aplicaciones enlazadas a datos sin escribir ni una sola línea de código. Ideal para numerosos escenarios de acceso a datos habituales, esta característica permitirá a los desarrolladores generar automáticamente un IU personalizable y enlazada a datos arrastrando y colocando una tabla o varias columnas en un formulario.

En resumen Visual Basic se centra en permitir el rápido desarrollo de aplicaciones que abarquen todos los niveles. Las características previstas en el depurador, los diseñadores visuales, el editor de código y el lenguaje incrementan en gran medida la productividad y permiten a los desarrolladores crear unas aplicaciones robustas y elegantes más rápidamente.

6.7.- Arquitectura Del Sistema

La arquitectura de software, es un conjunto de patrones y abstracciones coherentes que proporciona el marco de referencia necesario para guiar la construcción del software para un sistema de información. Para realizar una buena elección al momento de escoger la forma de programar el sistema debemos tomar en cuenta los siguientes puntos a parte de lo funcional:

- Mantenibilidad
- Auditabilidad
- Flexibilidad
- Interacción con otros sistemas

También debemos tener en cuenta las restricciones de la tecnología a utilizar, ya que hay arquitecturas que son recomendables con ciertas tecnologías y con otras no.

6.7.1.- Arquitectura Lógica Del Sistema

Generalmente, no es necesario inventar una nueva arquitectura de software para cada sistema de información. Lo habitual es adoptar una arquitectura conocida en función de sus ventajas e inconvenientes para cada caso en concreto

Por estas razones expuestas anteriormente damos paso al análisis de distintas arquitecturas con las cuales podríamos desarrollar el sistema “Software de control del proceso productivo para Vivero Ecogreen”.

En las arquitecturas estudiadas encontramos las siguientes:

- 1 .Monolítica
- 2 Cliente – Servidor
- 3 Arquitectura de 3 niveles

6.7.1.1.- Monolítica:

Esta arquitectura consiste en una estructura de grupos funcionales muy acoplados, en donde las funciones están en un solo lugar, capaces de lograr una performans de rapidez y velocidad en la respuesta, pero al momento de la mantenibilidad es muy complicado ya que cada cambio que se le realiza afecta a una o más funciones del sistema y hay que conocerlo muy bien en nivel de programación para no dejar un error muy grande al momento de mantener y modificar el sistema.

Para efectos del sistema en desarrollo, luego del análisis realizado con los casos de uso y la forma como se va a implementar, por efectos de mantenibilidad al código, en la interfaces de usuario y la seguridad de los datos creemos que esta arquitectura no es la apropiada para este proyecto.

6.7.1.2.- Arquitectura De 3 Niveles:

La arquitectura de n niveles consta de separa la lógica del negocio y la lógica del diseño, como ejem-plo se puede mostrar separar la capa de datos del cliente con la capa de datos. Cuando se habla de este tipo de arquitectura, la separación de los niveles en forma

óptima es de 3 niveles, en donde se conforman en la capa de presentación, capa de datos y la del negocio.

La ventaja más importante de esta forma de programación es que al momento de realizar cambio en una capa, solo afecta a esta y no se mezcla con las otra capas, logrando así una mantenibilidad del software muy fácil de realizar sin mezclar los distintos códigos de las capas, Además permite distribuir el trabajo de creación de una aplicación por niveles, de este modo, cada grupo de trabajo está totalmente abstraído del resto de niveles, simplemente es necesario conocer la API que existe entre niveles.

Es una de las arquitecturas más utilizada [11] hoy en día.

Análisis de por capa

1.- Capa de presentación: es la que ve el usuario (hay quien la denomina "capa de usuario"), presenta el sistema al usuario, le comunica la información y captura la información del usuario dando un mínimo de proceso (realiza un filtrado previo para comprobar que no hay errores de formato). Esta capa se comunica únicamente con la capa de negocio.

2.- Capa de negocio: es donde residen los programas que se ejecutan, recibiendo las peticiones del usuario y enviando las respuestas tras el proceso [12]. Se denomina capa de negocio (e incluso de lógica del negocio) pues es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos para almacenar o recuperar datos de él.

3.- Capa de datos: es donde residen los datos. Está formada por uno o más gestor de bases de datos que realiza todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

6.7.1.3.- Cliente Servidor:

Esta arquitectura consiste en separar las funciones del sistema en dos entes que entran en comunicación, en donde el servidor puede tener las funciones del sistema o sea, las

funcionalidades propias del sistema, los datos, la forma de ver la inserción de datos sea correcta y el cliente se configura con las funciones propias del cliente [13], como las interfaces, las peticiones al servidor, etc. entonces el cliente pide información o servicios al servidor, el cual procesa la información de petición enviada por los clientes y entrega los servicios pedidos a los clientes. En muchos casos hay que identificar bien las funciones que se deben dejar en el servidor y en el cliente, lo cual no es un trabajo trivial ya que para efectos de mantenibilidad debemos lograr que las funciones dada al cliente.

Ventajas

- Centralización del control: Los accesos, recursos y la integridad de los datos son controlados por el servidor de forma que un programa cliente defectuoso o no autorizado no pueda dañar el sistema. Esta centralización también facilita la tarea de poner al día datos u otros recursos.
- Escalabilidad: Se puede aumentar la capacidad de clientes y servidores por separado. Cualquier elemento puede ser mejorado en cualquier momento, o se pueden añadir nuevos nodos a la red (clientes y/o servidores).
- Fácil mantenimiento: Al estar distribuidas las funciones y responsabilidades entre varios ordenadores independientes, es posible reemplazar, reparar, actualizar, o incluso trasladar un servidor, mientras que sus clientes no se verán afectados por ese cambio o se afectarán mínimamente. Esta independencia de los cambios también se conoce como encapsulación.
- Existen tecnologías, suficientemente desarrolladas, diseñadas para el paradigma de Cliente Servidor que aseguran la seguridad en las transacciones, la amigabilidad del interfaz, y la facilidad de empleo.

Desventajas

- La congestión del tráfico ha sido siempre un problema en el paradigma de C/S. Cuando una gran cantidad de clientes envían peticiones simultáneas al mismo servidor, puede ser que cause muchos problemas para éste. Al contrario, en las redes P2P como cada nodo en la red hace también de servidor, cuanto más nodo hay, mejor es el ancho de banda se tiene.

- El paradigma de Cliente Servidor clásico no tiene la robustez de una red P2P. Cuando un servidor está caído, las peticiones de los clientes no pueden ser satisfechas. En la mayor parte de redes P2P, los recursos están generalmente distribuidos en varios nodos de la red. Aunque algunos salgan o abandonen la descarga; otros pueden todavía acabar de descargar consiguiendo datos del resto de los nodos en la red.
- El software y el hardware de un servidor son generalmente muy determinantes. Un hardware regular de un ordenador personal puede no poder servir a cierta cantidad de clientes. Normalmente se necesita software y hardware específico, sobre todo en el lado del servidor, para satisfacer el trabajo. Por supuesto, esto aumentará los costos.

6.7.1.4.- Arquitectura Elegida.

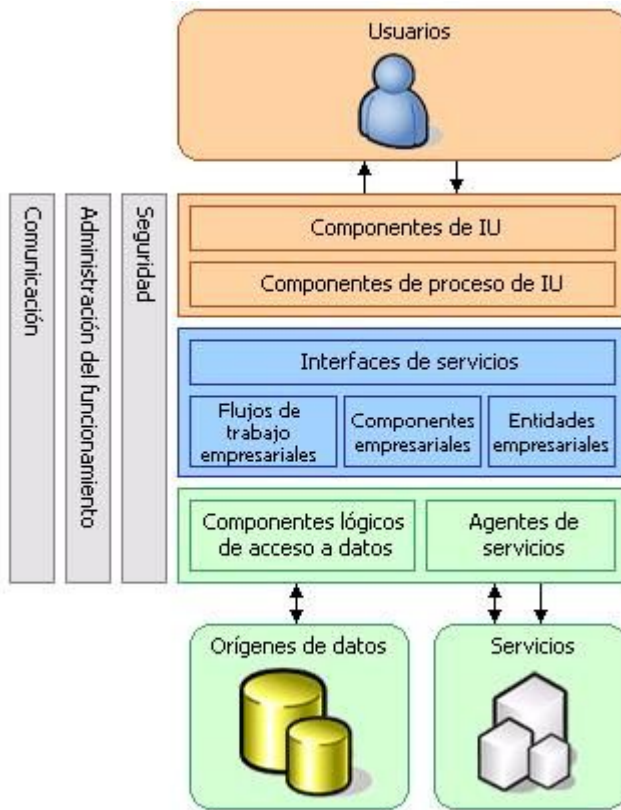
Considerando el análisis realizado del sistema que se va a construir, la arquitectura de 3 niveles es la que más se acomoda a los requerimientos y los resultados en cuanto a respuesta van a ser óptimas. Además como se explican anteriormente, la forma de programar es separada por capas, lo que hace que la manutención del código del software se realiza a la capa y sin afectar las otras.

Además las tecnologías en que se va desarrollar el sistema se adaptan muy bien a esta arquitectura, ya que visual estudio y su lenguaje VB permite este tipo de programación, lo que facilita del desarrollo.

En cuanto a la seguridad, los datos en el servidor correspondiente es mucho más seguro que tenerlos en cada Terminal ya que se deben almacenar los pagos de los empleados y las ventas de la empresa

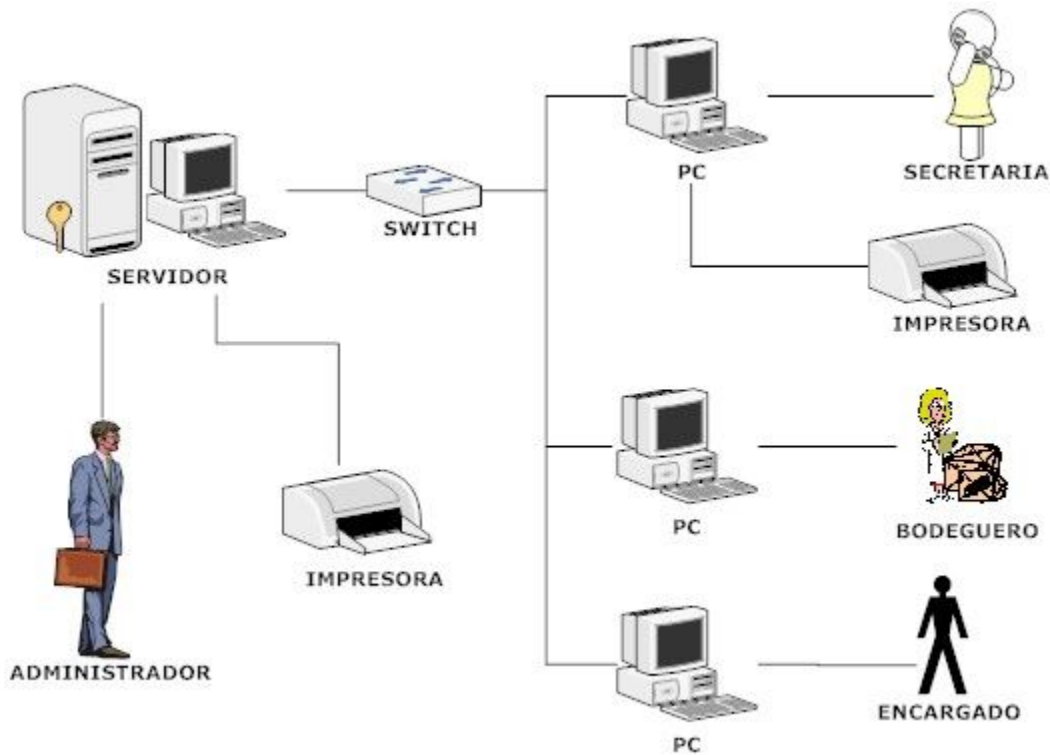
Por los distintos puntos dados en esta conclusión, la arquitectura a utilizar será la de 3 capas.

Imagen 13: Arquitectura 3 niveles



6.7.2.- Arquitectura Física Del Sistema

Imagen 14: Arquitectura Física Del Sistema



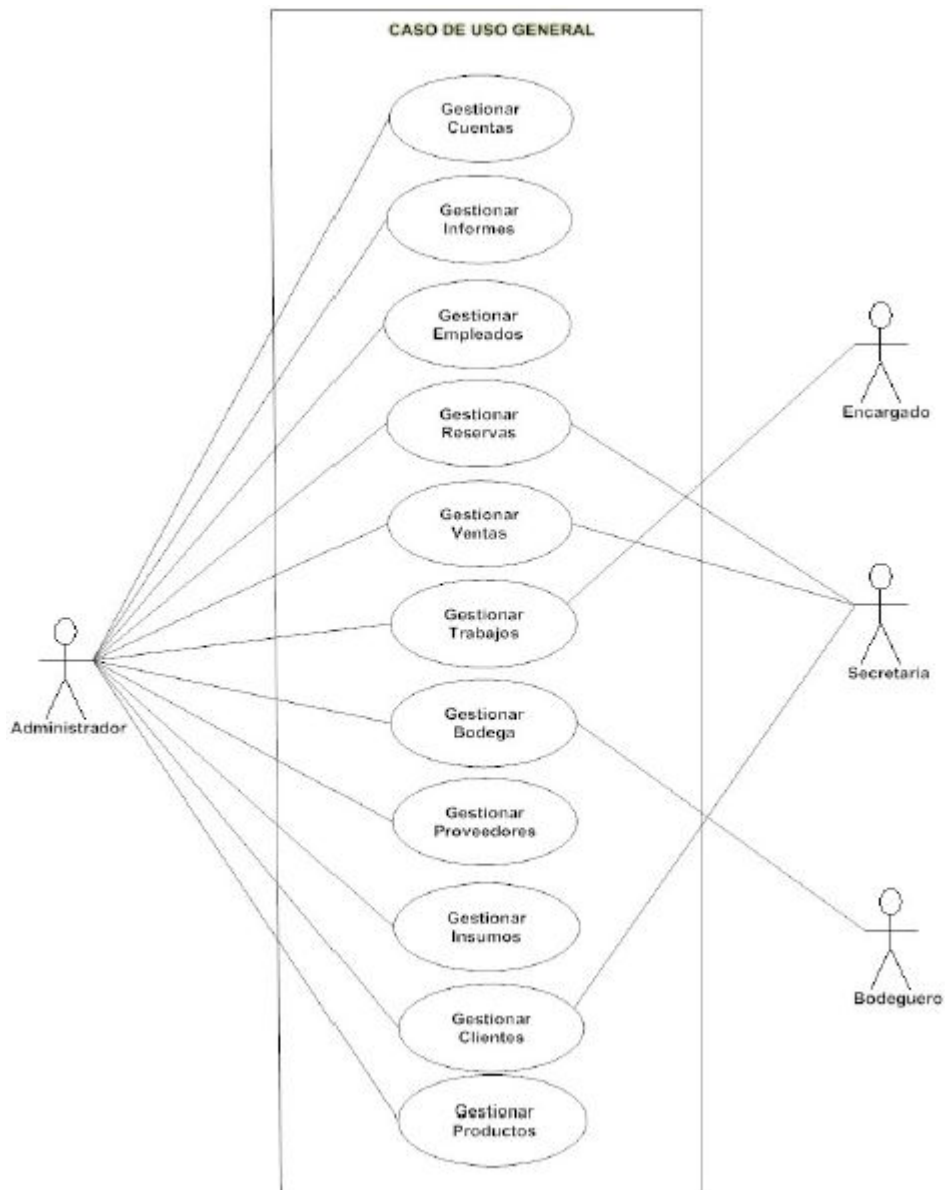
La arquitectura física está dada por las limitantes físicas del hardware en las instalaciones correspondientes de nuestro cliente las cuales son las siguientes:

- Athlon xp 2.6 GHz, 256 Mb de RAM, disco duro de 80 Gigas.
- Athlon xp 2.6 GHz, 512 Mb de RAM, disco duro de 60 Gigas.
- Athlon xp 2.6 GHz, 512 Mb de RAM, disco duro de 60 Gigas.
- Dos impresoras Epson Stylus C92
- El Switch DLink DES-1005D

6.8.- Casos De Uso

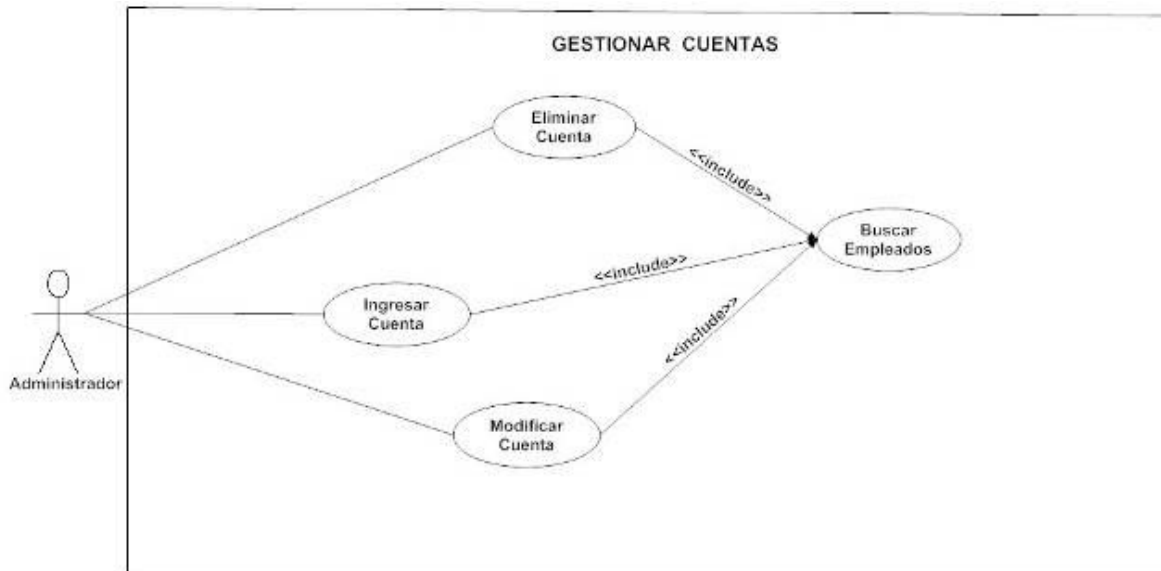
A continuación se mostrara el modelo de caso de uso general, producto del análisis de las reuniones con el cliente, la cual será por la elección de paradigma la base del desarrollo del proyecto.

