

Instituto Tecnológico de Costa Rica  
Escuela Ingeniería en Computación

“Sistema de Gestión de Cédulas de Aplicación de  
Insumos Agropecuarios”

Informe final de graduación  
para optar por el grado de Bachiller en  
Ingeniería en Computación

Mauro López Jiménez

Campus Tecnológico Local San Carlos, II semestre  
2018



# Resumen ejecutivo

Las tecnologías de información promueven las mejoras en el campo de la producción comercial transformando prácticas y labores para aumentar la precisión y simplificar los procesos. En el caso de las empresas del sector agrícola comercial, se presenta la necesidad de gestionar los ciclos de aplicación de insumos agropecuarios en sus períodos de cultivo, con el propósito llevar un control detallado de qué se aplica y así facilitar el proceso de auditoría y trazabilidad para la exportación de los productos.

Por lo anterior, se realizó un trabajo de 16 semanas, donde se implementaron los diversos procesos propios de un desarrollo ingenieril del software, orientados en la construcción de una solución para el sector agrícola comercial. Esos procesos corresponden a la propuesta de una solución, definición de los requerimientos, diseño de una plataforma software, desarrollo de la plataforma software y control de la calidad de la plataforma elaborada. Con ello, se obtuvo una solución que permite gestionar los ciclos de aplicación de insumos a la plantación de las fincas agrícolas y facilita el proceso de auditoría para la exportación de la producción; dicha solución se implementó mediante la construcción de tres módulos, los cuales permiten recrear virtualmente la finca y su organización, digitalizar los recursos y sus características, y gestionar ambos actores (terreno y recursos) en el proceso de las aplicaciones.

## Executive summary

Information technologies promote improvements in the field of commercial production by transforming practices and work to increase accuracy and simplify processes. In the case of companies in the commercial agricultural sector, there is a need to manage the cycle of application of agricultural inputs in their growing periods, in order to have a detailed control of what is applied and thus facilitate the process of audit and traceability for the export of products.

Therefore, a 16-week work was carried out, where the various processes of an engineering software development were implemented and oriented towards the construction of a solution for the commercial agricultural sector. These processes correspond to the proposal of a solution, definition of the requirements, design of a software platform, development of the software platform and control of the quality of the elaborated platform. With this, a solution was obtained that allows managing the cycles of application of inputs to the plantation of agricultural farms and facilitates the audit process for the export of production; This solution was implemented through the construction of three modules, which allow to virtually recreate the farm and its organization, digitize resources and their characteristics, and manage both actors (land and resources) in the application process.

## Índice de contenido

1	Introducción.....	9
1.1	Antecedentes .....	10
1.2	Descripción de la empresa .....	10
1.2.1	Misión.....	12
1.2.2	Visión.....	12
1.3	Problema.....	12
1.3.1	Contexto del problema.....	12
1.3.2	Descripción del problema.....	14
1.4	Justificación.....	16
1.4.1	Certificación.....	17
1.4.2	Aplicación de insumos.....	17
1.4.3	Evaluación de costos.....	17
2	Solución planteada.....	19
2.1	Objetivos.....	19
2.2	Patrocinadores.....	20
2.3	Metodología.....	22
2.4	Análisis de los riesgos.....	25
2.5	Cronograma de trabajo .....	27
3	Definición de requerimientos .....	28
4	Diseño de la plataforma software .....	32
4.1	Definición del modelo entidad-relación.....	32
4.2	Definición de la arquitectura del Back-end.....	35
4.3	Definición de mockups.....	39
4.4	Definición de arquitectura de Front-end .....	45
4.5	Análisis y selección de herramientas tecnológicas .....	48
5	Desarrollo de la plataforma software .....	51
5.1	Base de datos .....	51
5.2	Desarrollo del Back-end .....	54
5.3	Desarrollo del Front-end .....	57
6	Control de calidad .....	61
7	Conclusiones .....	63

7.1	Trabajos futuros .....	64
7.1.1	Sistema de muestreo .....	64
7.1.2	Conteo de plantación mediante inteligencia artificial .....	64
7.1.3	Visualización del terreno mediante imágenes GIS.....	65
7.1.4	Ampliar el espectro de impacto de la aplicación .....	65
8	Referencias.....	66
9	Anexos.....	67
9.1	Anexo 1. Requerimientos del Sistema .....	67
9.2	Anexo 2. Pruebas Funcionales del Sistema .....	76

## Índice de tablas

Tabla 1	Objetivos del proyecto.....	19
Tabla 2	Detalle de patrocinadores.....	20
Tabla 3	Definición de metodología.....	22
Tabla 4	Plantilla para documentar riesgos.....	25
Tabla 5	Riesgo RI-01.....	25
Tabla 6	Riesgo RI-02.....	25
Tabla 7	Riesgo RI-03.....	26
Tabla 8	Riesgo RI-04.....	26
Tabla 9	Elementos de estructuras conceptualizadas.....	30
Tabla 10	Tareas realizadas en el diseño de plataforma software.....	32
Tabla 11	Diseño de consultas del API: gestión del terreno.....	38
Tabla 12	Diseño de consultas del API: Gestión de recursos.....	38
Tabla 13	Diseño de consultas del API: Gestión de aplicaciones.....	39
Tabla 14	Contenedores del front-end.....	46
Tabla 15	Componentes del front-end.....	47
Tabla 16	Justificación de herramientas tecnológicas.....	48
Tabla 17	Descripción de consulta del API: Gestión del terreno.....	54
Tabla 18	Descripción de consulta del API: Gestión de recursos.....	56
Tabla 19	Descripción de consulta del API: Gestión de aplicaciones.....	57
Tabla 20	Plantilla para pruebas funcionales de control de calidad.....	61

## Índice de figuras

Figura 1 Organigrama del departamento .....	11
Figura 2 Composición de módulos conceptuales en la gestión del terreno .....	13
Figura 3 Principales procesos en la producción de piña .....	14
Figura 4 Cédula en caso inicial .....	15
Figura 5 Cédula con bifurcación .....	16
Figura 6 Procesos en la producción agrícola de exportación .....	17
Figura 7 Arquitectura de la información del sistema.....	21
Figura 8 Primera etapa de desarrollo de CAI .....	22
Figura 9 Diagrama de Gantt: cronograma de trabajo .....	27
Figura 10 Estrategia para definición de requerimientos.....	28
Figura 11 Conceptualización de actores del contexto del problema.....	29
Figura 12 Plantilla para la documentación de los requerimientos .....	31
Figura 13 Modelo entidad-relación de base de datos .....	33
Figura 14 Arquitectura de la información - modelo entidad-relación: Gestión del terreno .....	34
Figura 15 Arquitectura de la información - modelo entidad-relación: Gestión de Aplicaciones.....	34
Figura 16 Arquitectura de la información - modelo entidad-relación: Gestión de Recursos .....	35
Figura 17 Arquitectura del Back-end .....	36
Figura 18 Back-end: gestión del terreno - gestión de recursos .....	37
Figura 19 Back-end: gestión de aplicaciones .....	37
Figura 20 Mockup de estándar para CRUD .....	40
Figura 21 Mockup del estándar para crear .....	41
Figura 22 Mockup del estándar para editar.....	42
Figura 23 Mockup del estándar para eliminar .....	42
Figura 24 Mockup del estándar para tabla con checkbox .....	43
Figura 25 Mockup de ciclo individual.....	44
Figura 26 Mockup de ciclo individual: detalles .....	45
Figura 27 Reutilización de componentes en arquitectura del front-end .....	46
Figura 28 Arquitectura de la aplicación .....	50
Figura 29 Modelo relacional de la base de datos .....	51
Figura 30 Modelo Relacional - Modelo entidad relación: gestión del terreno.....	52
Figura 31 Modelo Relacional - Modelo entidad relación: gestión de recursos .....	53
Figura 32 Modelo Relacional - Modelo entidad relación: gestión de aplicaciones .....	54
Figura 33 Comportamiento de la aplicación utilizando React-Redux.....	58
Figura 34 Arquitectura del front-end: ejemplo aplicado .....	59
Figura 35 Modal crear .....	60

## 1 Introducción

Las tecnologías de información y sus aplicaciones han cambiado diversos aspectos de actividad humana, incluida la agricultura. Esas tecnologías han llegado a la actividad agropecuaria con una serie de innovaciones tecnológicas, dentro de las que se incluyen la recolección asistida de productos, las estimaciones de insumos para la producción, la predicción de comportamiento de variables de cultivo, la evaluación de las tierras de labranza, entre otras.

Las estimaciones de insumos para la producción constituyen un factor clave en la toma de decisiones de los sistemas de producción agropecuaria modernos. En un sentido amplio, esas decisiones demandan el manejar diversas categorías de información, como ambiental, técnica, económico-financiera, e incluso, comercial, legal y fiscal. Toda esa información se deriva de las interacciones de la empresa agropecuaria con el entorno, donde se requiere comprender los diferentes procesos para optimizar los recursos, y se exige disponer de técnicas y herramientas apropiadas.

Para estimar los insumos en la producción de cultivos se requiere gestionar los ciclos de aplicación de los insumos agropecuarios en sus períodos de cultivo, con el propósito llevar un control detallado de qué se aplica y así facilitar el proceso de auditoría y trazabilidad para la exportación de los productos. Para ello se utiliza la Cédula de Aplicación de Insumos. Ésta es un documento que recoge distintas especificaciones de los insumos agroquímicos que son aplicados a un cultivo en el proceso de producción. Ese documento facilita la administración de las aplicaciones y asegura la trazabilidad del producto, lo que permite dar seguimiento a su colocación en mercados nacionales e internacionales.