Imagen 15: Diagrama De Caso De Uso General



6.8.1.- Gestionar Cuentas

Imagen 16: Caso de Uso Gestión Cuentas



6.8.1.1.- Caso De Uso Eliminar Cuenta (Descripción)

Nombre:	Eliminar Cuentas
Descripción:	Permite realizar el proceso a través del cual se eliminaran los datos de de la cuenta de un empleado de la empresa.
Actores:	Administrador
Pre-condiciones:	Administrador logeado y confirmado
Post-condiciones:	Base de datos actualizada

6.8.1.2.- Caso De Uso Ingresar Cuenta (Descripción)

Nombre:	Ingresar Cuentas
----------------	-------------------------

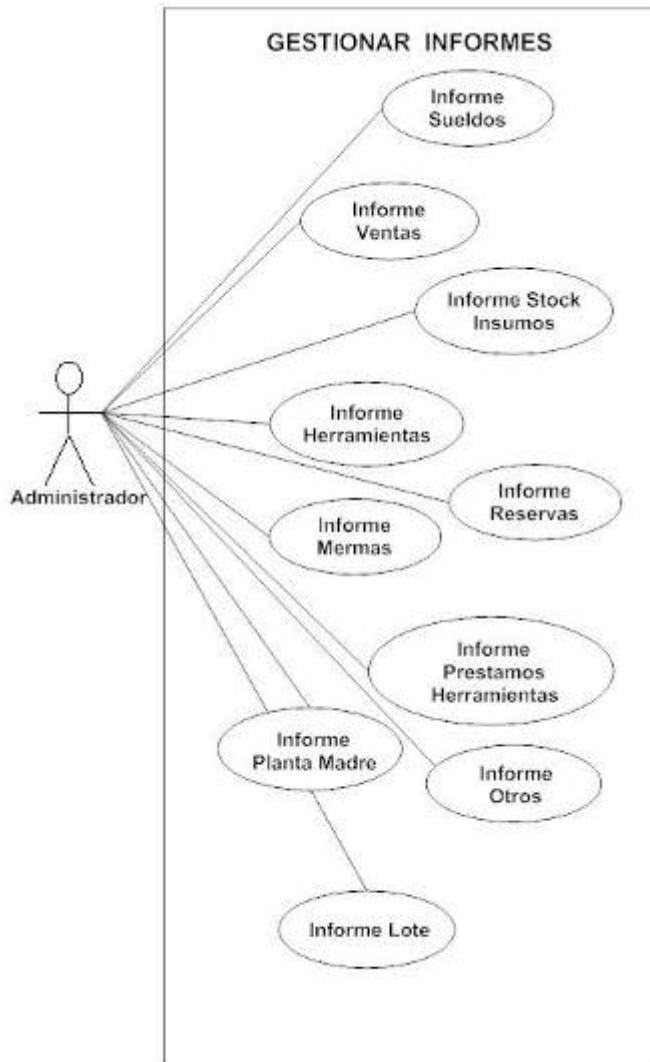
Descripción:	Permite realizar el proceso a través del cual se ingresaran las nuevas cuentas privadas correspondientes a un empleado de la empresa.
Actores:	Administrador
Pre-condiciones:	Administrador logeado y confirmado
Post-condiciones:	Base de datos actualizada con los datos de las nuevas cuentas

6.8.1.3.- Caso De Uso Modificar Cuentas (Descripción)

Nombre:	Modificar Cuentas
Descripción:	Permite realizar el proceso a través del cual se modifican las cuentas privadas de un empleado de la empresa.
Actores:	Administrador
Pre-condiciones:	Administrador logeado y confirmado
Post-condiciones:	Base de datos actualizada con las correspondientes modificaciones a la cuentas de los empleados

6.8.2.- Informes

Imagen 17: Caso De Uso Informes



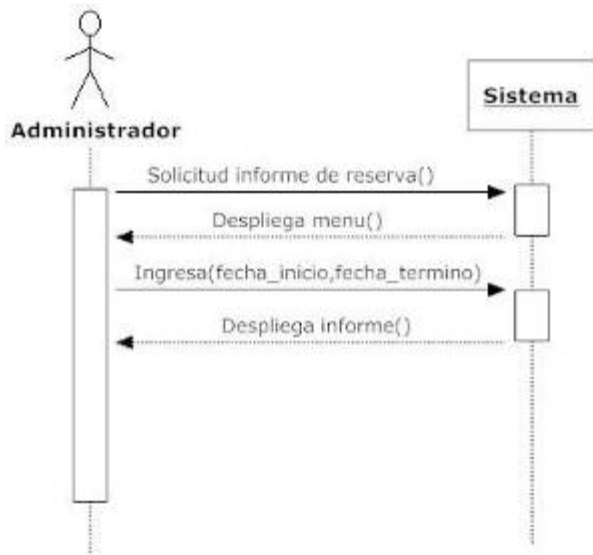
6.8.2.1.- Caso De Uso Informe Reserva

Nombre:	Informe de reserva
Descripción:	Generar un informe de reservas a la empresa que demuestre las situaciones actuales de sus productos
Actores:	Administrador
Pre-condiciones:	Administrador logeado y confirmado

Flujo Principal:	
<ol style="list-style-type: none"> 1. Administrador solicita obtener un informe de reserva 2. Sistema despliega opciones de informe de reserva 3. El administrador filtra los datos por fecha inicio, fecha termino o todas las reservas 4. Sistema despliega informe de reserva. 5. Administrador acepta y cierra. 	
Flujo Alternativo:	
<ol style="list-style-type: none"> 2. sistema no tiene datos para mostrar 1. Administrador acepta y cierra. 5.a Administrador imprime informe 1. Administrador ingresa opción imprimir 2. Sistema pide confirmación. 3. Administrador confirma. 	
Post-condiciones:	

6.8.2.2.- Diagrama De Secuencia Caso De Uso Informe Reserva

Imagen 18: Diagrama De Secuencia Caso De Uso Informe Reserva



6.8.2.3.- Caso De Uso Informe Ventas

Nombre:	Informe de ventas
Descripción:	Generar un informe de ventas
Actores:	Administrador
Pre-condiciones:	Administrador logeado y confirmado
Flujo Principal:	
<ol style="list-style-type: none"> 1. Administrador solicita obtener un informe de ventas 2. Sistema despliega opciones de informe de ventas 3. El administrador ingresa fecha de inicio y fecha de termino y selecciona informe completo 4. Sistema despliega informe de ventas. 5. Administrador acepta y cierra. 	
Flujo Alternativo:	

3.a Administrador selecciona informe detalle venta.

1. sistema despliega detalle de venta y continua si ejecución normal

5.a Administrador imprime informe

1. Administrador ingresa opción imprimir

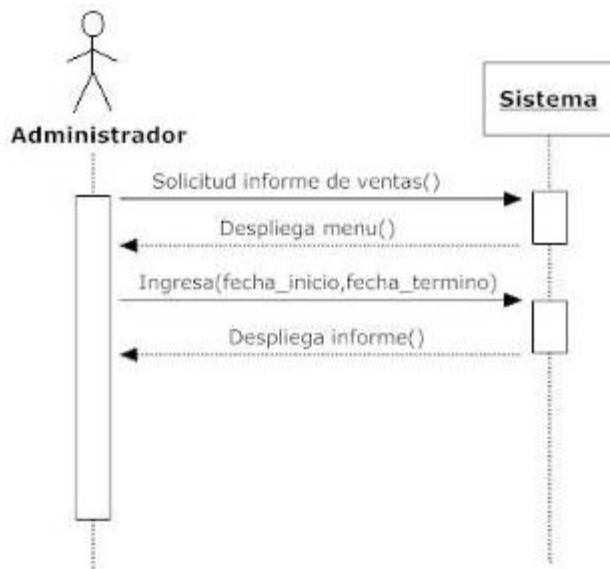
2. Sistema pide confirmación.

3. Administrador confirma.

Post-condiciones:

6.8.2.4.- Diagrama De Secuencia Caso De Uso Informe Ventas

Imagen 19: Diagrama De Secuencia Caso De Uso Informe Ventas



6.8.2.5.- Caso De Uso Informe Herramientas

Nombre:

Informe de herramientas

Descripción:	Generar un informe de herramientas
Actores:	Administrador
Pre-condiciones:	Administrador logeado y confirmado
Flujo Principal:	
<ol style="list-style-type: none"> 1. Administrador solicita obtener un informe de herramientas 2. Sistema despliega opciones de informe de herramientas 3. El administrador filtra por opción informe completo(todos) 4. Sistema despliega informe de herramientas. 5. Administrador acepta y cierra. 	
Flujo Alternativo:	
<p>3.a El informe es por herramienta tipo merma</p> <ol style="list-style-type: none"> 1. Sistema despliega informe de herramientas tipo merma. 2. continúa en paso 4. <p>3.b El informe es por herramienta tipo ocupado</p> <ol style="list-style-type: none"> 1. Sistema despliega informe de herramientas tipo ocupado. 2. continúa en paso 4. <p>3.a El informe es por herramienta tipo libre</p> <ol style="list-style-type: none"> 1. Sistema despliega informe de herramientas tipo libre. 	

2. continúa en paso 4.

5.a Administrador imprime informe

1. Administrador ingresa opcion imprimir

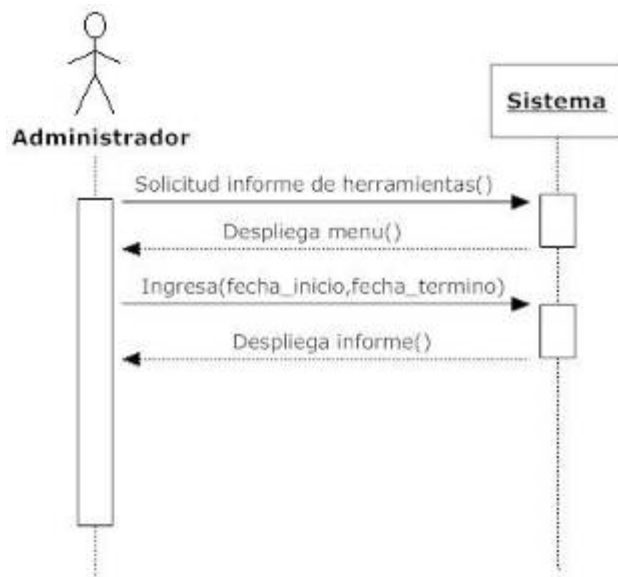
2. Sistema pide confirmación.

3. Administrador confirma.

Post-condiciones:

6.8.2.6.- Diagrama Secuencia Caso De Uso Informe Herramientas

Imagen 20: Diagrama De Secuencia Caso De Uso Informe Herramientas



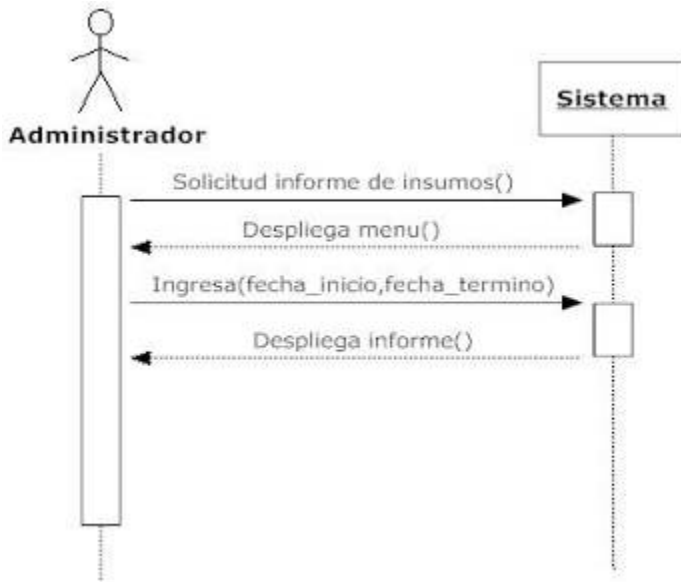
6.8.2.7.- Caso De Uso Informe Insumos

Nombre:	Informe de insumos
Descripción:	Generar un informe de insumos

Actores:	Administrador
Pre-condiciones:	Administrador logeado y confirmado
Flujo Principal:	
<ol style="list-style-type: none"> 1. Administrador solicita obtener un informe de insumos 2. Sistema despliega opciones de informe de insumos 4. Sistema despliega informe de insumos. 5. Administrador acepta y cierra. 	
Flujo Alternativo:	
<p>5.a Administrador imprime informe</p> <ol style="list-style-type: none"> 1. Administrador ingresa opcion imprimir 2. Sistema pide confirmacion. 3. Administrador confirma. 	
Post-condiciones:	

6.8.2.8.- Diagrama Secuencia Caso De Uso Informe Insumos

Imagen 21: Diagrama De Secuencia Caso De Uso Informe Insumos



6.8.2.9.- Caso De Uso Informe Mermas

Nombre:	Informe de mermas
Descripción:	Generar un informe de mermas
Actores:	Administrador
Pre-condiciones:	Administrador logeado y confirmado
Flujo Principal:	
<ol style="list-style-type: none"> 1. Administrador solicita obtener un informe de mermas 2. Sistema despliega opciones de informe de mermas 3. El administrador ingresa fecha de inicio y fecha de termino 4. Sistema despliega informe de mermas. 5. Administrador acepta y cierra. 	

Flujo Alternativo:

5.a Administrador imprime informe

1. Administrador ingresa opcion imprimir

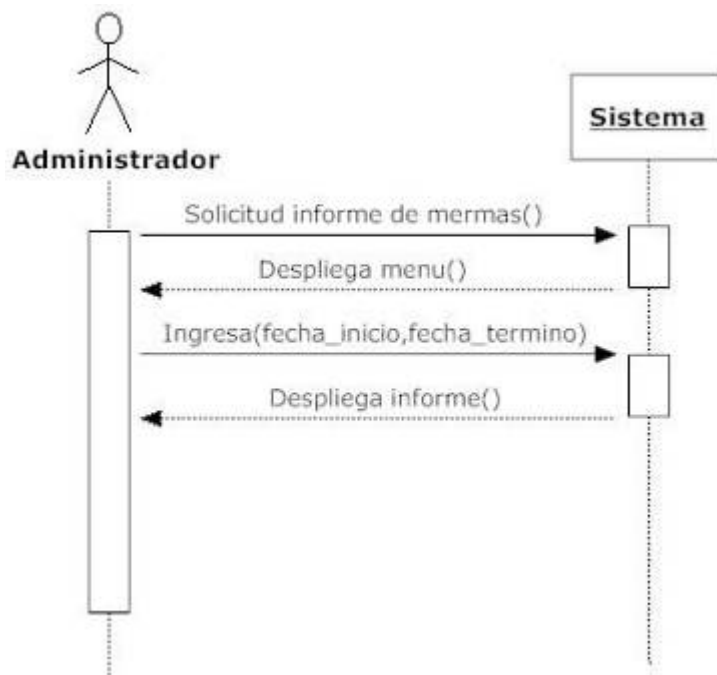
2. Sistema pide confirmacion.