El presente trabajo de Práctica Profesional muestra aspectos del proceso de desarrollo de un sistema software para la gestión de Cédulas de Aplicación de Insumos (CAI). El trabajo se realizó durante el segundo semestre de 2018 en la Empresa Avantica Technologies, en su sede ubicada en Ciudad Quesada, San Carlos.

## 1.1 Antecedentes

Las empresas que se dedican a la agricultura realizan una serie de aplicaciones de productos agroquímicos a los cultivos de manera cíclica, con el propósito de defenderlos de plagas y enfermedades, así como potenciar su crecimiento, entre otras razones. Dicho proceso se conoce como ciclo de aplicación. Por otra parte, los mercados tienden a ser cada vez más estrictos en cuanto al control de dichos ciclos de aplicación, por ello exigen lo denominado Cédulas de Aplicación de Insumos (en adelante, CAI) a los cultivos comerciales.

CAI es un documento que posee datos referentes a un ciclo de aplicación, como el ingrediente activo del producto, el nombre comercial, la dosis aplicada, las condiciones ambientales al momento de la aplicación, la fecha en la que se realizó, cuáles fueron las regiones aplicadas, entre otras. De modo que, las CAI facilitan la trazabilidad del cultivo.

En la actualidad la generación de las CAI se realiza a mano, lo que implica un costo excesivo en tiempo, además de ser una práctica propensa a errores. Se pueden utilizar hojas electrónicas, sin embargo, su implementación resulta complicada por tener que manejar alto volumen de información y conocimiento en una estructura como MS-Excel, además de seguir siendo propenso a errores por su poca adaptabilidad al conjunto de alternativas existentes en la ejecución de la cosecha.

Las tecnologías digitales ofrecen la oportunidad de realizar la gestión y trazabilidad de las aplicaciones de productos a cultivos comerciales mediante el uso de tecnologías de bases de datos, aplicaciones web y aplicaciones móviles, motores de inferencia y de recomendaciones.

## 1.2 Descripción de la empresa

Avantica Technologies inició actividades en el año 1995, con oficinas centrales en Silicon Valley, California, y un centro de ingeniería de software en San José, Costa Rica. Actualmente la compañía también tiene sedes en Liberia y San Carlos, además

cuentan con un centro de desarrollo en Perú. Se encuentra entre los más grandes especialistas en servicios de ingeniería de software en Latinoamérica, especializándose en colaborar con compañías establecidas o emergentes para crear productos innovadores. El software se desarrolla en conjunto con los clientes y las relaciones que se mantienen son usualmente de largo plazo y basadas en metodologías rigurosas e ingeniería de calidad. La experiencia en ingeniería de software incluye incursiones en Mobile, Cloud computing, Analytics, Search, Social Networking, Enterprise Web y Fin Tech (financial technology). Los servicios que brinda son: Arquitectura de Software, Desarrollo de Software desktop, web y mobile, Control de Calidad de Software, Servicios Profesionales para Software.



Figura 1 Organigrama del departamento

Avantica cuenta con una estructura definida por departamentos, los cuales se muestran organizados en Figura 1 Organigrama del departamento. El proyecto que se

describe en el presente documento se realizó en la unidad de desarrollo, específicamente en el sector de producción de Avantica San Carlos.

### 1.2.1 Misión

Avantica es un equipo de profesionales talentosos e innovadores desarrollando software de calidad mundial enfocados en que nuestros clientes alcancen sus objetivos de negocios.

### 1.2.2 Visión

La empresa se plantea como visión ser reconocida globalmente como líder por la calidad de sus servicios, productos y talento humano, enfocados en la satisfacción de sus clientes.

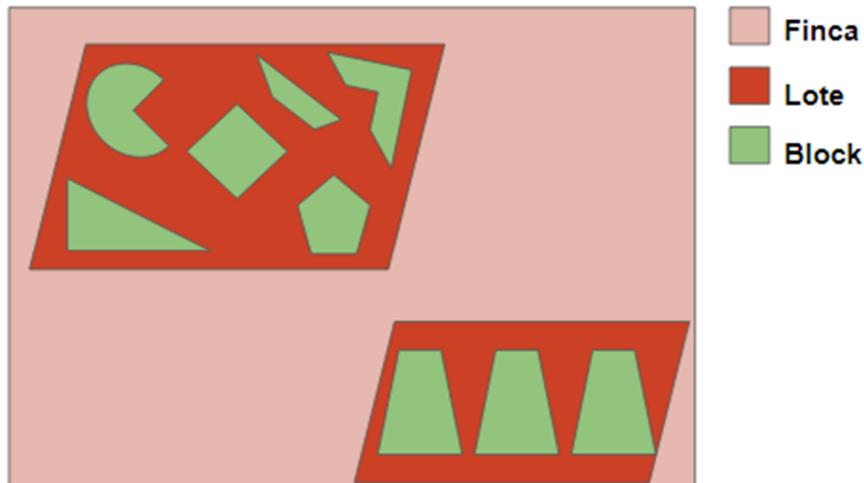
## 1.3 Problema

A continuación se muestra el análisis exhaustivo realizado en el entendimiento de la situación a abordar, lo cual permite esclarecer el contexto del problema, los procesos que interviene en él, las especificaciones que requiere la solución, entre otras cosas. La suma de estas acciones permite obtener información que es de suma utilidad al momento de proponer y construir la solución. A continuación se detalla la secuencia de acciones que permitieron entender el problema.

### 1.3.1 Contexto del problema

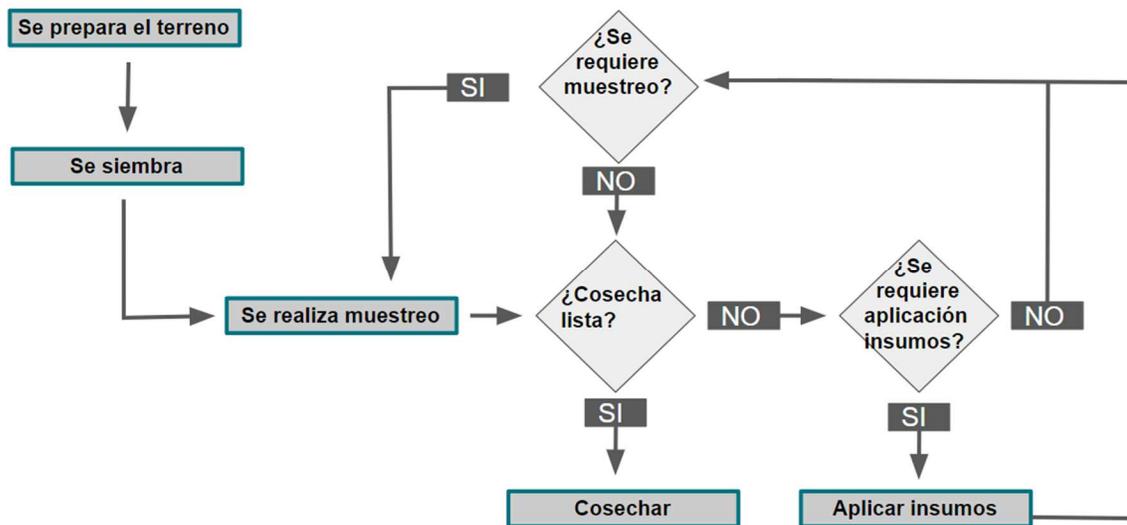
Aunque muchos cultivos requieren de la gestión de las Cédulas de Aplicación de Insumos, para este análisis se toma como referente predeterminado el cultivo de piña.

A su vez, al problema desarrollarse en un ámbito agrícola, es importante definir las estructuras conceptuales dentro del contexto, por lo tanto, en Figura 2 Composición de módulos conceptuales en la gestión del terreno se detalla la manera en que los agricultores dividen sus tierras, la cual se da en una composición de tres niveles, en donde existen fincas que poseen lotes y estos a su vez poseen bloques.



**Figura 2 Composición de módulos conceptuales en la gestión del terreno**

Seguidamente se debe de tener una noción del proceso del cultivo de piña (Figura 3 Principales procesos en la producción de piña) el cual da inicio con la preparación del terreno. Una vez terminado dicho proceso se da paso a la siembra de los hijos de la planta, el cual por lo general se realiza por lote, aunque en ocasiones puede darse tanto por finca como por bloque.



**Figura 3 Principales procesos en la producción de piña**

Una vez que el terreno esté sembrado, se da inicio con los ciclos de aplicaciones de insumos agrícolas, los cuales se encuentran condicionados por los resultados de los muestreos, que por lo general se aplican cada 12 días. Sin embargo, varía según metodología de cada centro productor agrícola, y se pueden dividir en dos tipos: desarrollo y fuerza. Desarrollo consiste en suministrar insumos que permitan la correcta evolución de la planta, mientras que fuerza pretende acelerar el florecimiento de la planta.

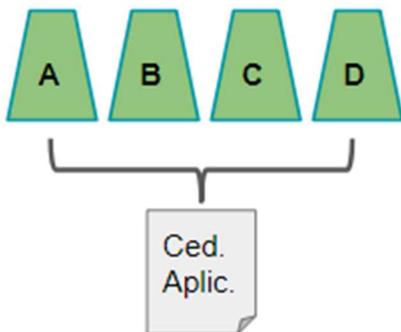
### 1.3.2 Descripción del problema

Los insumos y su respectiva aplicación en los cultivos comerciales corresponden a un proceso que a simple vista pareciese poco minucioso, sin embargo, su ejecución repercute en otros procesos que ocurren en las empresas productoras agrícolas, como lo son la fiscalización, exportación, impacto ambiental, costo económico, entre otros.

Por ende, ignorar las diligencias correspondientes a los ciclos de aplicaciones de insumos de forma detallada y concorde al estado de la plantación y sus requerimientos, repercute negativamente en el desempeño de la empresa.

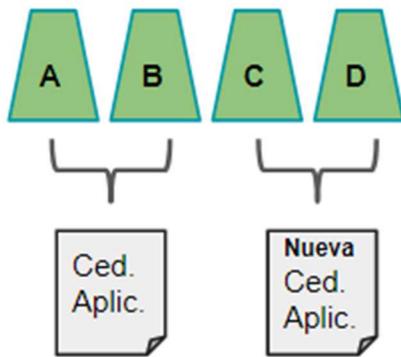
Adicional a ello, se agrega un incidente que afecta directamente el flujo del proceso de la gestión de las cédulas de aplicación, y es que debido a diversos factores externos (floración natural, condiciones del terreno, entre otras), la ejecución de estos ciclos de aplicación se pueden ramificar, es decir, uno o muchos bloques se pueden adelantar o atrasar respecto al resto de la plantación, por lo cual estos requieren cédulas de aplicación diferenciadas al resto y de acuerdo a sus necesidades. De misma manera puede ocurrir la situación descrita por lotes.

Por ejemplo, los bloques A, B, C y D pertenecen al lote X, los cuales registran una cédula de aplicación (Figura 4 Cédula en caso inicial).



**Figura 4 Cédula en caso inicial**

Sin embargo, debido a un factor externo, los bloques C y D sufrieron un adelanto de acuerdo a los otros bloques, por lo cual se generó una nueva cédula de aplicación para estos (Figura 5 Cédula con bifurcación).



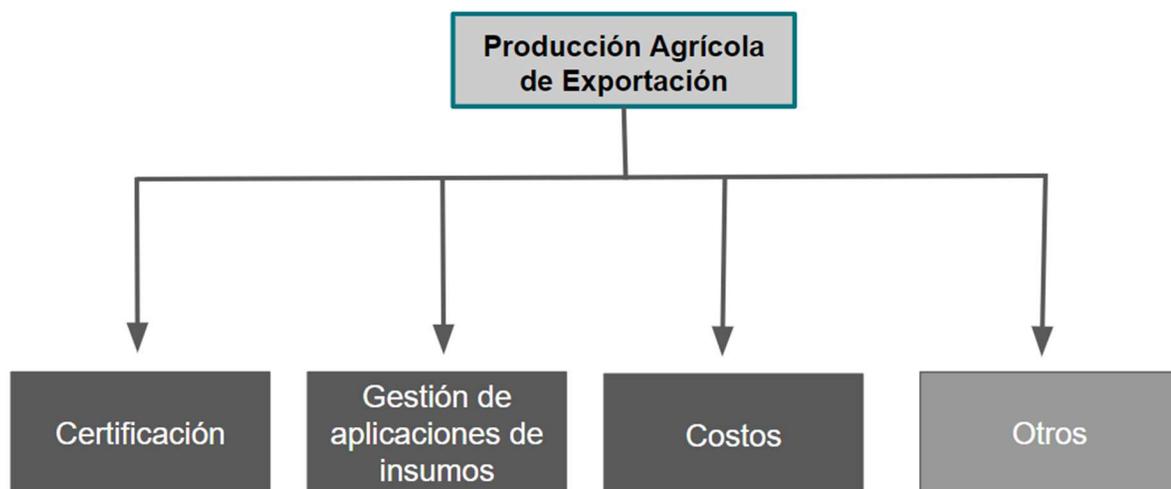
**Figura 5** Cédula con bifurcación

No contemplar los detalles anteriormente descritos conllevaría a realizar un manejo ineficiente de los recursos, tanto de insumos, equipos y mano de obra, con consecuencias negativas en todos los procesos que se ven afectados por la aplicación de los insumos agrícolas, entre los que destacan el impacto ambiental, económico y de rendimiento en cosecha.

Ante todo esto, el cliente requiere gestionar este proceso mediante la creación y supervisión de cédulas de aplicación, y a su vez, poder manejar dichos datos desde cualquier computador con conexión a internet.

## 1.4 Justificación

En la producción agrícola existen gran cantidad de procesos, entre ellos tres importantes que se encuentran relacionados. A continuación se representan en Figura 6.



**Figura 6** Procesos en la producción agrícola de exportación

#### 1.4.1 Certificación

Como parte del proceso para exportar algún producto agrícola, el cliente (comprador de la mercadería agrícola) solicita un conjunto de requisitos, entre ellos el cumplimiento de ciertos estándares y certificados. Para corroborar el cumplimiento de lo solicitado existen entes que se encargan de ello, los cuales solicitan al productor datos que permitan verificar el resultado de diversos análisis realizados al cultivo.

Entre estos datos solicitados, un caso en concreto corresponde a la lista de insumos aplicados, por ende, el productor debe de ser capaz de responder a tal interrogante de manera detallada.

#### 1.4.2 Aplicación de insumos

La aplicación de insumos al cultivo es un proceso cíclico en la producción agrícola. Sin embargo, existen condiciones (como la ramificación en las aplicaciones) que requieren especial cuidado, por lo cual, el productor debe de contemplar estos detalles.

#### 1.4.3 Evaluación de costos

Es importante para el productor conocer el costo de cada aplicación para posteriormente realizar un balance entre costos y ganancias

Así entonces, se pretende desarrollar una herramienta para el productor a través de un sistema software para la gestión de Cédulas de Aplicación de Insumos que permita simplificar los procesos de certificación, aplicación de insumos y evaluación de costos. El sistema simplificará los procesos mediante la trazabilidad de las aplicaciones y facilitará el manejo a través de una interfaz amigable con el usuario.

Por lo tanto, CAI representa una solución tecnológica para el área agrícola, abriendo paso a lo denominado Agricultura de Precisión, lo cual pretende contribuir en el ahorro de recursos como tiempo y dinero, mitigar el impacto ambiental y aumentar la productividad de las fincas usuarias del sistema.

## 2 Solución planteada

A continuación se detallan la serie de pasos efectuados para plantear una solución que solvente las especificaciones del problema.

### 2.1 Objetivos

Teniendo como base la importancia de la gestión de los insumos en las empresas agrícolas debido a su repercusión en diferentes procesos, se plantea una solución, la cuál orientará sus esfuerzos en contribuir en la trazabilidad de las aplicaciones de dichos insumos en los cultivos agropecuarios comerciales. A continuación se detalla en Tabla 1 Objetivos del proyecto los objetivos, tanto general como específicos, del proyecto.

**Tabla 1 Objetivos del proyecto**

<b>Objetivo general</b>
A. Contribuir en la trazabilidad de la aplicación de insumos en los cultivos agropecuarios comerciales mediante un sistema de software para la gestión de cédulas de aplicación de insumos en plantaciones agropecuarias.
<b>Objetivos específicos</b>
A. Definir los requerimientos funcionales para la gestión de los procesos de aplicación de productos a los cultivos
B. Diseñar una plataforma de software para la gestión de los procesos de aplicación de productos a los cultivos
C. Desarrollar una plataforma de software para la gestión de los procesos de aplicación de productos a los cultivos
D. Evaluar una plataforma de software para la gestión de los procesos de aplicación de productos a los cultivos

## 2.2 Patrocinadores

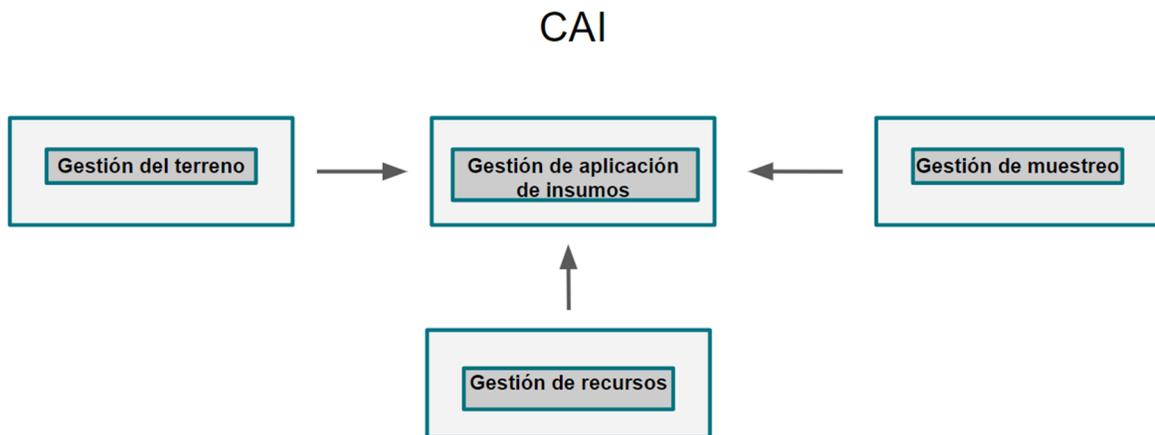
El desarrollo del presente trabajo cuenta con el apoyo de Rodrigo Vargas, ingeniero en computación con más de 15 años de experiencia y Operation Manager de la empresa Avantica San Carlos, quien actuó como orientador en las decisiones relacionadas a aspectos de software y metodología. Por su parte, el ingeniero en producción industrial Andrey Di Palma, operation manager y jefe de aplicaciones de la empresa agrícola Fratelli DPM S.A. brindó soporte en las aristas relacionadas a la lógica del negocio y entendimiento del problema. En Tabla 2 Detalle de patrocinadores se precisa algunos detalles de los mismos.