3. Administrador confirma.

Post-condiciones:

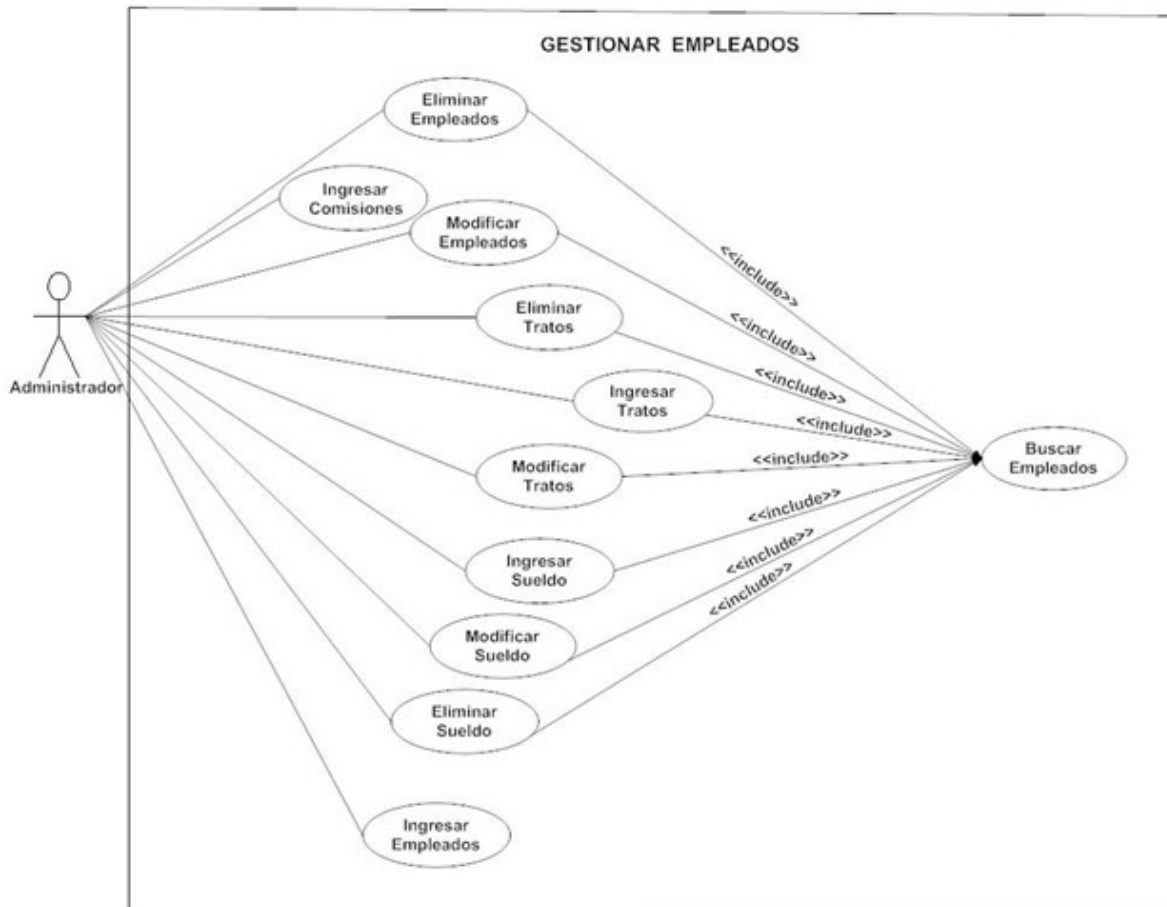
6.8.2.10.- Diagrama Secuencia Caso De Uso Informe Mermas

Imagen 22: Diagrama De Secuencia Caso De Uso Informe Mermas



6.8.3.- Caso De Uso Gestión Empleado

Imagen 23: Caso De Uso Gestión Empleados



6.8.3.1.- Caso De Uso Buscar Empleados

Nombre:	Buscar Empleados
Descripción:	Permite encontrar un empleado en la base.
Actores:	Administrador
Pre-condiciones:	Administrador logeado y confirmado

Flujo Principal:

1. Administrador hace clic en el menú, opción “búsqueda de empleado”.
2. Sistema despliega el modulo de búsqueda, un formulario para introducir datos respectivos (nombre) del empleado a buscar.
3. Sistema realiza búsqueda contra la base y despliega los resultados en forma de malla de datos, donde se ve un conjunto de filas y columnas que contienen todos los datos* del o los empleados que coincidieron con el termino buscado.

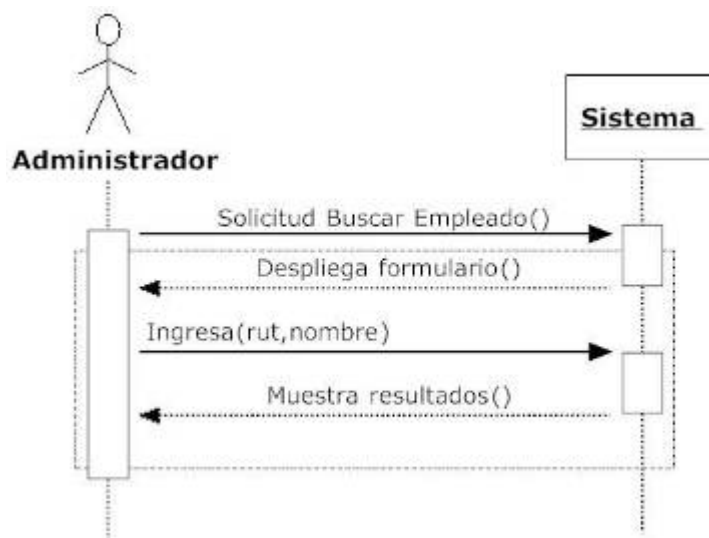
*Todos los datos: nombre, rut, teléfono, etc.

Flujo Alternativo:

Post-condiciones: No hay.

6.8.3.2.- Diagrama Secuencia Caso De Uso Buscar Empleados

Imagen 24: Diagrama De Secuencia Caso De Uso Buscar Empleado

**6.8.3.3.- Caso De Uso Ingresar Comisiones (descripción)**

Nombre:	Ingresar Comisiones
Descripción:	Permite asignar las comisiones de los empleados dependiendo de su tipo.
Actores:	Administrador
Pre-condiciones:	Administrador logeado y confirmado
Post-condiciones:	Comisiones por tipo empleado actualizadas

6.8.3.4.- Caso De Uso Modificar Empleado

Nombre:	Modificar Empleado
Descripción:	Permite realizar el proceso a través del cual se modifica los datos de un empleado.
Actores:	Administrador
Pre-condiciones:	Administrador logeado y confirmado

Flujo Principal:

1. Usuario hace clic en el menú, opción “Modificar empleado”
2. El sistema presenta el formulario de modificación de empleados
3. El administrador busca el empleado al cual desea modificar datos(ver caso de uso buscar empleados)
4. El sistema despliega los datos encontrados
5. El usuario modifica los campos necesarios y hace clic en aceptar.
6. El sistema actualiza la base.

Flujo Alternativo:

3.a Empleado no encontrado

1 Termina de la operación

5.a El usuario cancela.

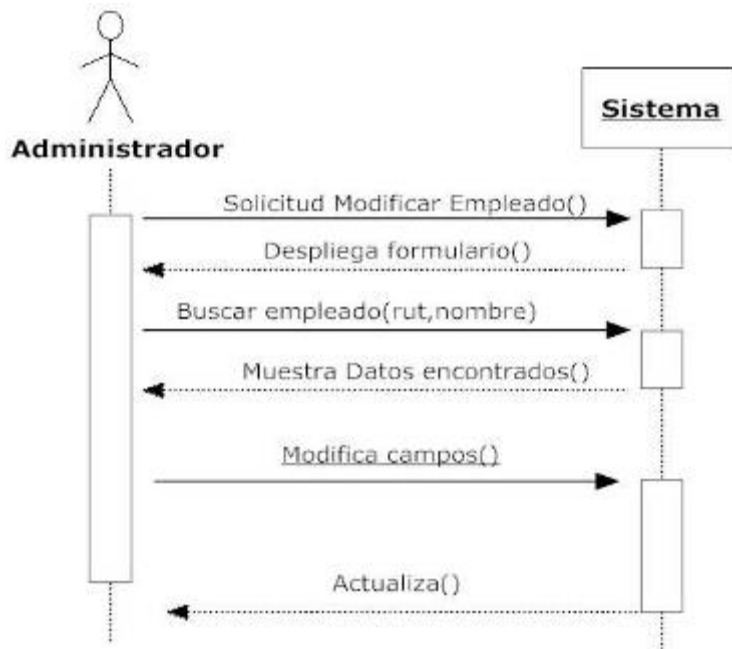
1 El sistema no actualiza

Post-condiciones:

Base actualizada.

6.8.3.5.- Diagrama secuencia Caso De Uso Modificar Empleado

Imagen 25: Diagrama De Secuencia Caso De Uso Modificar Empleados



6.8.3.6.- Caso De Uso Ingresar Cuenta (descripción)

Nombre:	Ingresar Cuenta
Descripción:	A través de este caso de uso de pueden Ingresar todas las cuentas de futuros usuarios del sistema
Actores:	Administrador
Pre-condiciones:	Administrador logeado y confirmado
Post-condiciones:	Base de datos actualizada

6.8.3.7.- Caso De Uso Eliminar Cuenta (descripción)

Nombre:	Eliminar Cuenta
----------------	------------------------

Descripción:	A través de este caso de uso de pueden eliminar todas las cuentas previamente ingresadas.
Actores:	Administrador
Pre-condiciones:	Administrador logeado y confirmado
Post-condiciones:	Base de datos actualizada

6.8.3.8.- Caso De Uso Eliminar Cuenta (descripción)

Nombre:	Modificar Cuenta
Descripción:	Permite realizar el proceso a través del cual se modifica los datos de una cuenta previamente ingresada en el sistema.
Actores:	Administrador
Pre-condiciones:	Administrador logeado y confirmado
Post-condiciones:	Base de datos actualizada con las modificaciones correspondientes a los trabajos

6.8.3.9.- Caso De Uso Ingresar Trato (descripción)

Nombre:	Ingresar Tratos
Descripción:	Permite realizar el proceso a través del cual se ingresaran los tratos llevados por los empleados al sistema.

Actores:	Administrador
Pre-condiciones:	Administrador logeado y confirmado
Post-condiciones:	Base de datos actualizada

6.8.3.10.- Caso De Uso Eliminar Trato (descripción)

Nombre:	Eliminar Tratos
Descripción:	Permite realizar el proceso a través del cual se eliminan los tratos llevados por los empleados.
Actores:	Administrador
Pre-condiciones:	Administrador logeado y confirmado
Post-condiciones:	Base de datos actualizada

6.8.3.11.- Caso De Uso Modificar Trato (descripción)

Nombre:	Modificar Tratos
Descripción:	Permite realizar el proceso a través del cual se modifican los tratos llevados por los empleados al sistema.
Actores:	Administrador
Pre-condiciones:	Administrador logeado y confirmado

Post-condiciones:	Base de datos actualizada con las modificaciones correspondientes a los trabajos
--------------------------	--

6.8.3.12.- Caso De Uso Ingresar Sueldo (descripción)

Nombre:	Ingresar Sueldos
Descripción:	En este caso de uso se ingresan los sueldos de los empleados al sistema.
Actores:	Administrador
Pre-condiciones:	Administrador logeado y confirmado
Post-condiciones:	Base de datos actualizada

6.8.3.13.- Caso De Uso Modificar Sueldo (descripción)

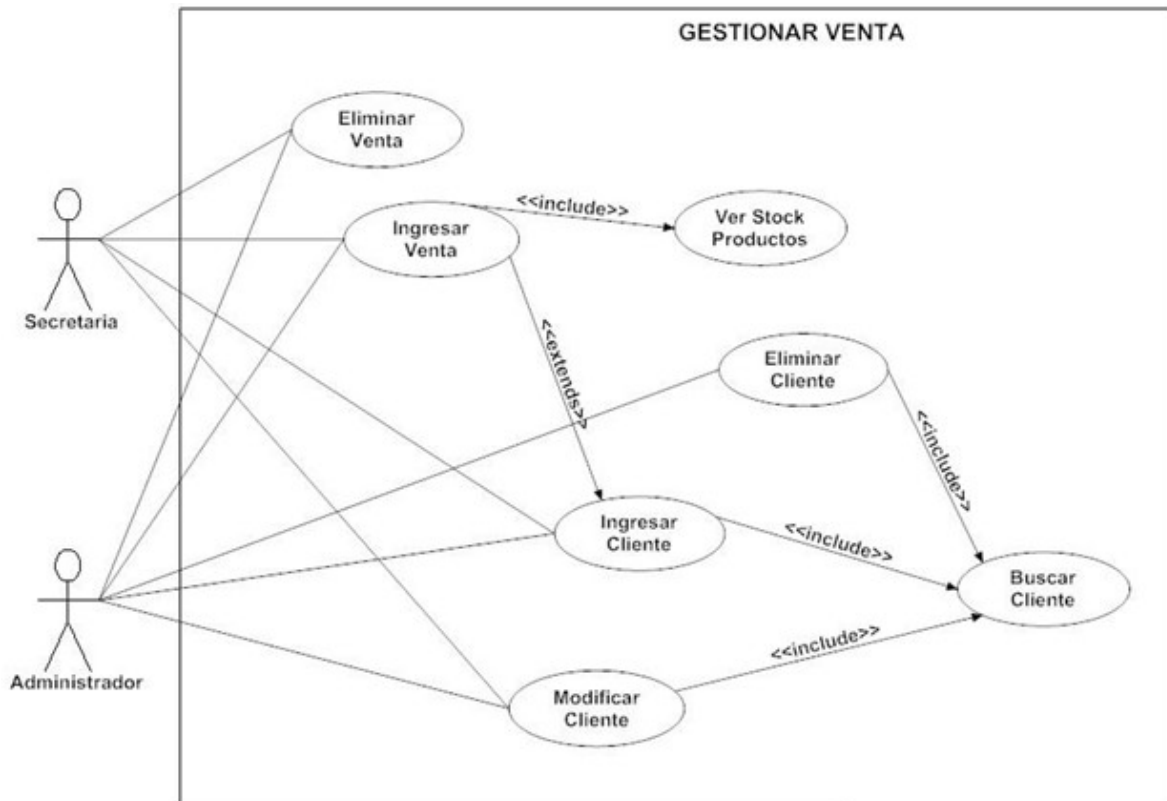
Nombre:	Modificar Sueldos
Descripción:	Permite realizar el proceso a través del cual se modifican los datos de un sueldo previamente ingresado
Actores:	Administrador
Pre-condiciones:	Administrador logeado y confirmado
Post-condiciones:	Base de datos actualizada con las modificaciones correspondientes a los trabajos

6.8.3.14.- Caso De Uso Eliminar Sueldo (descripción)

Nombre:	Eliminar Sueldos
Descripción:	Permite realizar el proceso a través del cual se elimina un sueldo previamente ingresado en el sistema
Actores:	Administrador
Pre-condiciones:	Administrador logeado y confirmado
Post-condiciones:	Base de datos actualizada

6.8.4.- Caso De Uso Reservas

Imagen 26: Caso De Uso Reservas



6.8.4.1.- Caso De Uso Pasar a Venta (descripción)

Nombre:	Pasar a Venta
Descripción:	Permite buscar y mostrar las reservas realizadas para poder hacer la entrega de los productos y pasar aquella reserva a una venta.
Actores:	Administrador
Pre-condiciones:	Administrador logeado y confirmado
Post-condiciones:	

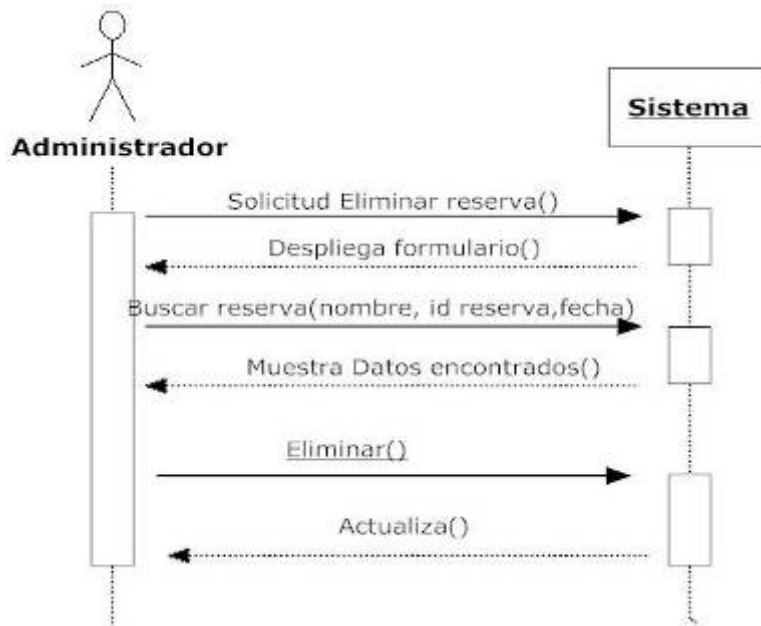
6.8.4.2.- Caso De Uso Eliminar Reserva

Nombre:	Eliminar Reserva
Descripción:	Permite eliminar la reserva realizada por un cliente vip de la empresa.
Actores:	Administrador
Pre-condiciones:	Administrador logeado y confirmado

Flujo Principal:	
<ol style="list-style-type: none"> 1. Administrador hace clic en el menú, opción “Eliminar Reserva”. 2. Sistema despliega el modulo de búsqueda, un formulario para introducir datos respectivos (rut cliente o nombre o id reserva) a buscar la respectivas reservas. 3. Sistema realiza búsqueda contra la base y despliega los resultados en forma de malla de datos, donde se ve un conjunto de filas y columnas que contienen todos los datos de los lotes reservados. 4. El administrador selecciona la reservas respectivas que desea borrar y presiona el eliminar 5. El sistema actualiza la base de datos 	
Flujo Alternativo:	
<p>3.a Sistema no encuentra la reserva deseada deseado</p> <p>1 vuelve al paso 2</p> <p>6.a Administrador no confirma y cancela la operación</p>	
Post-condiciones:	Base de datos actualizada

6.8.4.3.- Diagrama De Secuencia Caso De Uso Eliminar Reserva

Imagen 27: Diagrama De Secuencia Caso De Uso Eliminar Reserva



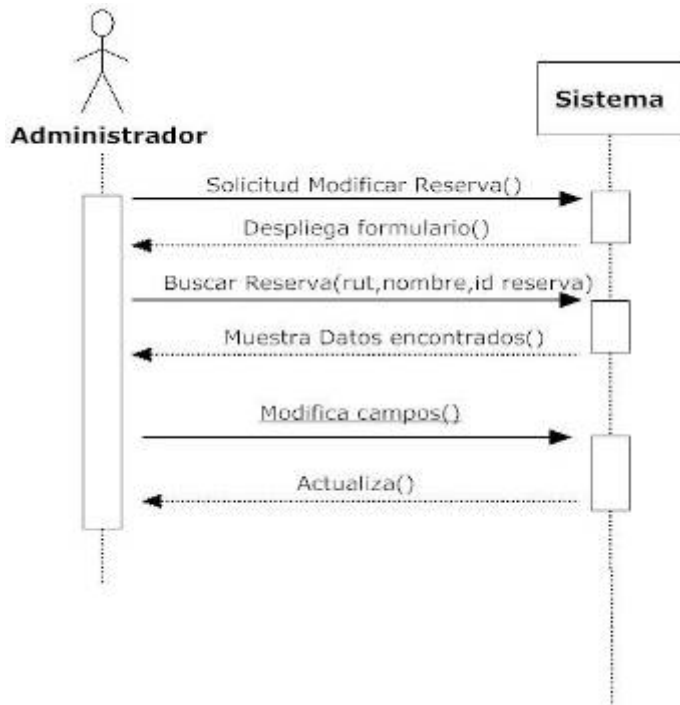
6.8.4.4.- Caso De Uso Modificar Reserva

Nombre:	Modificar Reserva
Descripción:	Permite realizar el proceso a través del cual se modifica los datos de una reserva
Actores:	Administrador
Pre- condiciones:	Administrador logeado y confirmado

Flujo Principal:	
<ol style="list-style-type: none"> 1. Usuario hace clic en el menú, opción “Modificar Cliente” 2. Sistema despliega el modulo de búsqueda, un formulario para introducir datos respectivos (nombre o nombre especie) a buscar su respectiva reserva. 3. Sistema realiza búsqueda contra la base y despliega los resultados en forma de malla de datos, donde se ve un conjunto de filas y columnas que contienen todos los lotes reservados de las reservas realizadas por este. 4. El Administrador selecciona reserva y modifica los campos necesarios y hace clic en aceptar. 5. El sistema actualiza la base de datos 	
Flujo Alternativo:	
<ol style="list-style-type: none"> 2.a Reserva no encontrada 1 Termina de la operación 6.a El Administrador cancela. 1 El sistema no actualiza 	
Post- condiciones:	<p>Base actualizada.</p> <p>Stock de insumo actualizado</p>

6.8.4.5.- Diagrama De Secuencia Caso De Uso Modificar Reserva

Imagen 28: Diagrama De Secuencia Caso De Uso Modificar Reserva



6.8.4.6.- Caso De Uso Ingresar Reserva

Nombre:	Ingresar Reserva
Descripción:	Permite realizar el proceso a través del cual se Ingresan los datos de una Reserva.
Actores:	Administrador
Pre-condiciones:	Administrador logeado y confirmado

Flujo Principal:

1. Administrador hace clic en el menú, opción “Ingresar Reservas”
2. El administrador busca los datos del cliente que desea realizar la reserva (ver caso de uso buscar clientes).
3. El sistema despliega todos los datos del cliente.
4. El Administrador busca y selecciona un lote
5. El sistema despliega el stock de productos
6. El Administrador ingresa los nuevos datos de la reserva y acepta..
7. El sistema actualiza la base.

Flujo Alternativo:

2.a Cliente no es vip

1 Termina la operación.

2.b Cliente no encontrado

1 Termina la operación.

3.a No hay stock de producto

1 Termina la operación.

8. a El Administrador cancela.

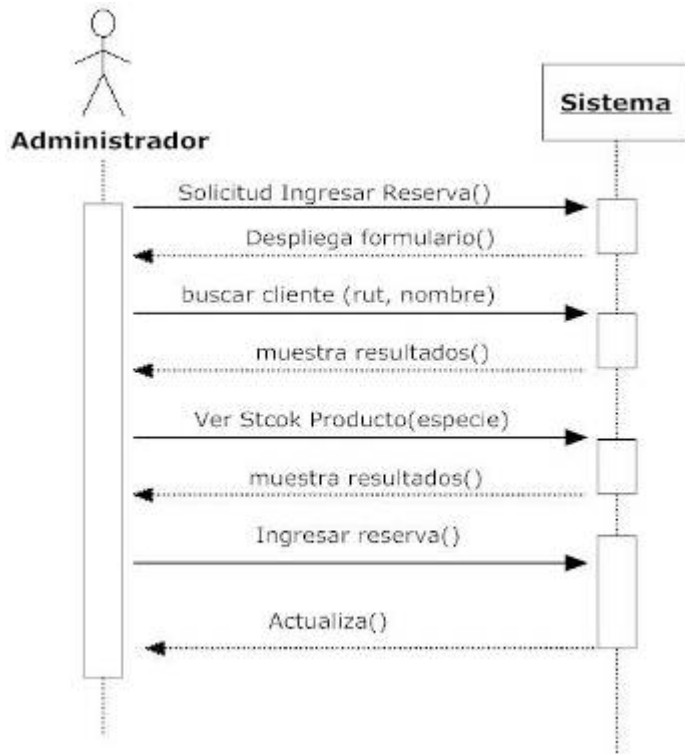
1 El sistema no actualiza

Post-condiciones:

Base actualizada.

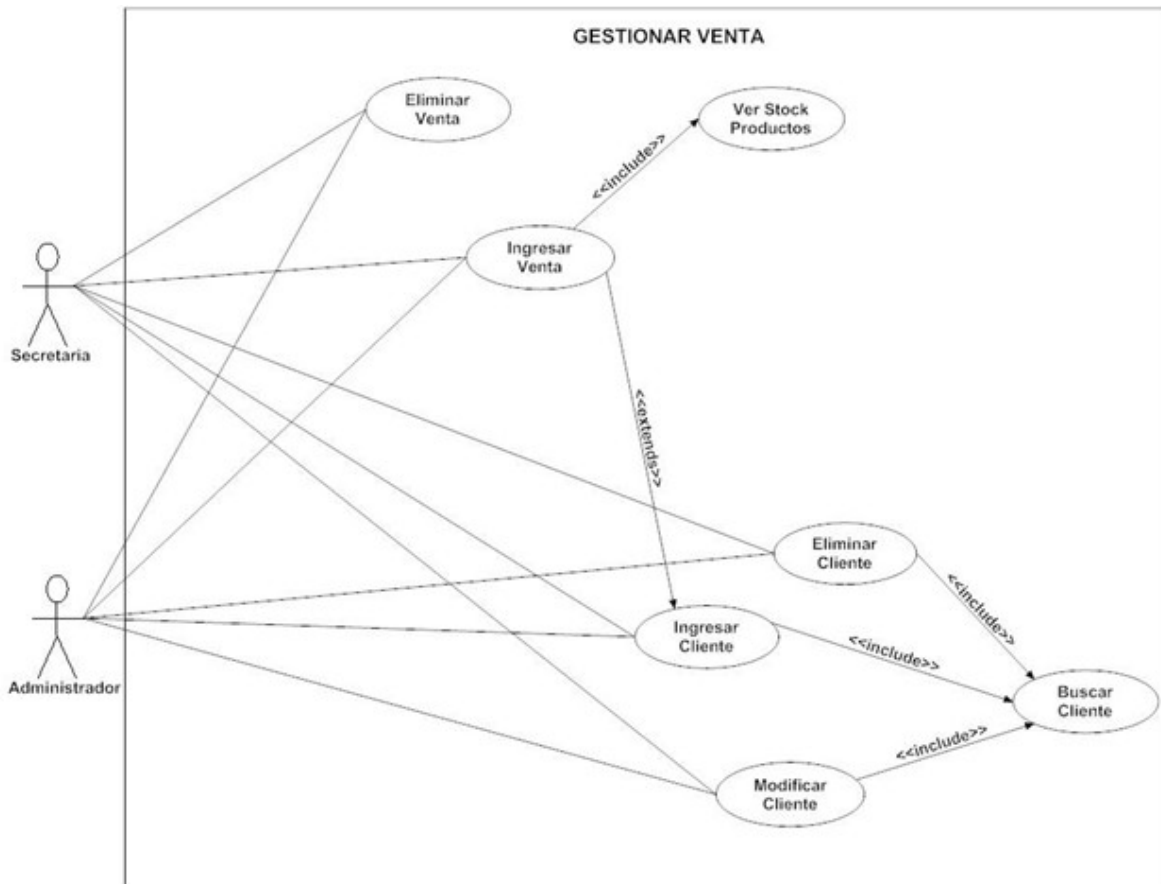
6.8.4.7.- Diagrama De Secuencia Caso De Uso Ingresar Reserva

Imagen 29: Diagrama De Secuencia Caso De Uso Ingresar Reserva



6.8.5.- Caso De Uso Ventas

Imagen 30: Diagrama De Secuencia Caso De Uso Ventas



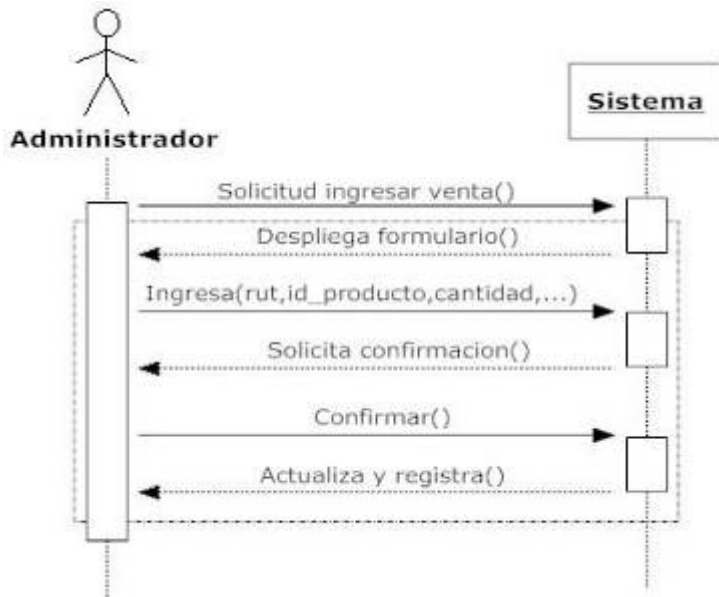
6.8.5.1.- Caso De Uso Ingresar Venta

Nombre:	Ingresar Venta
Descripción:	Permite Ingresar una nueva venta
Actores:	Administrador
Pre-condiciones:	Administrador logeado y confirmado
Flujo Principal:	<ol style="list-style-type: none"> 1. Administrador solicita creación de una nueva venta. 2. Sistema despliega formulario necesario para ingresar una nueva venta

<p>3. Administrador completa los campos solicitados</p> <p>4. Ir al caso de uso Ver Stock Insumo</p> <p>5. Sistema actualiza y almacena la nueva venta</p> <p>El administrador repite los pasos del 2 al 7 hasta que estime necesario</p>	
<p>Flujo Alternativo:</p> <p>3.a No hay suficiente Stock</p> <p>1. Se termina el caso de uso</p> <p>3a. El cliente es nuevo</p> <p>1. Ir al caso de uso Ingresar cliente.</p>	
<p>Post-condiciones:</p>	<p>No hay.</p>

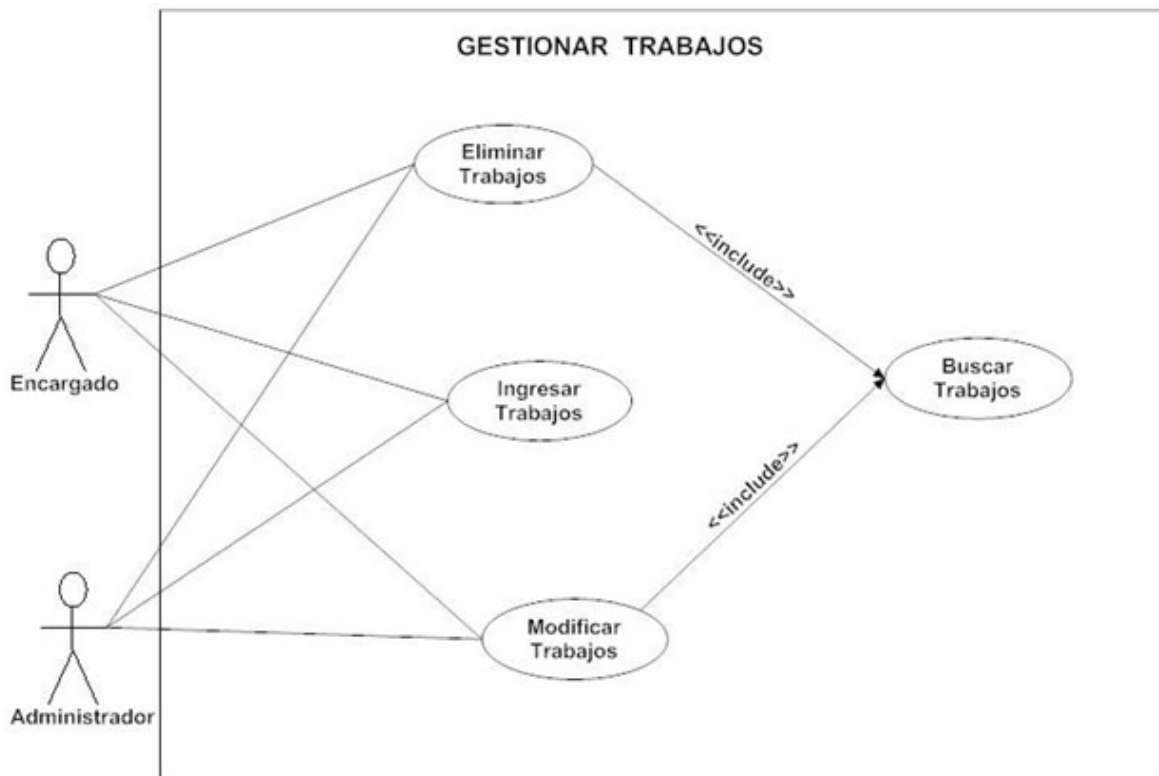
6.8.5.1.1.- Diagrama Secuencia Caso De Uso Ingresar Venta

Imagen 31: Diagrama De Secuencia Caso De Uso Ingresar Venta



6.8.6.- Caso De Uso Gestión Trabajos

Imagen 32: Diagrama De Secuencia Caso De Uso Informe Reserva



6.8.6.1.- Caso De Uso Eliminar Trabajos (Descripción)

Nombre:	Eliminar Trabajos
Descripción:	Permite realizar el proceso a través del cual se eliminarán los datos de los trabajos realizados en la empresa en un periodo de tiempo determinado.
Actores:	Administrador
Pre-condiciones:	Administrador logeado y confirmado
Post-condiciones:	Base de datos actualizada

6.8.6.2.- Caso De Uso Ingresar Trabajo (Descripción)

Nombre:	Ingresar Trabajos
Descripción:	Permite realizar el proceso a través del cual se ingresarán los nuevos datos de los trabajos realizados (empleado, lote, insumos) en la empresa en un día determinado.
Actores:	Administrador
Pre-condiciones:	Administrador logeado y confirmado
Post-condiciones:	Base de datos actualizada con los datos de los nuevos trabajos

6.8.6.3.- Caso De Uso Modificar Trabajo (Descripción)

Nombre:	Modificar Trabajos
----------------	---------------------------

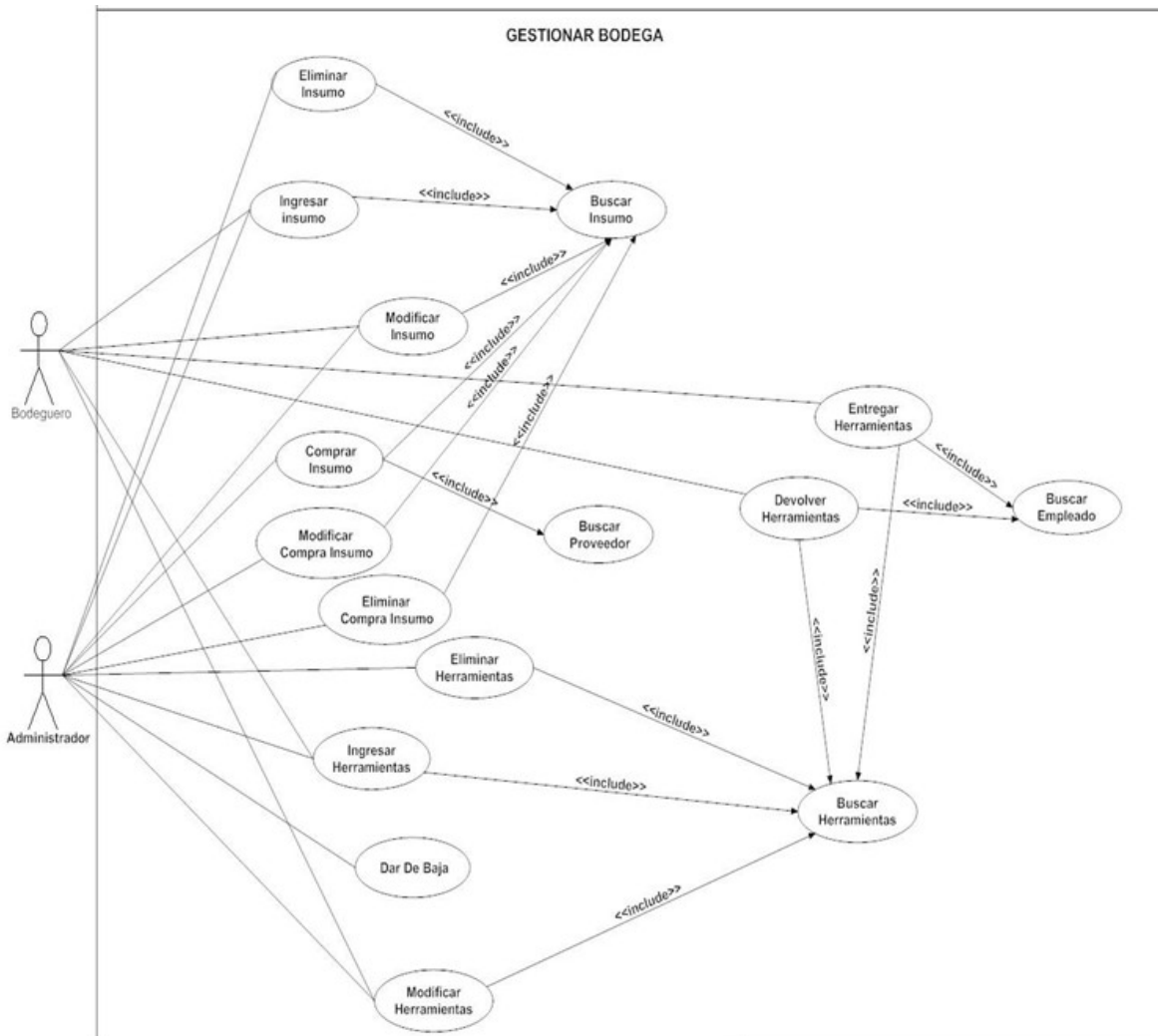
Descripción:	Permite realizar el proceso a través del cual se modifica los datos de un trabajo realizado en la empresa.
Actores:	Administrador
Pre-condiciones:	Administrador logeado y confirmado
Post-condiciones:	Base de datos actualizada con las modificaciones correspondientes a los trabajos

6.8.6.4.- Caso De Uso Buscar Trabajo (Descripción)

Nombre:	Buscar Trabajo
Descripción:	Permite realizar el proceso a través del cual se buscaran los datos de un trabajo realizado en la empresa en cierto periodo de tiempo
Actores:	Administrador
Pre-condiciones:	Administrador logeado y confirmado
Post-condiciones:	

6.8.7.- Caso De Uso Bodega

Imagen 33: Caso De Uso Gestionar Bodega



6.8.7.1.- Caso De Uso Dar De Baja

Nombre:	Dar De Baja
Descripción:	Permite buscar y mostrar las herramientas de la empresa
Actores:	Administrador
Pre-condiciones:	Administrador logeado y confirmado

Flujo Principal:

1. Administrador hace clic en el menú, opción "Ver Herramientas".
2. Sistema despliega el modulo de búsqueda, un formulario para introducir datos respectivos de la herramienta a mostrar y acepta.
3. El sistema muestra todos los resultados en forma de malla de datos del tipo de herramienta deseada

Flujo Alternativo:

- 2.a Administrador hace clic en cancelar y termina el caso de uso sin ver las herramientas.