**Tabla 2 Detalle de patrocinadores**

<b>Nombre</b>	<b>Cargo</b>	<b>Correo</b>
Rodrigo Vargas	Operation Manager - Avantica	rodrigo.vargas@avantica.net
Andrey Di Palma	Operation Manager - Fratelli DPM S.A	a.dipalma14@gmail.com

## 2.3 Detalle de la solución

La propuesta que se presenta corresponde a un sistema de información conformado por cuatro componentes conceptuales, tal y como se muestra en Figura 7 Arquitectura de la información del sistema, que mediante su interacción permitirá el manejo, simplificación y automatización del proceso de aplicación de insumos (y sus tareas correspondientes) dentro de las fincas agrícolas.



**Figura 7** Arquitectura de la información del sistema

La gestión del terreno permitirá al sistema abstraer el entorno físico y su jerarquía (finca>lote>bloque) para realizar diferentes operaciones de la lógica del proceso (siembra, conteo de semilla, aplicación de insumos, muestreos, entre otras) sobre estas jerarquías. Se podrá crear, ver, editar y eliminar fincas, lotes y bloques

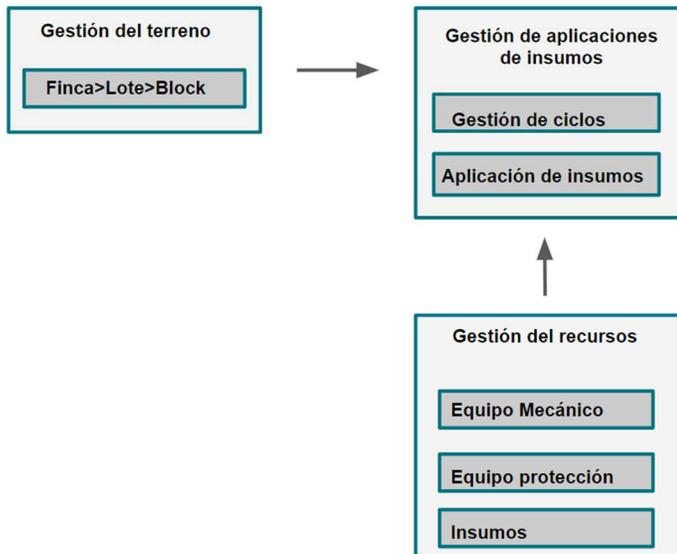
El módulo “Recursos” gestionará aquellos instrumentos que son utilizados en el manejo del terreno, como lo son los insumos agrícolas, el equipo mecánico y el equipo de protección ocupacional.

La gestión de aplicación de insumos corresponde al módulo central, el cual interactúa con el resto de componentes para realizar de manera cíclicas las aplicaciones que a la postre conformarán las cédulas. Para ello gestiona los diferentes elementos del terreno y recursos involucrados.

Por su parte, los muestreos se manejarán mediante la gestión de formularios que permitan validar el estado de la cosecha y mediante este indicador ayudar en la toma de decisiones sobre la aplicación de insumos.

Es importante recalcar que la propuesta consta de 4 pilares previamente descritos, sin embargo su desarrollo se divide en dos etapas, la primera (Figura 8 Primera etapa de desarrollo de CAI) abarca los módulos de gestión del terreno, gestión de recursos y

gestión de aplicación de insumos, y la segunda etapa está conformada por la gestión de muestreos.



**Figura 8** Primera etapa de desarrollo de CAI

Desde el plano arquitectónico, el sistema requiere segmentar su composición en tres bloques claramente definidos: base de datos, back-end y front-end. El segmento de base de datos almacenará los datos y utilizará la abstracción de tablas para estructurar los mismos. El back-end realizará consultas específicas a la base de datos para posteriormente direccionar los datos mediante la interfaz de programación de aplicaciones (más conocida como API). Mientras que el front-end será el encargado de mostrar los datos de una manera intuitiva y amigable al usuario.

## 2.4 Metodología

En Tabla 3 Definición de metodología se detalla la manera propuesta de abarcar cada objetivo definido, lo cual permitió la realización del desarrollo del sistema.

**Tabla 3** Definición de metodología

<p><b>Objetivo General:</b> Contribuir en la trazabilidad de la aplicación de insumos en los cultivos agropecuarios comerciales mediante sistema de software para la gestión de cédulas de aplicación de insumos en plantaciones agropecuarias.</p>
---

<b>Objetivo Específico</b>	<b>Tarea</b>	<b>Meta</b>	<b>Indicador</b>
Definir los requerimientos funcionales para la gestión de los procesos de aplicación de productos a los cultivos	<p><b>A.</b> Entrevistar a diferentes responsables de la gestión de cédulas de aplicación en empresas productoras de piña</p> <p><b>B.</b> Conceptualizar los actores del contexto del problema</p> <p><b>C.</b> Descubrir los elementos que componen dichos actores</p>	<p><b>A.</b> Reporte con la información obtenida de las entrevistas</p> <p><b>B.</b> Realizar diagrama con la identificación de los actores existentes y la relación entre estos</p>	<p><b>A.</b> Entrevista de al menos 5 responsables de la gestión de cédulas de aplicación</p> <p><b>B.</b> Lista priorizada de al menos 50 requerimientos para la gestión de cédulas de aplicación</p>
Diseñar una plataforma de software para la gestión de los procesos de aplicación de productos a los cultivos	<p><b>A.</b> Diseñar un back-end para la gestión de las cédulas de aplicación</p> <p><b>B.</b> Diseñar un front-end para la gestión de las cédulas de aplicación</p>	<p><b>A.</b> El diseño de un back-end para la gestión de las cédulas de aplicación</p> <p><b>B.</b> El diseño de un front-end para la gestión de las cédulas de aplicación</p>	<p><b>A.</b> El diseño de un back-end que incluya el 95 % de los requerimientos funcionales para la gestión de las cédulas de aplicación</p> <p><b>B.</b> El diseño de un front-end que incluya el 95 % de los requerimientos funcionales para la gestión de las cédulas de aplicación</p>
Desarrollar una plataforma de software para la gestión de los procesos de aplicación de productos a los cultivos	<p><b>A.</b> Desarrollar un back-end para la gestión de las cédulas de aplicación</p> <p><b>B.</b> Desarrollar un front-end para la gestión de las cédulas de aplicación</p>	<p><b>A.</b> Un back-end para la gestión de las cédulas de aplicación</p> <p><b>B.</b> Un front-end para la gestión de las cédulas de aplicación</p>	<p><b>A.</b> Un back-end que incluya el 95% de los requerimientos funcionales para la gestión de las cédulas de aplicación</p> <p><b>B.</b> Un front-end que incluya el 95% de los requerimientos funcionales para la gestión de las cédulas de aplicación</p>

<p>Evaluar una plataforma de software para la gestión de los procesos de aplicación de productos a los cultivos</p>	<p><b>A.</b> Diseñar un caso de estudio para evaluar la creación de cédulas de aplicación utilizando la plataforma de software  <b>B.</b> Diseñar un caso de estudio para evaluar la interacción con el cliente utilizando la plataforma de software  <b>C.</b> Diseñar un caso de estudio para evaluar la creación de cédulas de aplicación ramificadas utilizando la plataforma de software  <b>D.</b> Ejecutar un caso de estudio para evaluar la creación de cédulas de aplicación utilizando la plataforma de software  <b>E.</b> Ejecutar un caso de estudio para evaluar la interacción con el cliente utilizando la plataforma de software  <b>F.</b> Ejecutar un caso de estudio para evaluar la creación de cédulas de aplicación ramificadas utilizando la plataforma de software  <b>G.</b> Reportar los resultados de los casos de estudio</p>	<p><b>A.</b> El reporte del caso de estudio para evaluar la creación de cédulas de aplicación  <b>B.</b> El reporte del caso de estudio para evaluar la interacción con los clientes  <b>C.</b> El reporte del caso de estudio para evaluar la creación de cédulas de aplicación ramificadas</p>	<p><b>A.</b> Evaluar el 90% de los requerimientos funcionales de gestión de productos  <b>B.</b> Evaluar el 95% de los requerimientos funcionales de interacción con clientes</p>
---	---	--	---

## 2.5 Análisis de los riesgos

Ningún desarrollo de proyecto está exento de riesgos, por lo tanto, la identificación de los mismos en una etapa temprana del proyecto permite diseñar estrategias que den respuesta ante cualquier situación que los active.

En Tabla 4 Plantilla para documentar riesgos se muestra la plantilla utilizada para documentar los riesgos, en la cual se especifica el código, nombre y categoría del riesgo, de misma forma se considera la causa y el impacto, y se establece la estrategia a implementar con tal de evadir, mitigar o contener el riesgo.

Tabla 4 Plantilla para documentar riesgos

<b>Código</b>	
<b>Nombre</b>	
<b>Categoría</b>	
<b>Causa</b>	
<b>Impacto</b>	
<b>Estrategia de evasión</b>	
<b>Estrategia mitigación</b>	
<b>Estrategia contingencia</b>	

A continuación se muestra la documentación de los riesgos identificados

Tabla 5 Riesgo RI-01

<b>Código</b>	RI-01
<b>Nombre</b>	Usuarios no se adaptan al sistema
<b>Categoría</b>	Personas
<b>Causa</b>	Resistencia al cambio
<b>Impacto</b>	Crítico
<b>Estrategia de evasión</b>	Capacitación al personal y manual de uso del sistema
<b>Estrategia mitigación</b>	Diseño aún más intuitivo
<b>Estrategia contingencia</b>	Aumentar la capacitación al personal y detallar estrictamente el manual de uso

Tabla 6 Riesgo RI-02

<b>Código</b>	RI-02
---------------	-------

<b>Nombre</b>	Disponibilidad de la conexión a internet
<b>Categoría</b>	Tecnológico
<b>Causa</b>	Caída en la red de internet
<b>Impacto</b>	Crítico
<b>Estrategia de evasión</b>	No existe manera de mitigar el riesgo
<b>Estrategia mitigación</b>	No existe manera de mitigar el riesgo
<b>Estrategia contingencia</b>	Cierre seguro del sistema

**Tabla 7 Riesgo RI-03**

<b>Código</b>	RI-03
<b>Nombre</b>	Fallo en la conexión a la base de datos
<b>Categoría</b>	Tecnológico
<b>Causa</b>	Caída de la base de datos
<b>Impacto</b>	Catastrófico
<b>Estrategia de evasión</b>	Diseñar correctamente la estructura de la base de datos
<b>Estrategia mitigación</b>	Realizar pruebas de estrés
<b>Estrategia contingencia</b>	Realiza múltiples llamadas de conexión hasta lograr la reconexión. Restringir el acceso a las funcionalidades principales del sistema

**Tabla 8 Riesgo RI-04**

<b>Código</b>	RI-04
<b>Nombre</b>	Incorrecta estimación del tiempo
<b>Categoría</b>	Personas
<b>Causa</b>	Planificación errónea
<b>Impacto</b>	Crítico
<b>Estrategia de evasión</b>	Juicio experto en el diseño del sistema
<b>Estrategia mitigación</b>	Aplicar estrategias ágiles en la administración de proyectos
<b>Estrategia contingencia</b>	Realiza reajustes en los tiempos y priorizar nuevamente las tareas

## 2.6 Cronograma de trabajo

El Figura 9 Diagrama de Gantt: cronograma de trabajo se puede apreciar el cronograma de trabajo para el desarrollo del proyecto, en donde se estableció un tiempo de 16 semanas, entre las cuales se distribuyen 4 pasos fundamentales: entendimiento del problema y toma de requerimientos (3 semanas), etapa de diseño del programa (6 semanas), desarrollo del programa (9 semanas) y control de calidad (4 semanas)



Figura 9 Diagrama de Gantt: cronograma de trabajo

Como se aprecia en Figura 9 Diagrama de Gantt: cronograma de trabajo, el desarrollo del proyecto dio inicio con la etapa de requerimientos, la cual tuvo como objetivos principales entender y describir los requisitos del programa. Una vez esta etapa se encontró adelantada se autorizó el comienzo en paralelo de la etapa de diseño del sistema para posteriormente arrancar con la etapa de desarrollo.

Cabe resaltar que el proyecto se planteó bajo un desarrollo de tipo incremental, en donde se fragmentó el programa en tres subprogramas que se implementaron cronológicamente en el siguiente orden: gestión del terreno (amarillo), gestión de recursos (verde) y gestión de aplicaciones (azul). Por ello la etapa de diseño, desarrollo y control de calidad se ejecutó tres veces a lo largo del periodo del cronograma.

### 3 Definición de requerimientos

Para la definición de los requerimientos se utilizó una estrategia que corresponde a un híbrido entre la metodología ágil y la metodología convencional, la cual se basó en tres pasos claves para la obtención del resultado y se observa en Figura 10 Estrategia para definición de requerimientos.



Figura 10 Estrategia para definición de requerimientos

El primer paso consistió en la conceptualización de los elementos que inciden en el contexto del problema y la descripción de las asociaciones existentes entre ellos. El resultado de este proceso se aprecia en Figura 11 Conceptualización de actores del contexto del problema, en donde se describen tres pilares conceptuales: Terreno, Aplicaciones y Recursos, siendo Aplicaciones el eje central puesto que estas se aplican sobre Terreno y están conformadas por Recursos.



Figura 11 Conceptualización de actores del contexto del problema

Posteriormente, siguiendo el paso dos de la metodología expuesta en Figura 10 Estrategia para definición de requerimientos, se indagó sobre cada módulo conceptual definido para descubrir los elementos que componen estos módulos. En Tabla 9 Elementos de estructuras conceptualizadas se muestra el resultado de la ejecución del proceso descrito, en donde Terreno se encuentra conformado por una composición de tres niveles: Finca, Lote y Bloque, en orden descendente. Dichas sub-estructuras hacen referencia al espacio físico en donde la empresa agrícola realiza sus operaciones.

Por su parte, el módulo conceptual Recursos está conformado por Insumos, Equipo mecánico y Equipo de protección. Insumos es la abstracción que representa aquellos productos que se aplican sobre el terreno. Equipo mecánico hace referencia al equipo utilizado para aplicar dichos insumos sobre el terreno. Equipo de protección alude a los objetos de salud ocupacional que se requieren para aplicar los insumos.

Por último, Aplicaciones permite describir Cédula, Grupos, Ciclos y Receta. Cédula es el identificador de un bloque en una temporada de plantación, Ciclos es la ejecución de la aplicación de insumos sobre el terreno, Receta es la descripción de los recursos a utilizar en el Ciclo y Grupo es la asociación de Cédulas que comparten un mismo Ciclo.

**Tabla 9 Elementos de estructuras conceptualizadas**

<b>Módulo de la arquitectura de información</b>	<b>Nombre de la estructura</b>	<b>Descripción</b>
Terreno	Finca	Composición en terreno
	Lote	Composición en terreno
	Bloque	Composición en terreno
Aplicaciones	Cédula	Cada bloque tiene una cédula que corresponde al historial de aplicaciones
	Grupos	Los bloques se agrupan en grupos para realizar aplicaciones
	Ciclo	Corresponde a la aplicación de insumos a un grupo
	Receta	Se refiere a las especificación de los recursos a aplicar en el ciclo
Recursos	Insumo	Producto a aplicar
	Equipo mecánico	Equipo mecánico utilizado en el ciclo de aplicación
	Equipo de protección	Equipo de seguridad ocupacional utilizado en el ciclo de aplicación

La secuencia de los pasos descritos en Figura 10 Estrategia para definición de requerimientos para definir los requerimientos concluyó precisamente en ello, determinado los requerimientos. El recorrido de este trayecto permitió traducir el problema planteado en tres módulos generales que posteriormente se desglosaron para identificar las subestructuras que los conforman, lo cual arrojó un total de diez sub-módulos y que estos a su vez fueron traducidos en un total de cuarenta y tres requerimientos.

En Figura 12 Plantilla para la documentación de los requerimientos se observa la plantilla utilizada para la documentación de los requerimientos, en donde se toma a

consideración la identificación, nombre, plano conceptual al que pertenece y descripción del mismo. Los requerimientos son presentados en el Anexo 1.

Identificación		Nombre		Plano conceptual	
Descripción					

Figura 12 Plantilla para la documentación de los requerimientos

## 4 Diseño de la plataforma software

En esta etapa se planificó la solución software al problema planteado en la sección 1.3 del documento, por ello, para estructurar el desarrollo del producto, se realizó un conjunto de tareas, las cuales se aprecian en Tabla 10 Tareas realizadas en el diseño de plataforma software.

**Tabla 10 Tareas realizadas en el diseño de plataforma software**

A. Definición del modelo entidad-relación
B. Definición de arquitectura de back-end
C. Definición de mockups
D. Definición de arquitectura de front-end
E. Análisis y selección de herramientas tecnológicas

A continuación, se detalla cada una de las tareas de la etapa de diseño.

### 4.1 Definición del modelo entidad-relación

Teniendo en consideración los módulos de la arquitectura de la información (gestión del terreno, gestión de recursos y gestión de aplicaciones, en Tabla 9 Elementos de estructuras conceptualizadas) se procedió a analizar el comportamiento de las abstracciones existentes en dicha arquitectura con la finalidad de establecer un modelo que facilite la definición de estructuras para el almacenamiento de los datos.

Este proceso permitió resaltar relaciones y jerarquías entre dichas abstracciones, por lo cual se seleccionó una base de datos relacional o SQL como tipo de base de datos a utilizar en el desarrollo de la aplicación.

A continuación, se muestra el modelo entidad-relación generado a partir de este primer análisis.

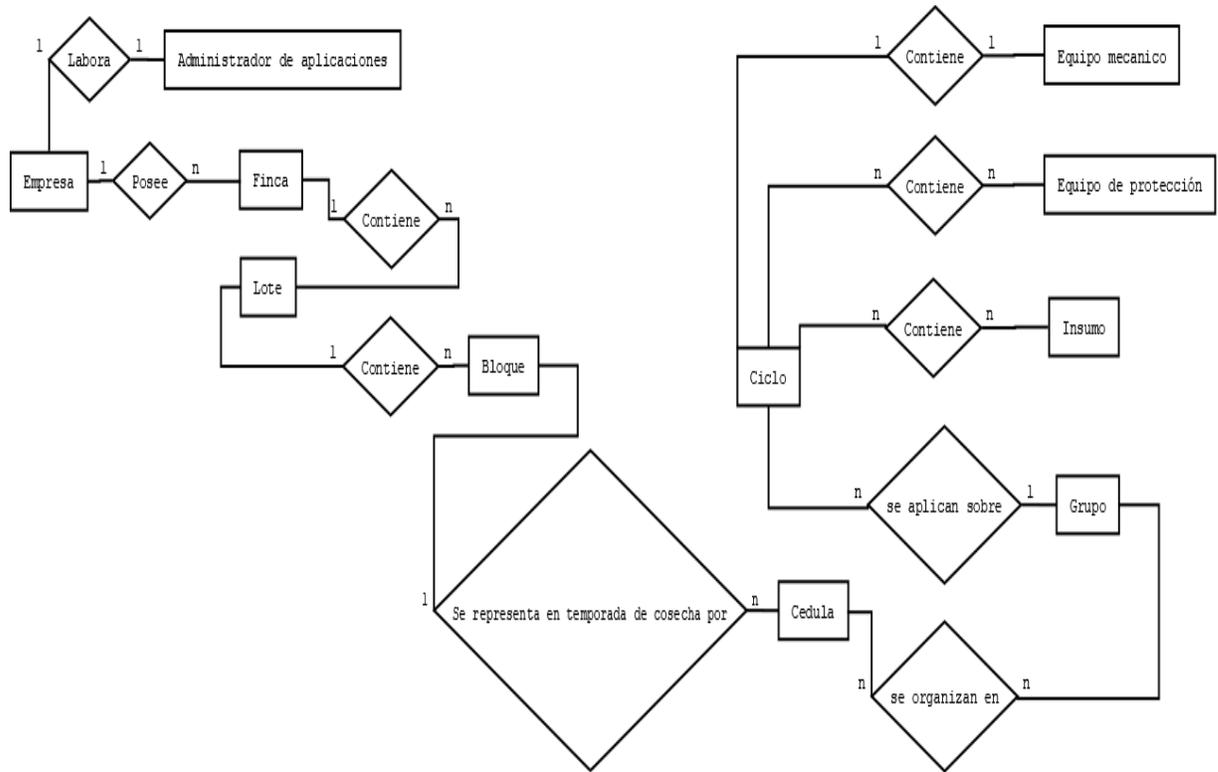


Figura 13 Modelo entidad-relación de base de datos

En Figura 13 Modelo entidad-relación de base de datos se muestra la abstracción de Empresa, la cual tiene un colaborador bajo el rol de administrador de aplicaciones y posee Fincas; que corresponde al terreno que utiliza la empresa para producir su producto.