Post-condiciones:	No hay.
--------------------------	---------

6.8.7.2.- Caso De Uso Buscar Herramientas

Nombre:	Buscar Herramientas
Descripción:	Permite encontrar una herramienta en la base.
Actores:	Administrador
Pre-condiciones:	Administrador logeado y confirmado

Flujo Principal:

1. Administrador hace clic en el menú, opción “búsqueda de Herramientas”.
2. Sistema despliega el modulo de búsqueda, un formulario para introducir datos respectivos (id herramienta o nombre herramienta) de la herramienta a buscar.
3. Sistema realiza búsqueda contra la base y despliega los resultados en forma de malla de datos, donde se ve un conjunto de filas y columnas que contienen todos los datos* del o las herramientas que coincidieron con el termino buscado.

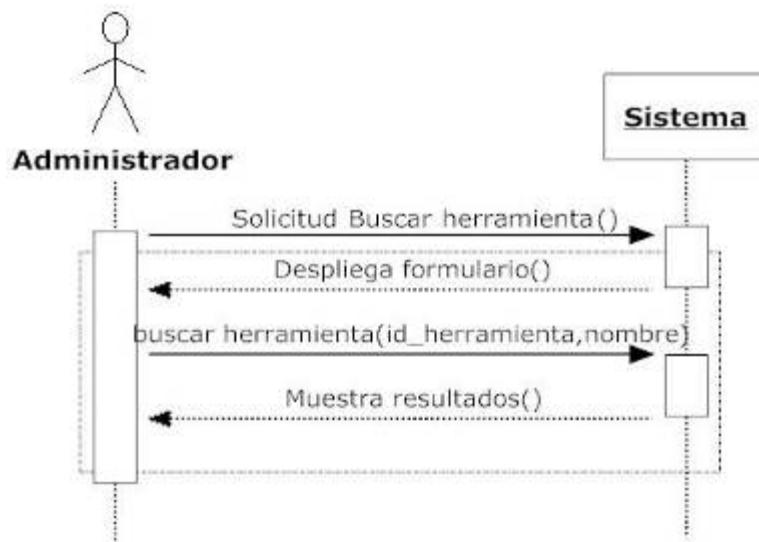
*Todos los datos: id insumo, nombre, descripción.

Flujo Alternativo:

Post-condiciones: No hay.

6.8.7.2.1.- Diagrama De Secuencia Caso De Uso Buscar Herramientas

Imagen 34: Diagrama De Secuencia Caso De Uso Buscar Herramienta



6.8.7.3.- Caso De Uso Eliminar Herramientas

Nombre:	Eliminar Herramientas
Descripción:	Permite eliminar una herramienta en la base.
Actores:	Administrador
Pre-condiciones:	Administrador logeado y confirmado
Flujo Principal: <ol style="list-style-type: none">1. Administrador hace clic en el menú, opción “Eliminar Herramientas”.2. Sistema despliega el modulo de búsqueda, un formulario para introducir datos respectivos (nombre herramientas) de la herramienta a buscar.3. Sistema realiza búsqueda contra la base y despliega los resultados en forma de malla de datos, donde se ve un conjunto de filas y columnas que contienen todos los datos del o las herramientas que coincidieron con el término buscado.4. El administrador selecciona la o las herramienta(as) que desea borrar y presiona el eliminar5. El sistema actualiza la base de datos <p>*Todos los datos: id herramienta, nombre, descripción.</p>	

Flujo Alternativo:

3.a Sistema no encuentra la herramienta deseado

1 vuelve al paso 2

4.a El administrador presiona eliminar sin seleccionar herramienta

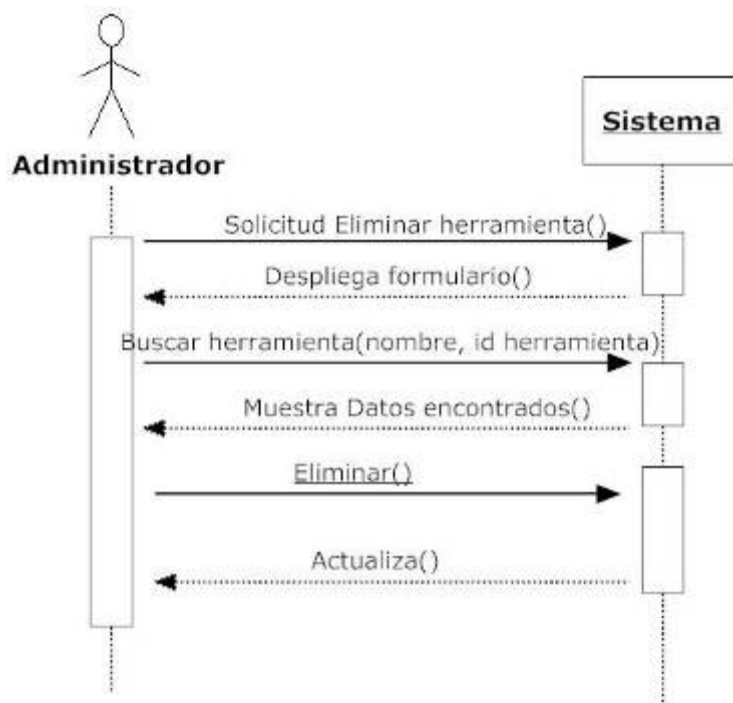
1 El sistema informa de la no selección y vuelve al paso 2

6.a Administrador no confirma y cancela la operación

Post-condiciones:	Base de datos actualizada
--------------------------	---------------------------

6.8.7.3.1.- Diagrama De Secuencia Caso De Uso Eliminar Herramientas

Imagen 35: Diagrama De Secuencia Caso De Uso Eliminar Herramientas



6.8.7.4.- Caso De Uso Modificar Herramientas

Nombre:	Modificar Herramientas
Descripción:	Permite realizar el proceso a través del cual se modifica los datos de una herramienta.
Actores:	Administrador
Pre-condiciones:	Administrador logeado y confirmado
Flujo Principal: <ol style="list-style-type: none">1. Administrador hace clic en el menú, opción “Modificar Herramienta”2. El administrador busca la Herramienta a la cual desea modificar datos(ver caso de uso buscar Herramientas)3. El sistema presenta el formulario de modificación de Herramientas.4. El Administrador modifica los campos necesarios.5. El sistema solicita confirmación.6. El Administrador confirma7. El sistema actualiza la base.	

Flujo Alternativo:

2.a Herramienta no encontrado

1 Terminó de la operación

4.a Administrador deja campos obligatorios incompletos

1. Sistema da aviso de la falta de datos obligatorios y espera por ellos

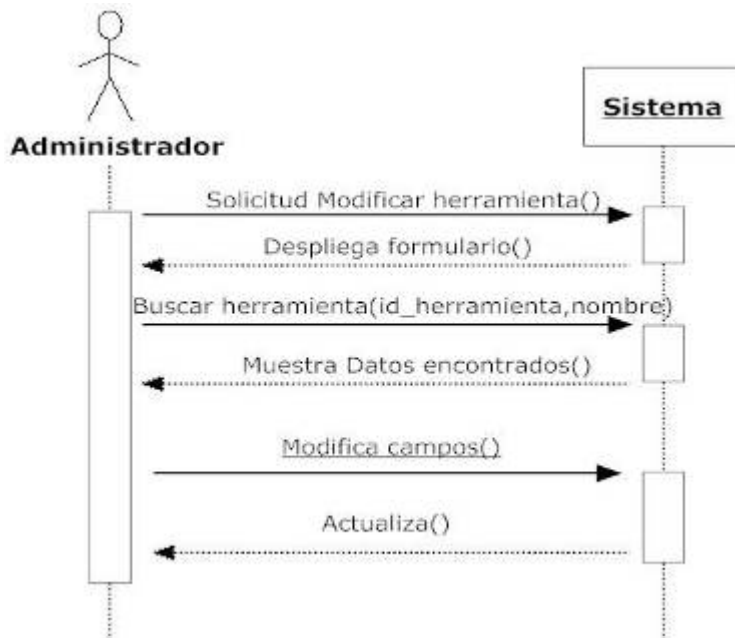
6.a El Administrador cancela.

1 El sistema no actualiza

Post-condiciones:	Base actualizada.
--------------------------	-------------------

6.8.7.4.1.- Diagrama De Secuencia Modificar Herramienta

Imagen 36: Diagrama De Secuencia Caso De Uso Modificar Herramienta



6.8.7.5.- Caso De Uso Ingresar Herramientas

Nombre:	Ingresar Herramientas
Descripción:	Permite realizar el proceso a través del cual se Ingresan los datos de una Herramienta.
Actores:	Administrador
Pre-condiciones:	Administrador logeado y confirmado
Flujo Principal: <ol style="list-style-type: none">1. Administrador hace clic en el menú, opción “Ingresar Herramientas”2. El sistema presenta el formulario de ingreso herramientas con los datos de la herramienta a ingresar3. El Administrador ingresa los nuevos datos (nombre, descripción ,id_herramienta).4. El sistema solicita confirmación.5. El Administrador confirma6. El sistema actualiza la base.	

Flujo Alternativo:

3.a el administrador deja datos obligatorios incompletos

1 El sistema avisa de la falta de datos obligatorios y espera por ellos

3.b id_herramienta esta ocupado

1 El sistema avisa del error, no ingresa y espero por el cambio del dato a uno correcto

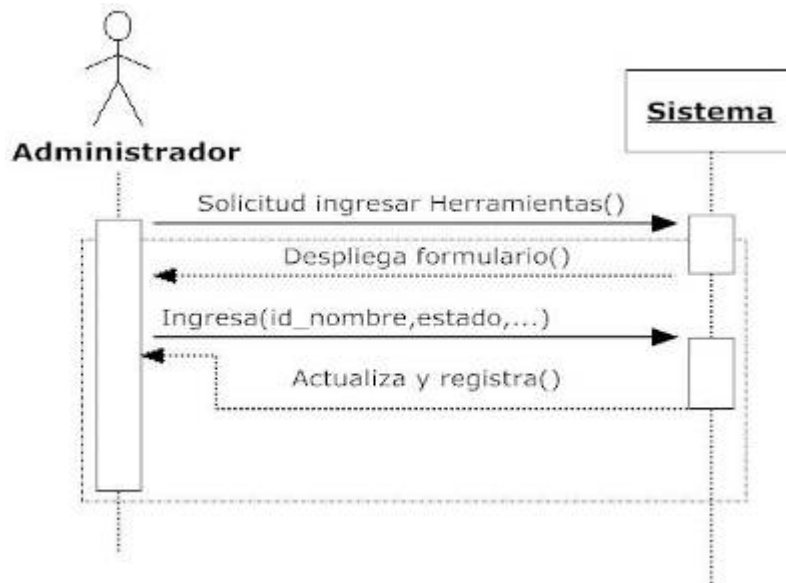
6. a El Administrador cancela.

1 El sistema no actualiza

Post-condiciones:	Base actualizada.
--------------------------	-------------------

6.8.7.5.1.- Diagrama De Secuencia Caso De Uso Ingresar Herramienta

Imagen 37: Diagrama De Secuencia Caso De Uso Ingresar Herramienta



6.8.7.6.- Caso De Uso Entregar Herramientas

Nombre:	Entregar Herramientas
Descripción:	Permite realizar el proceso a través del cual se entrega una Herramientas a un empleado de la empresa.
Actores:	Bodeguero
Pre-condiciones:	Bodeguero logeado y confirmado
Flujo Principal:	
<ol style="list-style-type: none"> 1. Bodeguero hace clic en el menú, opción “Entregar Herramientas” 2. El bodeguero busca la herramienta (ver caso de uso buscar herramienta) y el empleado (ver caso de uso buscar empleado) a la cual se le asignara la herramienta. 3. El sistema presenta el formulario de Entregar herramientas con los datos de la herramienta encontrada y los del empleado 4. El bodeguero selecciona los datos de la entrega y hace clic en aceptar. 5. El sistema solicita confirmación 6. Administrador confirma 7. El sistema actualiza la base. 	

Flujo Alternativo:

2.a Herramienta no encontrado o en estado de prestada

1 Se termina la operación

2.b Empleado no encontrado

1 Se termina lo operación

4.a Bodeguero realiza selección incompleta de los datos de entrega

1. sismte avisa y espera por la selección de datos faltantes

6. a El bodeguero cancela.

1 El sistema no actualiza

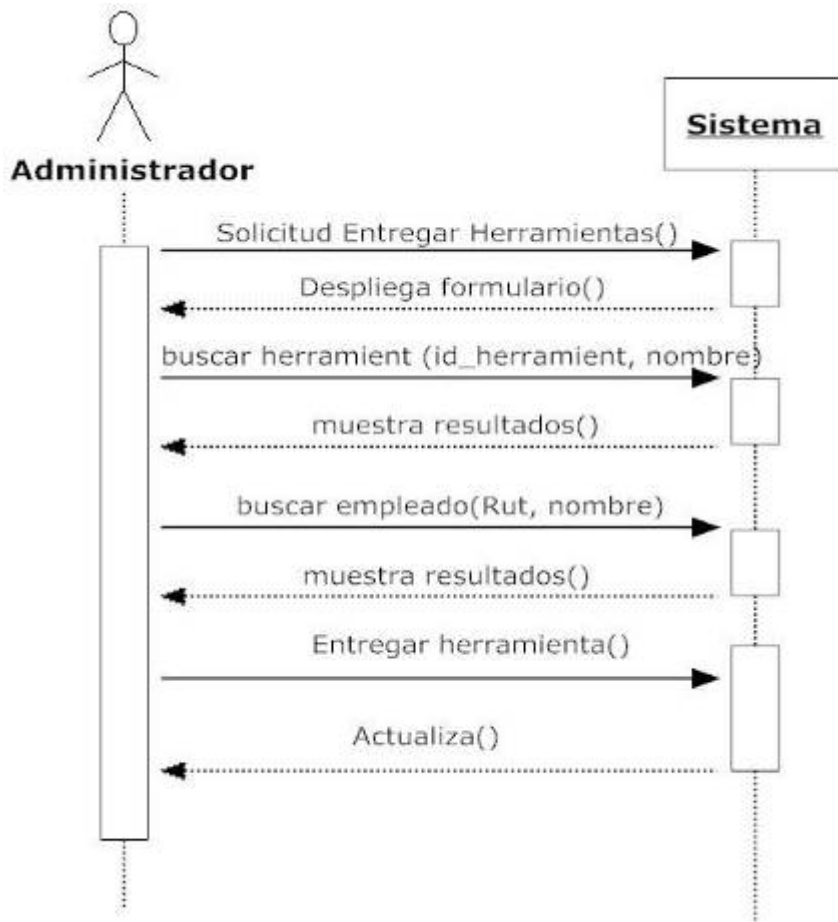
Post-condiciones:

Base actualizada con el préstamo de herramienta correspondiente.

Herramienta actualizada en estado de prestada.

6.8.7.6.1.- Diagrama De Secuencia Entregar Herramientas

Imagen 38: Diagrama De Secuencia Caso De Uso Entregar Herramienta



6.8.7.7.- Caso De Uso Devolver Herramientas

Nombre:	Devolver Herramientas
Descripción:	Permite realizar el proceso a través del cual se devuelve una Herramienta prestada al empleado para realizar un trabajo específico.
Actores:	Bodeguero
Pre-condiciones:	Bodeguero logeado y confirmado

Flujo Principal:

1. Bodeguero hace clic en el menú, opción “Devolver Herramientas”
2. El bodeguero busca por herramienta (ver caso de uso buscar herramienta) o por empleado (ver caso de uso buscar empleado) los préstamos correspondientes.
3. El sistema presenta el formulario de devolver herramientas con los datos de los préstamos encontrados con las características antes mencionadas.
4. El bodeguero selecciona los datos del préstamo para su devolución, selecciona fecha de devolucion y hace clic en aceptar.
5. El sistema solicita confirmación
6. Administrador confirma
7. El sistema actualiza la base.