Continuando, la empresa organiza el terreno en una composición de 3 niveles, en donde Finca corresponde al nivel más alto, mientras que Bloque corresponde a la unidad atómica del módulo de la arquitectura de la información Gestión del terreno.

Véase Figura 14 Arquitectura de la información - modelo entidad-relación: Gestión del terreno.

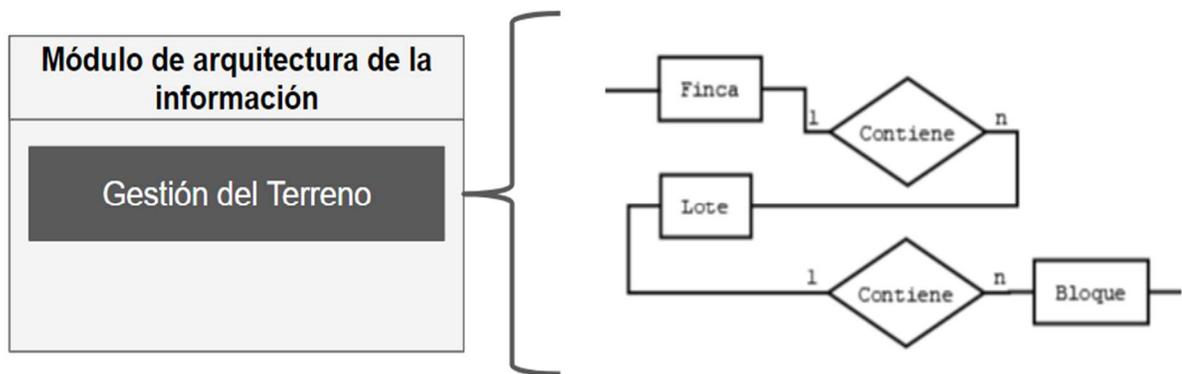


Figura 14 Arquitectura de la información - modelo entidad-relación: Gestión del terreno

Por otra parte, cada Bloque tiene una Cédula, la cual es la identificación de cada bloque para una temporada de cosecha en específico. Estas cédulas se organizan en una abstracción denominada Grupo. Un Grupo es un conjunto de Bloques (representados por sus cédulas) que comparten un mismo Ciclo. El Ciclo hace referencia a la aplicación de ciertos recursos sobre los bloques. En Figura 15 Arquitectura de la información - modelo entidad-relación: Gestión de Aplicaciones se aprecia la traducción de arquitectura de información a modelo relacional para el módulo de gestión de aplicaciones.

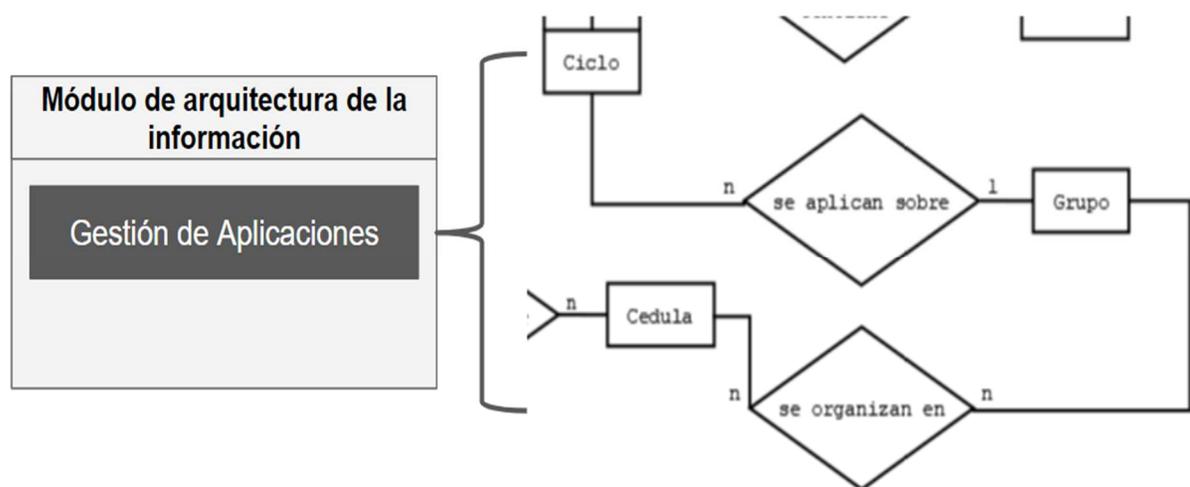


Figura 15 Arquitectura de la información - modelo entidad-relación: Gestión de Aplicaciones

En Figura 16 Arquitectura de la información - modelo entidad-relación: Gestión de Recursos se presenta un Ciclo que contiene una receta, la cual se describe como el conjunto de Insumos que serán aplicados en el Ciclo. La estructura Insumo representa el producto a aplicar en el terreno. A su misma vez un Ciclo tiene Equipo Mecánico que especifica el equipo mecánico utilizado en el Ciclo, y Equipo de Protección que detalla los equipos de protección utilizados en el Ciclo.

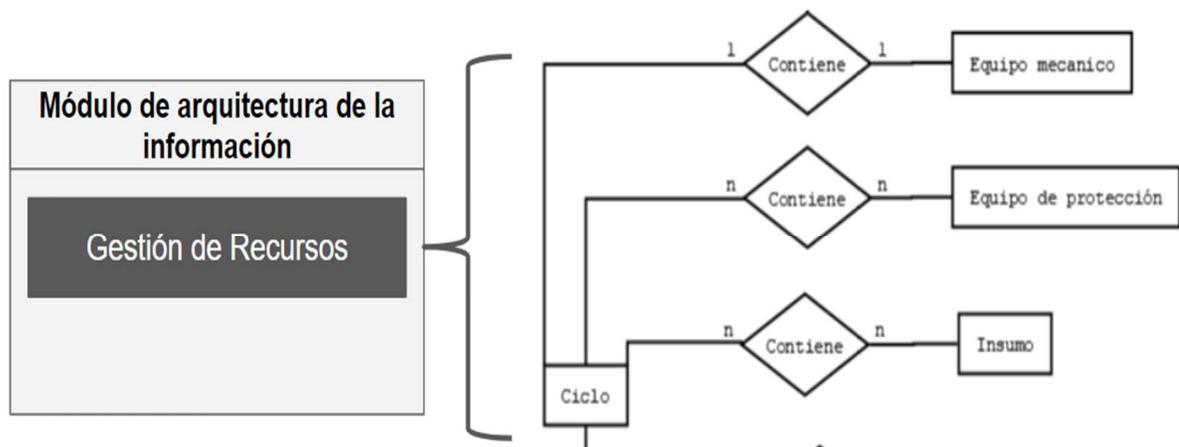


Figura 16 Arquitectura de la información - modelo entidad-relación: Gestión de Recursos

#### 4.2 Definición de la arquitectura del Back-end

En este caso, el back-end funciona como un controlador por sí solo, dado que el modelo se encuentra en la base de datos y la vista corresponde al front-end, sin embargo, tal y como se aprecia en Figura 17 Arquitectura del Back-end, la arquitectura diseñada para esta sección se compone de tres secciones bases que permiten la recepción y transferencia de los datos para el correcto funcionamiento de la aplicación.



Figura 17 Arquitectura del Back-end

App.js es el índice del módulo. Es a este archivo donde llega la petición y direcciona la misma a la ruta correspondiente. El segmento Ruta funciona como enlace entre la petición que recién se acaba de recibir en el archivo App.js y el respectivo controlador. Por su parte, el controlador contiene los métodos que realizan las consultas a la base de datos.

Así entonces, con la presente arquitectura recién descrita (Figura 17 Arquitectura del Back-end) y con la arquitectura conceptual del problema (véase Tabla 9 Elementos de estructuras conceptualizadas) se requiere de un archivo de ruta y un archivo de controlador por cada subestructura de la arquitectura de la información como se muestra en Figura 18 Back-end: gestión del terreno - gestión de recursos y Figura 19 Back-end: gestión de aplicaciones .