Flujo Alternativo:	
2.a Prestamos no encontrado por herramienta	
1 Se termina la operación	
2.b Prestamos no encontrado por empleado	
1 Se termina lo operación	
4.a Bodeguero no selecciona prestamos y presiona boton aceptar.	
1.sistema avisa y espera por la selección de datos faltantes	
4.b Bodeguero selecciona una fecha errónea	
1.sistema avisa que la fecha es previa al préstamo y espera por una nueva fecha correcta.	
6. a El bodeguero cancela.	
1 El sistema no actualiza	
Post-condiciones:	Base actualizada con el préstamo de herramienta correspondiente. Herramienta actualizada en estado de prestada.

6.8.7.8.- Caso De Uso Ingresar Insumos

Nombre:	Ingresar Insumo
Descripción:	Permite realizar el proceso a través del cual se Ingresa los datos de un nuevo Insumo.
Actores:	Administrador

Pre-condiciones:	Administrador logeado y confirmado
Flujo Principal:	
<ol style="list-style-type: none"> 1. Administrador hace clic en el menú, opción “Ingresar Insumo” 2. El sistema presenta el formulario de ingreso insumos con los datos del insumo a ingresar 3. El Administrador ingresa los nuevos datos (nombre, descripción, tipo). 4. El sistema solicita confirmación. 5. El Administrador confirma 6. El sistema actualiza la base. 	
Flujo Alternativo:	
<ol style="list-style-type: none"> 3.a el administrador deja datos obligatorios incompletos 1 El sistema avisa de la falta de datos obligatorios y espera por ellos 6. a El Administrador cancela. 1 El sistema no actualiza 	
Post-condiciones:	Base actualizada.

6.8.7.9.- Caso De Uso Eliminar Insumo (Descripción)

Nombre:	Eliminar Insumo
----------------	------------------------

Descripción:	Permite realizar el proceso a través del cual se eliminaran los datos de los insumos de la empresa.
Actores:	Administrador
Pre-condiciones:	Administrador logeado y confirmado
Post-condiciones:	Base de datos actualizada

6.8.7.10.- Caso De Uso Modificar Insumo (Descripción)

Nombre:	Modificar Insumo
Descripción:	Permite realizar el proceso a través del cual se modifica los datos de un Insumo del vivero.
Actores:	Administrador
Pre-condiciones:	Administrador logeado y confirmado
Post-condiciones:	Base de datos actualizada con las modificaciones correspondientes a los trabajos

6.8.7.11.- Caso De Uso Comprar Insumos

Nombre:	Comprar Insumo
----------------	-----------------------

Descripción:	Permite realizar el proceso a través del cual se Ingresan los datos de una nueva compra de Insumos para los trabajos realizados en el vivero.
Actores:	Administrador
Pre-condiciones:	Administrador logeado y confirmado
Flujo Principal: <ol style="list-style-type: none">1. Administrador hace clic en el menú, opción “Comprar Insumo”2. El sistema presenta el formulario de comprar insumos con los datos correspondientes a la nueva compra.3. El administrador busca los datos del proveedor y del insumo que se desea comprar.4. El sistema despliega los datos encontrados.5. El administrador asigna el precio y la cantidad del insumo a comprar y presiona agregar.6. Administrador Presiona aceptar nueva compra7. El sistema actualiza la base. <p>Administrador repite los pasos 2 al 5 hasta que estime conveniente.</p>	

Flujo Alternativo:	
3.a Datos del proveedor no encontrados	
1. sistema da aviso del problema y entrega la opción de cancelar la operación o ingresar proveedor(ver caso e uso ingresar proveedor)	
3.a Datos del insumo no encontrados	
1. sistema da aviso del problema y entrega la opción de cancelar la operación o ingresar insumo(ver caso e uso ingresar insumo)	
5.a Administrador presiona agregar sin asignar precio o cantidad	
1 El sistema avisa de la falta de datos obligatorios y espera por ellos	
6. a El Administrador cancela.	
1 El sistema no actualiza	
Post-condiciones:	Base actualizada.

6.8.7.12.- Caso De Uso Eliminar Compra Insumo (Descripción)

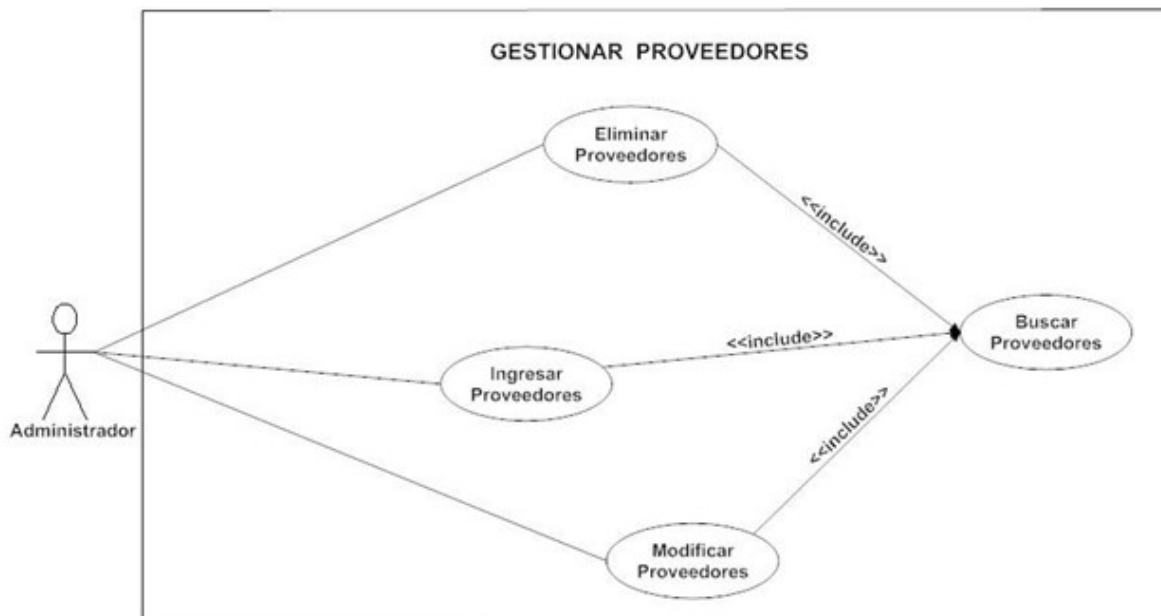
Nombre:	Eliminar Compra Insumo
Descripción:	Permite realizar el proceso a través del cual se eliminarán los de las compras realizadas por el vivero a los distintos proveedores
Actores:	Administrador
Pre-condiciones:	Administrador logeado y confirmado
Post-condiciones:	Base de datos actualizada

6.8.7.13.- Caso De Uso Modificar Compra Insumo (Descripción)

Nombre:	Modificar Compra Insumo
Descripción:	Permite realizar el proceso a través del cual se modifica los datos de una compra de insumo a alguno de los proveedores del vivero
Actores:	Administrador
Pre-condiciones:	Administrador logeado y confirmado
Post-condiciones:	Base de datos actualizada con las modificaciones correspondientes a los trabajos

6.8.8.- Caso De Uso Proveedores

Imagen 39: Caso De Uso Gestión Proveedores



6.8.8.1.- Caso De Uso Eliminar Proveedores (Descripción)

Nombre:	Eliminar Proveedor
Descripción:	Permite realizar el proceso a través del cual se eliminaran los datos de un proveedor.
Actores:	Administrador
Pre-condiciones:	Administrador logeado y confirmado
Post-condiciones:	Base de datos actualizada

6.8.8.2.- Caso De Uso Modificar Proveedor (Descripción)

Nombre:	Modificar Proveedor
Descripción:	Permite realizar el proceso a través del cual se modifica los datos de un proveedor de la empresa.
Actores:	Administrador
Pre-condiciones:	Administrador logeado y confirmado
Post-condiciones:	Base de datos actualizada

6.8.8.3.- Caso De Uso Ingresar Proveedor (Descripción)

Nombre:	Ingresar Proveedor
----------------	---------------------------

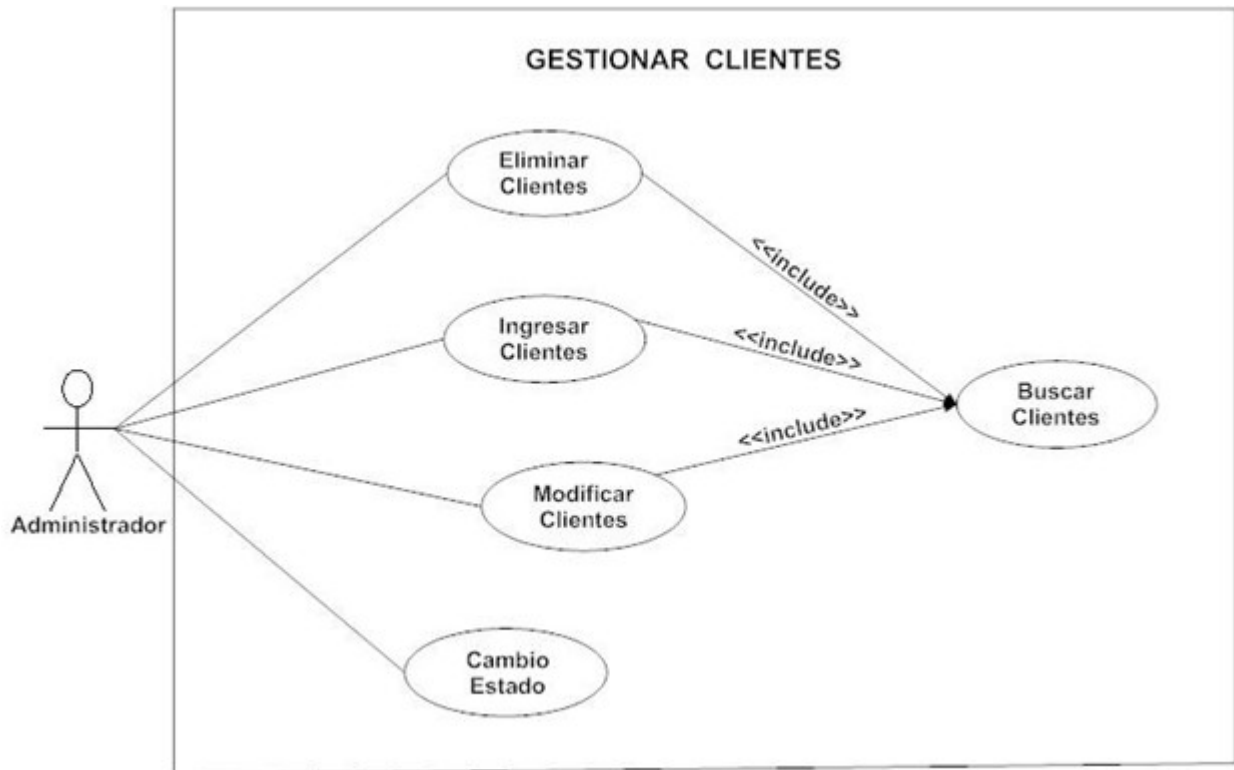
Descripción:	Permite realizar el proceso a través del cual se ingresaran los nuevos datos de un proveedor de la empresa.
Actores:	Administrador
Pre-condiciones:	Administrador logeado y confirmado
Post-condiciones:	Base de datos actualizada

6.8.8.4.- Caso De Uso Buscar Proveedor (Descripción)

Nombre:	Buscar Proveedor
Descripción:	Permite realizar el proceso a través del cual se buscaran los datos de un proveedor de la empresa.
Actores:	Administrador
Pre-condiciones:	Administrador logeado y confirmado
Post-condiciones:	

6.8.9.- Caso De Uso Gestión Clientes

Imagen 40: Caso De Uso Gestión Clientes



6.8.9.1.- Caso De Uso Cambiar Estado Cliente (Descripción)

Nombre:	Cambio Estado Cliente
Descripción:	Permite realizar el proceso a través del cual se cambiara el estado basado en la reglas del negocio del vivero de un cliente de normal a vip o viceversa.
Actores:	Administrador
Pre-condiciones:	Administrador logeado y confirmado
Post-condiciones:	

6.8.9.2.- Caso De Uso Eliminar Clientes (Descripción)

Nombre:	Eliminar Clientes
Descripción:	Permite realizar el proceso a través del cual se eliminarán los datos de un Cliente del vivero.
Actores:	Administrador
Pre-condiciones:	Administrador logeado y confirmado
Post-condiciones:	Base de datos actualizada

6.8.9.3.- Caso De Uso Ingresar Cliente (Descripción)

Nombre:	Ingresar Insumos
Descripción:	Permite realizar el proceso a través del cual se ingresarán los nuevos datos de un Cliente del vivero.
Actores:	Administrador
Pre-condiciones:	Administrador logeado y confirmado
Post-condiciones:	Base de datos actualizada

6.8.9.4.- Caso De Uso Modificar Cliente (Descripción)

Nombre:	Modificar Insumos
Descripción:	Permite realizar el proceso a través del cual se modifica los datos de un Cliente del vivero

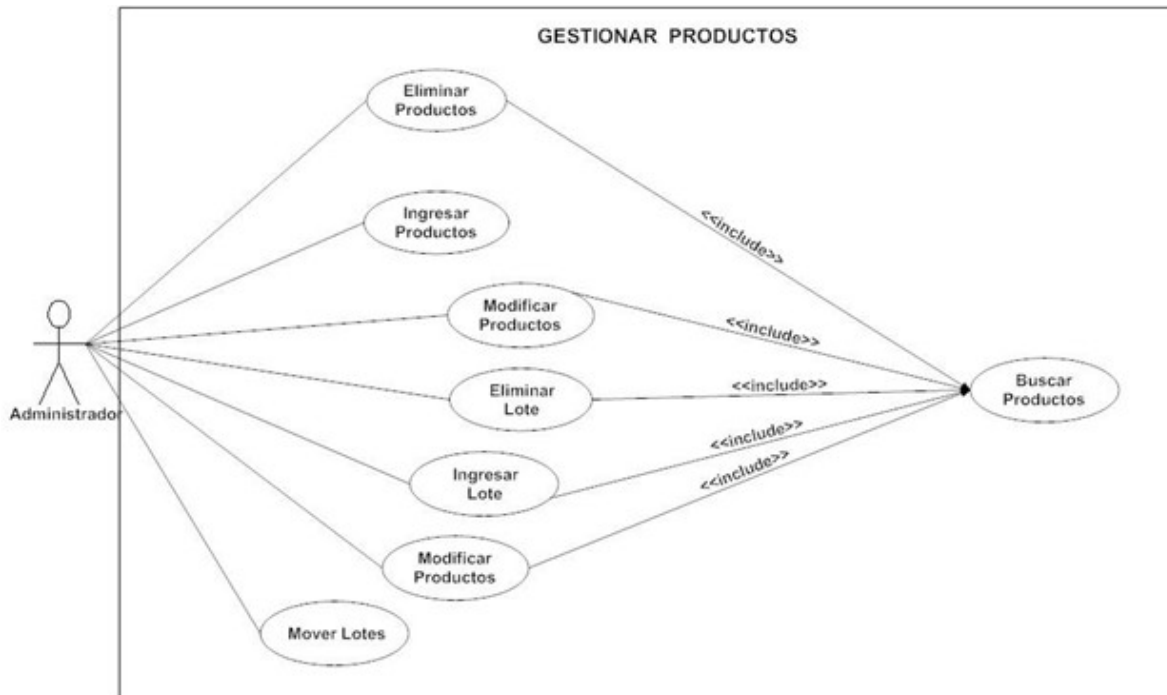
Actores:	Administrador
Pre-condiciones:	Administrador logeado y confirmado
Post-condiciones:	Base de datos actualizada

6.8.9.5.- Caso De Uso Buscar Cliente (Descripción)

Nombre:	Buscar Insumo
Descripción:	Permite realizar el proceso a través del cual se buscaran y mostraran los datos de un cliente del vivero.
Actores:	Administrador
Pre-condiciones:	Administrador logeado y confirmado
Post-condiciones:	

6.8.10.- Caso De Uso Gestión Productos

Imagen 41: Caso De Uso Gestión Productos



6.8.10.1.- Caso De Uso Eliminar Productos (Descripción)

Nombre:	Eliminar Productos
Descripción:	Permite realizar el proceso a través del cual se eliminarán los datos de un productos del vivero
Actores:	Administrador
Pre-condiciones:	Administrador logeado y confirmado
Post-condiciones:	Base de datos actualizada

6.8.10.2.- Caso De Uso Ingresar Productos (Descripción)

Nombre:	Ingresar Productos
Descripción:	Permite realizar el proceso a través del cual se ingresaran los nuevos datos de los Productos del vivero
Actores:	Administrador
Pre-condiciones:	Administrador logeado y confirmado
Post-condiciones:	Base de datos actualizada

6.8.10.3.- Caso De Uso Modificar Productos (Descripción)

Nombre:	Modificar Productos
Descripción:	Permite realizar el proceso a través del cual se modifica los datos de un producto del vivero
Actores:	Administrador
Pre-condiciones:	Administrador logeado y confirmado
Post-condiciones:	Base de datos actualizada

6.8.10.4.- Caso De Uso Buscar Productos (Descripción)

Nombre:	Buscar Productos
Descripción:	Permite realizar el proceso a través del cual se buscaran y mostraran los datos de un producto del vivero.