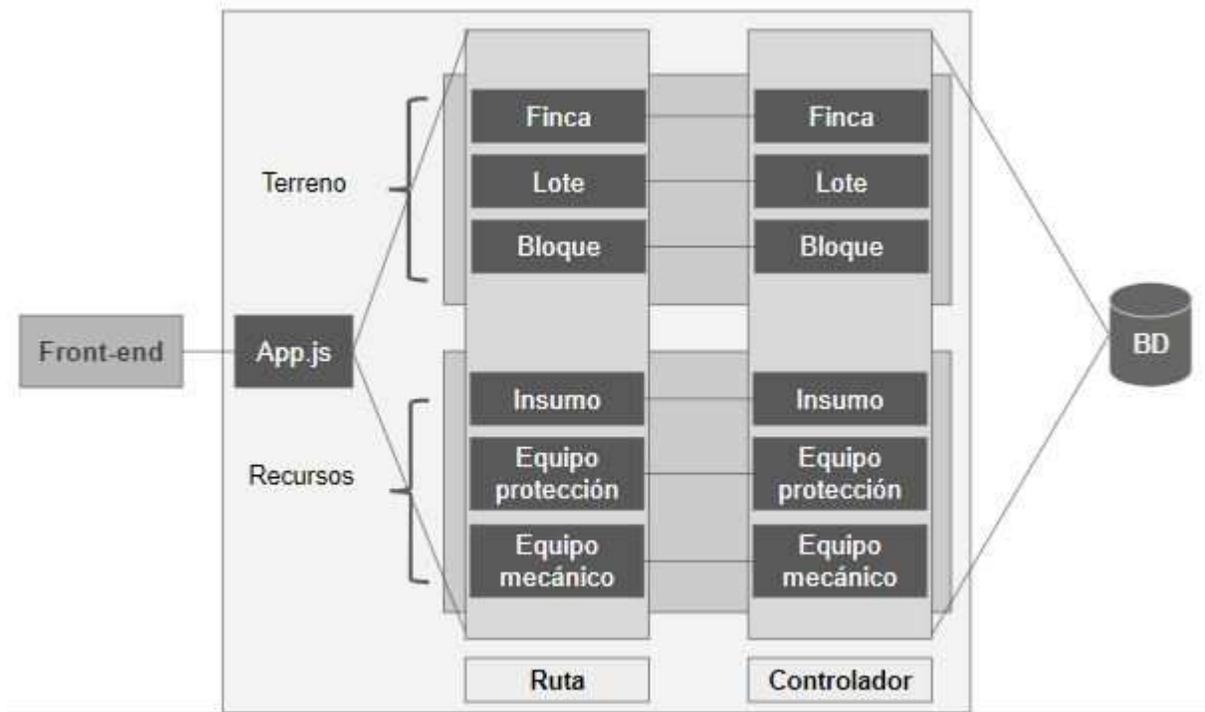


Figura 18 Back-end: gestión del terreno - gestión de recursos

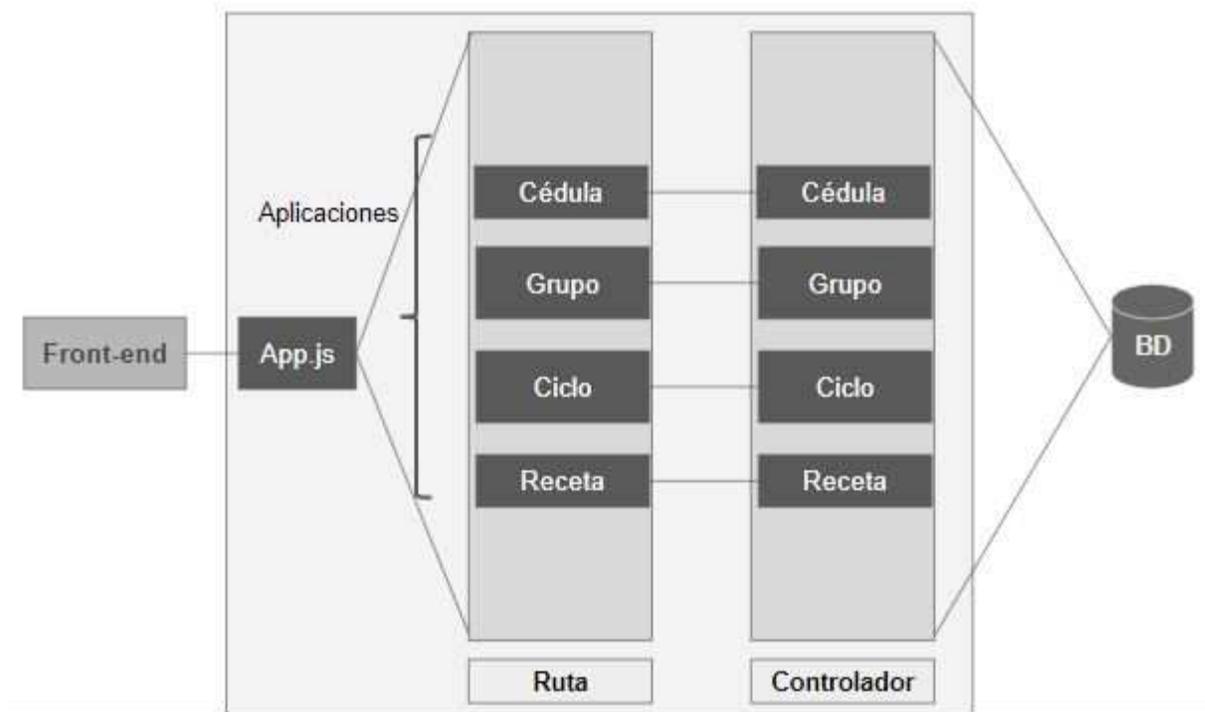


Figura 19 Back-end: gestión de aplicaciones

En Tabla 11 se aprecia el diseño de las consultas a realizar a la base de datos por parte de cada controlador de back-end de la gestión del terreno.

Tabla 11 Diseño de consultas del API: gestión del terreno

Arquitectura de la información	Controlador	Nombre del método	Parámetro	Acción en la base de datos
Gestión del terreno	Finca	getAll()	<i>Sin parametros</i>	Get
		getById()	fincaId	Get
		getAllBloques()	fincaId/blocks	Get
		create()	<i>Sin parametros</i>	Post
		delete()	fincaId	Delete
		update()	fincaId	Put
Gestión del terreno	Lote	getAll()	fincaId	Get
		create()	fincaId	Post
		delete()	lotId	Delete
		update()	lotId	Put
Gestión del terreno	Bloque	getAll()	lotId	Get
		create()	lotId	Post
		delete()	bloqueId	Delete
		update()	bloqueId	Put

En Tabla 12 Diseño de consultas del API: Gestión de recursos se aprecia las consultas a realizar a la base de datos por parte de cada controlador de back-end de la gestión de recursos.

Tabla 12 Diseño de consultas del API: Gestión de recursos

Arquitectura de la información	Controlador	Nombre del método	Parámetro	Acción en la base de datos
Gestión de recursos	Insumo	getAll()	<i>Sin parametros</i>	Get
		getById()	idInsumo	Get
		create()	<i>Sin parametros</i>	Post

		delete()	idInsumo	Delete
		update()	idInsumo	Put
	Equipo de protección	getAll()	<i>Sin parametros</i>	Get
		getById()	idEquipoProteccion	Get
		create()	<i>Sin parametros</i>	Post
		delete()	idEquipoProteccion	Delete
		update()	idEquipoProteccion	Put
	Equipo mecánico	getAll()	<i>Sin parametros</i>	Get
		getById()	idEquipoMecanico	Get
		create()	<i>Sin parametros</i>	Post
		delete()	idEquipoMecanico	Delete
		update()	idEquipoMecanico	Put

En Tabla 13 Diseño de consultas del API: Gestión de aplicaciones se aprecia las consultas a realizar a la base de datos por parte de cada controlador de back-end de la gestión de aplicaciones.

Tabla 13 Diseño de consultas del API: Gestión de aplicaciones

Arquitectura de la información	Controlador	Nombre del metodo	Parametro	Acción en la base de datos
Gestión de aplicaciones	Cedula	getAll()	<i>Sin parametros</i>	Get
		create ()	<i>Sin parametros</i>	Post
	Grupo	getAll()	<i>Sin parametros</i>	Get
		create ()	<i>Sin parametros</i>	Post
	Cedula_grupo	getAll()	<i>Sin parametros</i>	Get
		create ()	<i>Sin parametros</i>	Post
	Ciclo	getAll()	<i>Sin parametros</i>	Get
		create ()	<i>Sin parametros</i>	Post

### 4.3 Definición de mockups

Los mockups son diseños que permiten visualizar los datos que el programa maneja en la interfaz gráfica de usuario. En la sección 4.1 y 4.2 se ha hecho referencia al manejo de los datos del sistema, primeramente en el aspecto de base de datos y

seguidamente en el aspecto de back-end; por lo tanto a continuación se muestra el orden que estos datos tomarán para su presentación visual al usuario.

Nombre de la estructura				
Encabezado	Encabezado	Encabezado	Encabezado	Encabezado
Dato	Dato	Dato	Dato	Opciones
Dato	Dato	Dato	Dato	Opciones
Dato	Dato	Dato	Dato	Opciones
Dato	Dato	Dato	Dato	Opciones
Dato	Dato	Dato	Dato	Opciones
Dato	Dato	Dato	Dato	Opciones



Figura 20 Mockup de estándar para CRUD

En Figura 20 Mockup de estándar para CRUD se representa el standard para realizar las operaciones correspondientes al mantenimiento o CRUD (crear, leer, editar y eliminar, por sus siglas en inglés) de la aplicación. Dicha vista está compuesta en la parte superior izquierda por un título que corresponde al nombre de la estructura que se está renderizando, por ejemplo Fincas, Lotes, Bloques.