Actores:	Administrador
Pre-condiciones:	Administrador logeado y confirmado
Post-condiciones:	

6.8.10.5.- Caso De Uso Eliminar Lote (Descripción)

Nombre:	Eliminar Lote
Descripción:	Permite realizar el proceso a través del cual se eliminara un lote del vivero
Actores:	Administrador
Pre-condiciones:	Administrador logeado y confirmado
Post-condiciones:	Base de datos actualizada

6.8.10.6.- Caso De Uso Ingresar Lote (Descripción)

Nombre:	Ingresar Lote
Descripción:	Permite realizar el proceso a través del cual se ingresaran los nuevos datos de un lote del vivero
Actores:	Administrador
Pre-condiciones:	Administrador logeado y confirmado

Post-condiciones:	Base de datos actualizada
--------------------------	---------------------------

6.8.10.7.- Caso De Uso Modificar Lote (Descripción)

Nombre:	Modificar Insumos
Descripción:	Permite realizar el proceso a través del cual se modifica los datos de un lote del vivero
Actores:	Administrador
Pre-condiciones:	Administrador logeado y confirmado
Post-condiciones:	Base de datos actualizada

6.8.10.8.- Caso De Uso Mover Lotes

Nombre:	Comprar Insumo
Descripción:	Permite realizar el proceso a través del cual se moverán los productos de los lotes a las distintas categorías definidas por reglas del negocio(merma, otros, planta madre) desde el stock o viceversa
Actores:	Administrador
Pre-condiciones:	Administrador logeado y confirmado

Flujo Principal:

1. Administrador hace clic en el menú, opción “Mover Lote”.
2. El sistema presenta el formulario de mover lotes con los datos correspondientes a los lotes y el producto que lo conforma, cantidad de stock actual, cantidad de mermas, cantidad de planta madre, cantidad de otros
3. El administrador busca por producto el lote que desea y selecciona el tipo de categoría a mover.
4. El sistema despliega los datos encontrados en la categoría deseada.
5. El administrador acepta e ingresa la cantidad a sumar, la fecha, descripción, la categoría a la cual desea mover esa cantidad y acepta
6. Sistema actualiza la base de datos y asigna las nuevas cantidades y

Flujo Alternativo:

2. a Sistema no tiene datos para mostrar en lotes.

1. se termina la operación.

3.a Datos lote con el producto buscado no encontrado

1. se termina la operación o vuelve a realizar búsqueda con otro producto.

3.b Datos categoría seleccionada no tiene elementos

1. sistema despliega grilla vacía, espera por nueva búsqueda y selección.

5.a Administrador presiona aceptar sin asignar datos

1 El sistema avisa de la falta de datos obligatorios y espera por ellos.

5. b El Administrador cancela.

1 El sistema no actualiza y termina la operación.

Post-condiciones:

Base actualizada.

6.9.- Diagrama De Clases

A continuación se puede ver el diagrama de clases de la segunda iteración de la segunda fase RUP, el cual aun puede sufrir alguna modificaciones pero mínimas en las siguientes etapas de construcción creando o agregando nuevas entidades para satisfacer las necesidades y requerimientos del proyecto, pero si es la base para empezar la tercera fase, la de construcción del sistema

Imagen 42: Diagrama De Clases

6.10.- Modelo De Prueba Del Software

Existen varios flujos de trabajo para realizar las pruebas:

Modelo de Prueba: Éste describe cómo se prueban los distintos componentes del sistema.

Caso de Prueba: Especifica la forma en que se probará el sistema, indica cómo debe ser la entrada y los resultados con los que se va a realizar la prueba. Entre los casos de prueba más común podemos mencionar:

Aquel que establece cómo probar los casos de uso. En estos casos se especifica una prueba del sistema utilizando Caja Negra

Aquel que especifica cómo probar la realización de un caso de uso específico. En estos casos, se especifica una prueba del sistema utilizando Caja Blanca

Procedimiento de Prueba: Indica cómo hacer la realización de uno o varios casos de prueba.

Componente de Prueba: Permite la automatización de uno o varios procedimientos de prueba. Esto se puede realizar utilizando un lenguaje de programación

Plan de Prueba: Permite describir estrategias, recursos y planificación de la prueba. Se definen especificaciones del tipo de prueba que se va a utilizar y qué objetivos debe lograr.

Evaluación de Prueba: Se evaluará los resultados que arrojaron las pruebas realizadas al sistema

Planificación de Prueba: Se determinará qué pruebas se van a realizar, cuándo y a qué partes del sistema.

6.10.1.- Casos de prueba

La importancia de la prueba del software y sus implicaciones en la calidad del software no se pueden pasar por alto. La prueba del software es un elemento crítico para la garantía de calidad del software y representa una revisión final de las especificaciones, del diseño y de la codificación.

El objetivo es diseñar pruebas que sistemáticamente saquen a la luz diferentes clases de errores, haciéndolo con la menor cantidad de tiempo y esfuerzo. La prueba es uno de los pasos de la ingeniería del software que se puede ver como destructivo en lugar de constructivo, ya que aquí el Ingeniero en software intenta demoler lo logrado en las etapas de análisis y diseño, con el fin de encontrar algún tipo de error.

A continuación se presentarán los fundamentos de la prueba del software y las técnicas de diseño de casos de prueba. Los fundamentos de la prueba del software definen los objetivos esenciales de la prueba del software. El diseño de casos de prueba se centra en un conjunto de técnicas para la creación de casos de prueba que satisfagan los objetivos globales de la prueba

6.10.2.- Prueba de La Caja Negra

Los métodos de prueba de la caja negra se centran en los requisitos funcionales del software. O sea, la prueba de la caja negra permite al ingeniero del software obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. La prueba de la caja negra *no* es una alternativa a las técnicas de prueba de la caja blanca. Más bien se trata de un enfoque complementario que intenta descubrir diferentes tipos de errores que los métodos de la caja blanca.

La prueba de la caja negra intenta encontrar errores de las siguientes categorías:

Funciones incorrecta o ausente.

Errores de interfaz.

Errores en estructuras de datos o en accesos a bases de datos externas.

Errores de rendimiento.

Errores de iniciación y de terminación.

A diferencia de la prueba de la caja blanca, que se lleva a cabo previamente en el proceso de prueba, la prueba de la caja negra tiende a aplicarse durante posteriores fases de prueba. Ya que la prueba de la caja negra ignora intencionadamente la estructura de control, centra

su atención en el campo de información. Las pruebas se diseñan para responder a las siguientes preguntas:

¿Cómo se prueba la validez funcional?

¿*Qué clases* de entrada compondrán unos buenos casos de prueba?

¿Es el sistema particularmente sensible a ciertos valores de entrada?

¿De qué forma están aislados los límites de una clase de datos?

¿Qué volúmenes y miles de datos tolerará el sistema?

¿Qué efectos sobre la operación del sistema tendrán combinaciones específicas de datos?

Mediante las técnicas de prueba de la caja negra se deriva un conjunto de casos de prueba que satisfacen los siguientes criterios:

Casos de prueba que reducen, en un coeficiente que es mayor que uno, el número de casos de prueba adicionales que se deben diseñar para alcanzar una prueba razonable.

Casos de prueba que nos dicen algo sobre la presencia o ausencia de clases de errores en lugar de un error asociado solamente con la prueba, en particular, que se encuentre disponible.

6.10.2.1.- Partición equivalente

La partición equivalente es un método de prueba de la caja negra que divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. Un caso de prueba ideal descubre de forma inmediata una clase de errores (p. ej.: procesamiento incorrecto de todos los datos de caracteres) que, de otro modo, requerirían la ejecución de muchos casos antes de detectar el error genérico. La partición equivalente se dirige a la definición de casos de prueba que descubran clases de errores, reduciendo así el número total de casos de prueba que hay que desarrollar.

El diseño de casos de prueba para la partición equivalente se basa en una evaluación de las clases de equivalencia para una *condición de entrada*. *Una clase de equivalencia*

representa un conjunto de estados válidos o inválidos para condiciones de entrada. Típicamente, una condición de entrada es un valor numérico específico, un rango de valores, un conjunto de valores relacionados o una condición lógica. Las clases de equivalencia se pueden definir de acuerdo con las siguientes directrices:

Si una condición de entrada especifica un *rango*, se define una clase de equivalencia válida y dos inválidas.

Si una condición de entrada requiere un *valor* específico, se define una clase de equivalencia válida y dos inválidas.

Si una condición de entrada especifica un miembro de un *conjunto*, se define una clase de equivalencia válida y una inválida.

Si una condición de entrada es *lógica*, se define una clase válida y una inválida.

6.10.2.2.- Análisis de valores límite

Por razones que no están del todo claras, los errores tienden a darse más en los límites del campo de entrada que en el "centro". Por ello, se ha desarrollado el *análisis de valores límites* (AVL) como técnica de prueba. El análisis de valores límite nos lleva a una elección de casos de prueba que ejerciten los valores límite.

El análisis de valores límite es una técnica de diseño de casos de prueba que complementa a la partición equivalente. En lugar de seleccionar cualquier elemento de una clase de equivalencia, el AVL lleva a la elección de casos de prueba en los "bordes" de la clase. En lugar de centrarse solamente en las condiciones de entrada, el AVL deriva casos de prueba también para el campo de salida.

Las directrices de AVL son similares en muchos aspectos a las que proporciona la partición equivalente:

Si una condición de entrada especifica un *rango* delimitado por los valores *a* y *b*, se deben diseñar casos de prueba para los valores *a* y *b* y para los valores justo por debajo y justo por encima de *a* y *b*, respectivamente.

Si una condición de entrada especifica un número de valores, se deben desarrollar casos de prueba que ejerciten los valores máximo y mínimo. También se deben probar los valores justo por encima y justo por debajo del máximo y del mínimo.

Aplicar las directrices 1 y 2 a las condiciones de salida. Por ejemplo, supongamos que se requiere una tabla de "temperatura frente a presión" como salida de un programa de análisis de ingeniería. Se deben diseñar casos de prueba que creen un informe de salida que produzca el máximo (y el mínimo) número permitido de entradas en la tabla.

6.10.3.- Prueba de La Caja Blanca

La prueba de la caja blanca es un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para derivar los casos de prueba. Mediante los métodos de prueba de la caja blanca, el ingeniero del software puede obtener casos de prueba que:

1. Garanticen que se ejercitan por lo menos una vez todos los caminos independientes de cada módulo
2. Ejerciten todas las decisiones lógicas en sus vertientes verdadera y falsa
3. Ejecuten todos los bucles en sus límites y con sus límites operacionales
4. Ejerciten las estructuras internas de datos para asegurar su validez.

Las pruebas de caja blanca, pueden llegar a corregir los siguientes tipos de errores:

- Los errores lógicos y las suposiciones incorrectas son inversamente proporcionales a la probabilidad de que se ejecute un camino del programa.
- A menudo creemos que un camino lógico tiene pocas posibilidades de ejecutarse cuando, de hecho, se puede ejecutar de forma regular.
- Los errores tipográficos son aleatorios.

6.10.3.1.- Prueba del Camino Básico

La prueba del camino básico es una técnica de prueba de la caja blanca. El método del camino básico permite al diseñador de casos de prueba obtener una medida de la complejidad lógica de un diseño procedimental y usar esa medida como guía para la definición de un conjunto básico de caminos de ejecución. Los casos de prueba derivados del conjunto básico garantizan que durante la prueba se ejecuta por lo menos una vez cada sentencia del programa.

Derivación de casos de prueba

El método de prueba del camino básico se puede aplicar a un diseño procedimental detallado o a un código fuente.

Se presentará la prueba del camino básico como una serie de pasos:

Usando el diseño o el código como base se dibuja el correspondiente grafo de flujo.

Se determina la complejidad ciclomática del grafo de flujo resultante.

Se determina un conjunto básico de caminos linealmente independientes.

Finalmente se preparan los casos de prueba que forzarán la ejecución de cada camino del conjunto básico.

Se ejecuta cada caso de prueba y se comparan los resultados con los esperados. Una vez terminados todos los casos de prueba, el probador podrá estar seguro de que todas las sentencias del programa se han ejecutado por lo menos una vez.

Matrices de grafos

El procedimiento de obtención del grafo de flujo e incluso la determinación de un conjunto básico de caminos es susceptible de ser mecanizado. Para desarrollar una herramienta de software que ayude en la prueba del camino básico, puede ser bastante útil una estructura de datos denominada matriz de grafo.

Una matriz de grafo es una matriz cuadrada cuyo tamaño (o sea, el número de filas y de columnas) es igual al número de nodos del grafo de flujo. Cada fila y cada columna corresponde a un nodo específico y las entradas de la matriz corresponden a las *conexiones* (aristas) entre los nodos. La matriz de grafo no es nada más que una representación tabular del grafo de flujo. Sin embargo, añadiendo un peso de enlace a cada entrada de la matriz, la matriz de grafo se puede convertir en una potente herramienta para la evaluación de la estructura de control del programa durante la prueba. El peso de enlace nos da información adicional sobre el flujo de control. En su forma más sencilla, el peso de enlace es 1 (existe una conexión) ó 0 (no existe conexión).

Pero a los pesos de enlace se les pueden asignar otras propiedades más interesantes:

La probabilidad de que un enlace (arista) sea ejecutado

El tiempo de procedimiento asociado al recorrido de un enlace

La memoria requerida durante el recorrido de un enlace

Los recursos requeridos durante el recorrido de un enlace

6.10.3.2.- Prueba de la Estructura de Control

La técnica de prueba del camino base descrita anteriormente es una de muchas técnicas para la *prueba de la estructura de control*. Aunque la prueba del camino base es sencilla y altamente efectiva, no es suficiente por sí sola. Aquí se tratarán otras variantes de la prueba de estructura de control. Estas variantes amplían la cobertura de la prueba y mejoran la calidad de la prueba de la caja blanca.

6.10.3.3.- Prueba de condiciones

La prueba de condiciones es un método de diseño de casos de prueba que ejercita las condiciones lógicas contenidas en el módulo de un programa.

El método de la *prueba de condiciones* se centra en la prueba de cada una de las condiciones del programa. Las estrategias de prueba de condiciones tienen, generalmente, dos ventajas. La primera, es que la medición de la cobertura de la prueba de una condición

es sencilla. La segunda, es que la cobertura de la prueba de las condiciones de un programa da una orientación para la generación de pruebas adicionales del programa.

El propósito de la prueba de condiciones es detectar, no sólo los errores en las condiciones de un programa, sino también otros errores en dicho programa. Si un conjunto de pruebas de un programa P es efectivo para detectar errores en las condiciones que se encuentran en P , es probable que el conjunto de pruebas también sea efectivo para detectar otros errores en el programa P . Además, si una estrategia de prueba es efectiva para detectar errores en una condición, entonces es probable que dicha estrategia también sea efectiva para detectar errores en el programa.

Se ha propuesto una serie de estrategias de prueba de condiciones:

La prueba de ramificaciones es posiblemente la estrategia de prueba de condiciones más sencilla. Para una condición compuesta C , es necesario ejecutar al menos una vez las ramas verdadera y falsa de C y cada condición simple de C .

La prueba del campo requiere la realización de tres o cuatro pruebas para una expresión relacionar.

También se pueden derivar pruebas sensibles a error para expresiones lógicas. Para una expresión lógica singular (una expresión lógica en la cual cada variable lógica sólo aparece una vez) con n variables lógicas ($n > 0$), podemos generar fácilmente un conjunto de pruebas con menos de 2^n pruebas, de tal forma que ese conjunto de pruebas garantice la detección de múltiples errores de operadores lógicos y también sea efectivo para detectar otros errores.

Tai sugiere una estrategia de prueba de condiciones que se basa en las técnicas destacadas anteriormente. La técnica, denominada *prueba del operador relacional y de ramificación* (del inglés, BRO), garantiza la detección de errores de operadores relacionales o de ramificaciones en una condición dada, en la que todas las variables lógicas y operadores relacionales de la condición aparecen sólo una vez y no tienen variables en común. La estrategia BRO utiliza *restricciones de condición* para una condición C .

6.10.3.4.- Prueba de flujo de datos

El método de *prueba de flujo de datos* selecciona caminos de prueba de un programa de acuerdo con la ubicación de las definiciones y los usos de las variables del programa. Se han estudiado y comparado varias estrategias de prueba de flujo de datos.

Las estrategias de prueba de flujo de datos son útiles para seleccionar caminos de prueba de un programa que contengan sentencias condicionales o de bucles anidadas.

Dado que las sentencias de un programa están relacionadas entre sí de acuerdo con las definiciones de las variables, el enfoque de prueba de flujo de datos es efectivo para la protección contra errores. Sin embargo, los problemas de medida del cubrimiento de la prueba y de selección de caminos de prueba para la prueba de flujo de datos son más difíciles que los correspondientes problemas para la prueba de condiciones.

6.10.3.5.- Prueba de bucles

La prueba de bucles es una técnica de prueba de la caja blanca que se centra exclusivamente en la validez de las construcciones de bucles. Se pueden definir cuatro clases diferentes de bucles: *bucles simples*, *bucles concatenados*, *bucles anidados* y *bucles no estructurados*.

Bucles simples. A los bucles simples se les debe aplicar el siguiente conjunto de pruebas, donde **n** es el número máximo de pasos permitidos por el bucle:

Pasar por alto totalmente el bucle

Pasar una sola vez por el bucle

Pasar dos veces por el bucle

Hacer m pasos por el bucle con $M < n$

Hacer $n - 1$, n y $n + 1$ pasos por el bucle

Bucles anidados. Si extendemos el enfoque de prueba de los bucles simples a los bucles anidados, el número de posibles pruebas aumenta geométricamente a medida que aumenta

el nivel de anidamiento. Esto llevaría a un número impracticable de pruebas. Beizer sugiere un enfoque que ayuda a reducir el número de pruebas:

Comenzar en el bucle más interior. Establecer los demás bucles en sus valores mínimos.