En la sección central del mockup se aprecia una tabla, la cual cuenta con su propio encabezado que detalla el título de los datos que se muestran. Cada fila de la tabla corresponde a un elemento de la estructura que se muestra.

A la derecha de cada fila de la tabla se aprecian las opciones que se pueden realizar sobre dicho elemento, ya sea editar, eliminar, ver, entre otras; puesto que las opciones dependen de la lógica de cada estructura en cuestión.

Por último, en la esquina inferior derecha se aprecia un botón, el cual permite crear una nueva instancia de la estructura que se muestra.

Las acciones de editar, eliminar y crear son activadas a través de un botón y son presentadas mediante un modal, como se aprecian en Figura 21 Mockup del estándar para crear, Figura 22 Mockup del estándar para editar, Figura 23 Mockup del estándar para eliminar. Estas acciones son ejecutadas según los datos de la estructura, esto quiere decir que el editar de Insumos será de acuerdo a sus datos y no de acuerdo a los datos de Equipo Mecánico, por ejemplo.

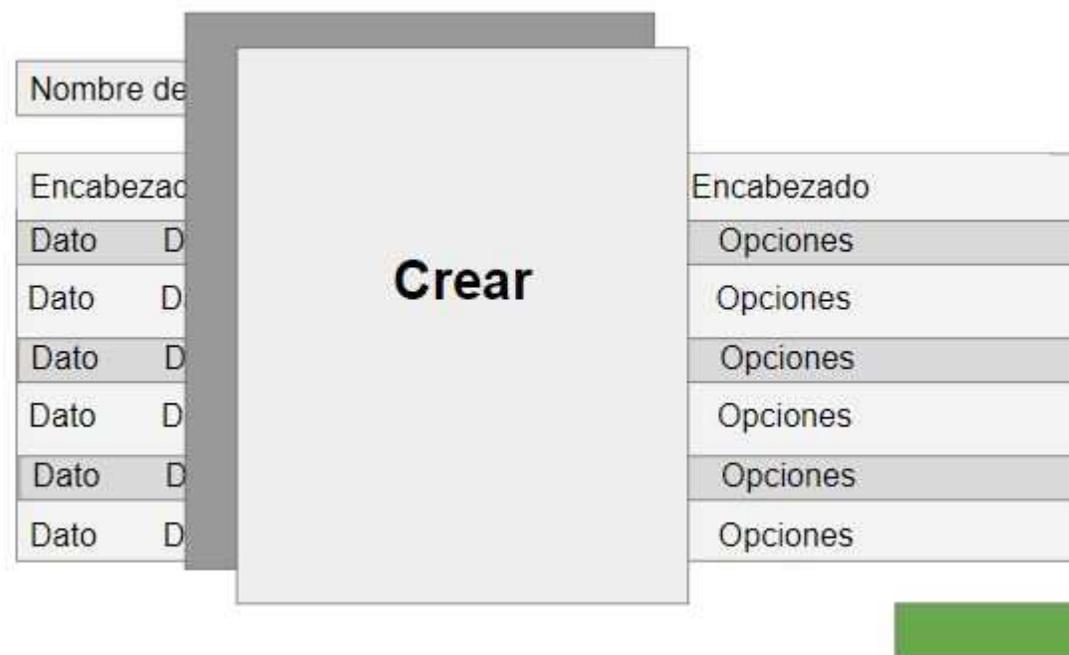


Figura 21 Mockup del estándar para crear



Figura 22 Mockup del estándar para editar

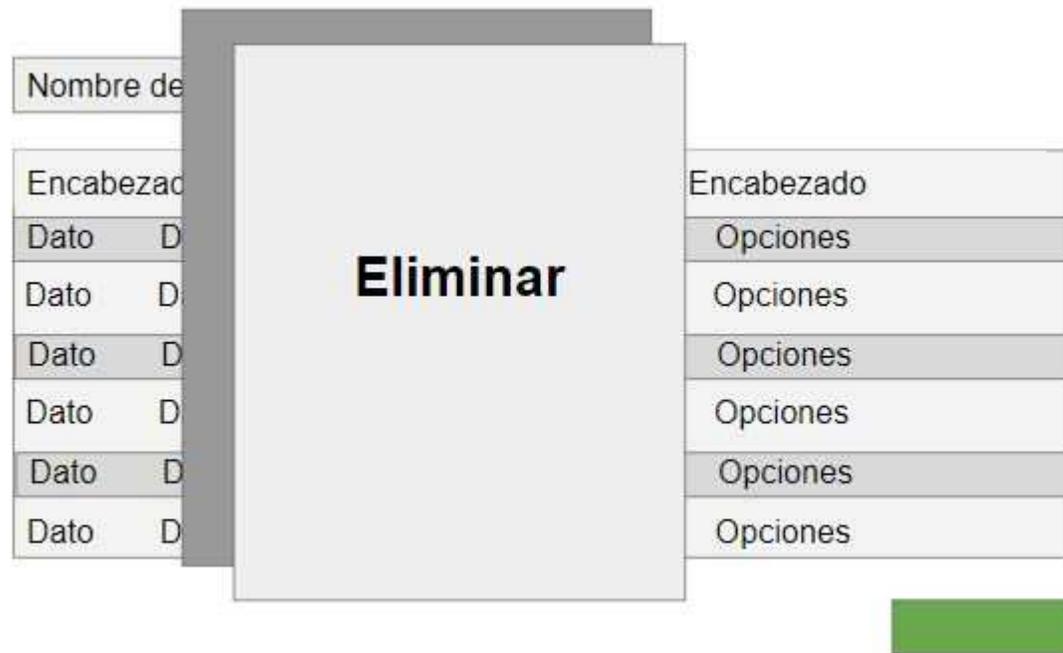


Figura 23 Mockup del estándar para eliminar

Asimismo, a lo largo de la aplicación existen diversos procesos que para una mayor claridad en su flujo, su representación gráfica se debe de escapar del estándar presentado en el estándar para el CRUD (Figura 20 Mockup de estándar para CRUD), tal es el caso de algunas estructuras como Grupo, Receta, lista de Equipos de protección en ciclo, entre otras.

Dichas abstracciones presentan la particularidad de que para su creación se debe de seleccionar dentro de un conjunto de opciones previamente establecidas, por ello no es posible que el usuario digite los datos. Ante esta circunstancia, en Figura 24 Mockup del estándar para tabla con checkbox se muestra una tabla que en su primera columna de izquierda a derecha presenta una caja de chequeo o un “check box”. Esta opción permite al usuario seleccionar las opciones que así desee. En la esquina inferior derecha se aprecia un rectángulo que representa el botón que permite enviar los elementos seleccionados para crear la estructura.



Figura 24 Mockup del estándar para tabla con checkbox

La vista que permite visualizar un ciclo de manera individual (Figura 25 Mockup de ciclo individual) presenta un diseño que se diferencia del estándar. Aunque bien presenta

una tabla que organiza los elementos por filas y los datos por columna, como se ha realizado a lo largo del programa, en esta ocasión los botones activadores de acciones no se presentan en la sección derecha de la tabla; por su parte, estos botones se encuentran en un carrusel en la parte de debajo de la vista.

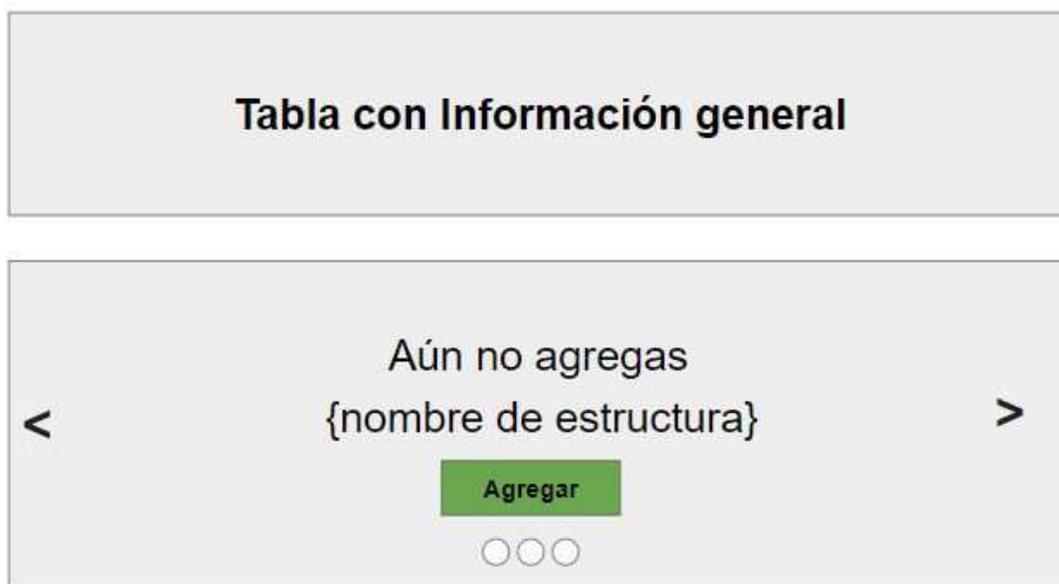


Figura 25 Mockup de ciclo individual

Dicho carrusel está compuesto por tres diapositivas y presenta un comportamiento dinámico, ya que como se ha explicado previamente, un ciclo de aplicación contiene la especificación del equipo mecánico, equipos de protección e insumos a utilizar. Entonces, cuando dichas especificaciones no han sido ingresadas en el sistema, la información de cada diapositiva hace sugerencia a ingresar los datos correspondientes (Figura 25 Mockup de ciclo individual). Por otra parte, cuando estos datos ya han sido ingresado, la información de la diapositiva invita al usuario del sistema a ver los detalles con la posibilidad de realizar acciones de de ver, editar o eliminar (Figura 26 Mockup de ciclo