Llevar a cabo las pruebas de bucles simples para el bucle más interior, mientras se mantienen los parámetros de iteración (p. ej.: contadores de bucles) de los bucles externos en sus valores mínimos. Añadir otras pruebas para valores fuera de rango o excluidos.

Progresar hacia fuera, llevando a cabo pruebas para el siguiente bucle, pero manteniendo todos los bucles externos en sus valores mínimos y los demás bucles anidados en sus valores "típicos".

Continuar hasta que se hayan probado todos los bucles.

Bucles concatenados. Los bucles concatenados se pueden probar mediante el enfoque anteriormente definido para los bucles simples, mientras cada uno de los bucles sea independiente del resto. Por ejemplo, si hay dos bucles concatenados y se usa el contador del bucle 1 como valor inicial del bucle 2, entonces los bucles no son independientes. Cuando los bucles no son independientes, se recomienda usar el enfoque aplicado para los bucles anidados.

Bucles no estructurados. Siempre que sea posible, esta clase de bucles se deben rediseñar para que se ajusten a las construcciones de la programación estructurada.

6.10.4.- Pruebas a Utilizar en el Proyecto

Luego de estudiar los tipos de pruebas posibles a realizarse sobre un software, en el proyecto se utilizará el enfoque de pruebas de Caja Negra. Por que como se dijo anteriormente la Prueba de la Caja Negra permite al ingeniero del software obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. Estas pruebas intentan encontrar errores de distintas categorías, como por ejemplo; Funciones incorrectas o ausentes, Errores de interfaz, Errores en estructuras de datos o en accesos a bases de datos externas, Errores de rendimiento y Errores de iniciación y de terminación.

Para este caso se utilizará el enfoque de Pruebas Caja Negra, Pruebas de Causa – Efecto, y las Pruebas de interfaces gráficas de Usuario, además para algunos módulos específicos como reservas se utilizaran pruebas de valores limites para si asegurarse del buen comportamiento del sistema en fechas de reserva extremas.

6.10.5.- Casos De Prueba

En el flujo de trabajo de las pruebas se verifica el resultado de la implementación probando cada construcción, incluyendo tanto construcciones internas como intermedias, así como las versiones finales del sistema a ser entregadas a terceros.

Caso de Prueba	A probar	Condición	Entrada	Salida
Ingresar Venta	Sistema	Estar ingresado en el sistema como administrador o secretaria	Rut empleado y cliente. Lotes y sus cantidades respectivas. Datos relacionados a la forma de pago y al documento de pago.	1.- Datos correctos: ingreso de los datos, actualización de la base de datos 2.- Datos Incorrectos: no realizar transacción avisar del problema solicitar reingreso de los datos. 3.- Faltan datos: avisar y solicitar ingreso del o los datos faltantes. 4.-Rut empleado no valido: avisar y solicitar nuevo ingreso de Rut Valido. 5.- Rut empleado no encontrado: avisar y solicitar

				<p>nuevo ingreso de Rut Valido.</p> <p>6.- Rut cliente no valido: avisar y solicitar nuevo ingreso de Rut Valido.</p> <p>7.- Rut cliente no encontrado: avisar y solicitar nuevo ingreso de Rut Valido.</p> <p>8.- Lote no encontrado: avisar que no fue encontrado</p> <p>9.-stock insuficiente para satisfacer la demanda sistema no realiza la transacción y avisa del problema</p>
Eliminar Venta	Sistema	Estar ingresado en el sistema como administrador o secretaria	Ventas seleccionadas para ser dadas de baja	<p>1.- Datos correctamente seleccionados: eliminar ventas</p> <p>2.- Selección nula de ventas: avisar la no selección de estos</p> <p>3.- Selección nula de ventas: avisar la no selección de estos</p>
Ingresar Reserva	Sistema	Estar ingresado en el sistema como administrador	Rut cliente. Lotes y sus cantidades respectivas a reservar	<p>1.- Datos correctos: ingreso de los datos, actualización de la base de datos</p> <p>2.- Datos Incorrectos: no realizar transacción avisar del</p>

		o secretaria		<p>problema solicitar reingreso de los datos.</p> <p>3.- Faltan datos: avisar y solicitar ingreso del o los datos faltantes.</p> <p>4.- Rut cliente no valido: avisar y solicitar nuevo ingreso de Rut Valido.</p> <p>5.- Rut cliente no encontrado: avisar y solicitar nuevo ingreso de Rut Valido.</p> <p>6.-stock insuficiente para satisfacer la demanda sistema no realiza la transacción y avisa del problema</p>
Modificar Reserva	Sistema	Estar ingresado en el sistema como administrador	Datos de la Reserva a modificar seleccionados	<p>1.- Datos correctamente seleccionados y modificaciones llevadas a cabo con éxito, actualización de la base de datos.</p> <p>2.- Caso contrario: no realizar transacción avisar del problema</p> <p>3.- Selección nula de clientes: avisar la no selección de estos</p> <p>4.-datos obligatorios incompletos. No se pueden</p>

				<p>modificar los datos el sistema avisa el problema y pide ingreso de datos</p>
<p>Eliminar Reserva</p>	Sistema	<p>Estar ingresado en el sistema como administrador o secretaria</p>	<p>Reserva seleccionadas para ser dadas de baja</p>	<p>1.- Datos correctamente seleccionados: eliminar empleados.</p> <p>2.- Selección nula de clientes: avisar la no selección de estos</p> <p>3.- Selección nula de reservas: avisar la no selección de estos</p>
<p>Ingresar Herramienta</p>	Sistema	<p>Estar ingresado en el sistema como bodeguero o administrador</p>	<p>Nombre, Id herramienta y Descripción</p>	<p>1.- Datos correctos: ingreso de los datos, actualización de la base de datos.</p> <p>2.- Caso contrario: no realizar transacción avisar del problema solicitar reingreso de los datos.</p> <p>3.- Faltan datos: avisar y solicitar ingreso del o los datos faltantes.</p> <p>4.-Id herramienta esta utilizado, sistema no realiza el ingreso y avisa del problema</p>
<p>Modificar Herramienta</p>	Sistema	<p>Estar ingresado en el sistema como</p>	<p>Datos de la herramienta seleccionada a modificar</p>	<p>1.- Datos correctamente modificados: actualización de los datos de la herramienta correspondiente</p>

		bodeguero o administrador		<p>2.- Caso contrario: no realizar transacción avisar del problema y limpiar formulario para realizar nueva operación</p> <p>3.- Faltan datos: avisar y solicitar ingreso del o los datos faltantes.</p> <p>4.-Id herramienta a ingresar ya está ocupado por otra herramienta, no realizar modificación y limpiar formulario para realizad nuevamente la operación</p>
Eliminar Herramienta	Sistema	Estar ingresado en el sistema como bodeguero o administrador	Herramientas seleccionadas para ser eliminadas	<p>1.- Datos correctamente modificados: eliminación de las herramientas seleccionadas</p> <p>2.- Caso contrario: no realizar transacción avisar del problema</p> <p>3.- Selección nula de herramienta: avisar la no selección de herramientas</p>
Herramienta Dar de baja	Sistema	Estar ingresado en el sistema como bodeguero o administrador	Herramientas seleccionadas para ser dadas de baja	<p>1.- Datos correctamente manipulados: dar de baja las herramientas seleccionadas.</p> <p>2.- Caso contrario: no realizar transacción avisar del</p>

				<p>problema.</p> <p>3.- Selección nula de herramienta: avisar la no selección de herramientas.</p> <p>4.-herramienta esta prestada a un empleado. No puede dar de baja hasta ser devuelta a bodega.</p>
Herramienta Entregar	Sistema	Estar ingresado en el sistema como bodeguero o administrador	Datos herramientas y datos del empleado seleccionados para entregarle dicha herramienta	<p>1.- Datos correctamente modificados: eliminación de las herramientas seleccionadas.</p> <p>2.- Caso contrario: no realizar transacción avisar del problema.</p> <p>3.- Selección nula de herramienta: avisar la no selección de herramientas</p> <p>4.-Selección nula de empleado. Avisar de la no selección de empleado</p>
Herramienta Devolver	Sistema	Estar ingresado en el sistema como bodeguero o administrador	Datos herramienta a devolver seleccionados	<p>1.- Datos correctamente actualizados: herramienta devuelta a la bodega.</p> <p>2.- Caso contrario: no realizar transacción avisar del problema.</p>

				<p>3.- Selección nula de herramienta: avisar la no selección de herramientas a devolver</p> <p>4.-Selección errónea de fecha de devolución, avisar que la fecha es no es valida</p>
Ingresar Tratos	Sistema	Estar ingresado en el sistema como administrador	Seleccionar datos del empleados e ingresar los datos del trato a realizar	<p>1.- Datos correctamente ingresados: asignación del nuevo trato con el empleado.</p> <p>2.- Caso contrario: no realizar transacción avisar del problema.</p> <p>3.- Selección nula de trato: avisar la no selección de los tratos</p> <p>4.-No asignas monto a cancelar por el trato, sistema avisa tal error y no realiza la transacción hasta completas datos faltantes.</p>
Modificar Tratos	Sistema	Estar ingresado en el sistema como administrador	Seleccionar datos del empleados e ingresar los datos del trato a	<p>1.- Datos correctamente modificados, actualización del trato con el empleado.</p> <p>2.- Caso contrario: no realizar transacción avisar del</p>

			realizar	<p>problema.</p> <p>3.- Selección nula de trato: avisar la no selección de un trato a modificar.</p> <p>4.- No asignas monto a cancelar por el trato, sistema avisa tal error y no realiza la transacción hasta completas datos faltantes</p>
Eliminar Tratos	Sistema	Estar ingresado en el sistema como administrador	Seleccionar datos del tratos a eliminar	<p>1.- Datos correctamente ingresados: asignación del nuevo trato con el empleado.</p> <p>2.- Caso contrario: no realizar transacción avisar del problema.</p> <p>3.- Selección nula de trato: avisar la no selección de los tratos a eliminar.</p>
Ingresar Empleado	Sistema	Estar ingresado en el sistema como administrador	Datos personales del empleado	<p>1.- Datos correctos: ingreso de los datos, actualización de la base de datos.</p> <p>2.- Caso contrario: no realizar transacción avisar del problema solicitar reingreso de los datos.</p> <p>3.- Faltan datos: avisar y solicitar ingreso del o los datos</p>

				<p>faltantes.</p> <p>4.- Rut no valido: avisar y solicitar nuevo ingreso de Rut. Valido.</p>
Eliminar Empleado	Sistema	Estar ingresado en el sistema como administrador	Herramientas seleccionadas para ser dadas de baja	<p>1.- Datos correctamente seleccionados: eliminar empleados</p> <p>2.- Caso contrario: no realizar transacción avisar del problema</p> <p>3.- Selección nula de empleados: avisar la no selección de estos.</p> <p>4.-herramienta esta prestada a un empleado. No puede se puede eliminar el empleado hasta devolver la herramienta</p>
Modificar Empleado	Sistema	Estar ingresado en el sistema como administrador	Datos del empleado a modificar seleccionados	<p>1.- Datos correctamente seleccionados y modificaciones llevadas a acabo con éxito, actualización de la base de datos.</p> <p>2.- Caso contrario: no realizar transacción avisar del problema</p> <p>3.- Selección nula de empleados: avisar la no selección de estos</p>

				4.- Datos obligatorios incompletos. No pueden modificar los datos el sistema avisa el problema y pide ingreso de datos
Ingresar Cliente	Sistema	Estar ingresado en el sistema como administrador	Datos del cliente	<p>1.- Datos correctos: ingreso de los datos del cliente, actualización de la base de datos.</p> <p>2.- Caso contrario: no realizar transacción avisar del problema solicitar reingreso de los datos.</p> <p>3.- Faltan datos: avisar y solicitar ingreso del o los datos faltantes.</p> <p>4.- Rut no valido: avisar y solicitar nuevo ingreso de Rut. Valido.</p>
Modificar Cliente	Sistema	Estar ingresado en el sistema como administrador	Datos del cliente a modificar seleccionados	<p>1.- Datos correctamente seleccionados y modificaciones llevadas a cabo con éxito, actualización de la base de datos.</p> <p>2.- Caso contrario: no realizar transacción avisar del problema</p> <p>3.- Selección nula de empleados: avisar la no</p>

				<p>selección de estos.</p> <p>4.-datos obligatorios incompletos. No pueden modificar los datos el sistema avisa el problema y pide ingreso de datos</p>
Eliminar Cliente	Sistema	Estar ingresado en el sistema como administrador	Herramientas seleccionadas para ser dadas de baja	<p>1.- Datos correctamente seleccionados: eliminar empleados.</p> <p>2.- Caso contrario: no realizar transacción avisar del problema</p> <p>3.- Selección nula de empleados: avisar la no selección de los clientes.</p>

7.- Conclusiones

Como muestra del estado final del proyecto se pueden dar por cumplido la lista de objetivos propuesto al inicio de este, ya que se han depurado todos los modelos necesarios que permitieron terminar con del desarrollo y construcción del sistema.

Objetivos cumplidos:

- Gracias a las reuniones con el cliente se logro recopilar información detallada para poder refinar el modelo relacional de la base de datos que soporte las reglas del negocio, además del modelo de clases base para comenzar la construcción del sistema.
- Se termino la construcción de los 10 módulos: ventas de plantas, modulo bodega, mantenedor de cliente, mantenedor de empleado, modulo de trabajos, modulo de reserva, modulo de productos, modulo proveedores, modulo de cuentas y modulo de informes. Siendo ventas y bodega de vital importancia en los procesos que desarrolla a diario el cliente.
- Se han definido los planes de prueba pertinentes de los módulos, los cuales garantizarán el completo y correcto funcionamiento del sistema.
- Se ha elaborado el manual de usuario con el cual se instruye al cliente para un correcto uso del sistema y así poder reducir y evitar en todo lo que sea posible los errores al utilizar los diferentes módulos del sistema.
- Se logro demostrar la viabilidad económica del proyecto mediante el cálculo del VAN, se estimo el retorno de la inversión es aproximadamente 2 años y los beneficios mensuales del uso del sistema por parte del vivero Ecogreen es de aproximadamente \$450000, estos estimados son gracias a las mejoras en cuanto a ofertas al clientes y formas de pago, además de la optimización del proceso productivo.

Un aspecto importante es la gran ayuda que ha prestado el cliente, ya que sin ellos el avance de este proyecto sería mucho menor, se trabajó junto con el cliente fijando variadas reuniones en donde han tenido mucha paciencia en darnos a conocer la metodología de su trabajo y de sus procesos, además de facilitar documentos muy importantes que ayudaron a realizar un mejor análisis de la situación actual, por lo que, hasta el momento los objetivos y contenidos de este documento han sido de total agrado del cliente y de mucha utilidad para el refinamiento de los modelos y detalles del sistema.

Se ha mostrado el funcionamiento de la situación actual de la producción del vivero Ecogreen y sus desventajas para contraponerlas con el flujo futuro y sus correspondientes mejoras con el sistema, además se realizó un estudio de factibilidad que comprobó la viabilidad del proyecto, y se hizo el estudio sobre los riesgos para poder crear los planes de contingencia y mitigación.

Se han realizado los estudios correspondientes a los tipos de diseño de pruebas de sistema, eligiendo el tipo de diseño más acorde a las existentes limitaciones tanto de tiempo y dinero, se logró crear un plan de prueba lo suficientemente potente, el cual ha ayudado a entregar al final de la etapa de transición un sistema que cumple con todas las funcionalidades fijadas al comienzo del desarrollo y así se pudo cumplir satisfactoriamente la expectativas del cliente.

Sin duda los objetivos planteados a principios del proyecto han sido logrados, pero no por eso no se desarrollarán mejoras. Gracias al nivel de escalabilidad del software ya se están estudiando posibles futuras expansiones del mismo, como por ejemplo trabajar con facturas electrónicas, manejar en más detalle los sueldos (liquidación al empleado), y algo que le interesa mucho al cliente, es poder crear una página web que aparte de dar a conocer la empresa y sus productos, disponga de una especie de carrito de compras que sirva para reservar plantas y así aumentar el número de clientes, sobre todo a países como Holanda, España y Francia que son grandes consumidores del rubro.

8.- Referencias

- [1] Pressman, Roger. (2002). Ingeniería en Software. Madrid: McGraw Hill.
- [2] Sommerville, Ian. (2002). Ingeniería en Software. Ciudad de México: Pearson.
- [3] Jacobson, Ivar. (2000). Proceso unificado de desarrollo de software. Madrid: Addison Wesley.
- [4] <http://www.uml.org/>
- [5] http://www.microsoft.com/spanish/msdn/centro_recursos/vbnet/default.msp
- [6] <http://www.bcn.cl/leyes/pdf/actualizado/30590.pdf> (Biblioteca del Congreso Nacional)
- [7] <http://www ldc.usb.ve/~teruel/ci4713/clases2002/riesgos.html>
- [8] http://download.microsoft.com/download/1/f/4/1f439877-28cc-4591-84f338fc00bd2744/DS_SQL_Server_2005_Express_es.pdf
- [9] <http://www.monografias.com/trabajos5/inso/inso.shtml#ana>
- [10] <http://www.willydev.net/descargas/prev/ExplicaXP.pdf>
- [11] <http://www ldc.usb.ve/~teruel/ci3715/clases/arqCapas.html>
- [12] <http://www.generatorfd.com/Arquitectura.aspx>
- [13] <http://www.monografias.com/trabajos24/arquitectura-cliente-servidor/arquitectura-cliente-servidor.shtml>

9.- Anexos

Documento **Anexos** PDF en: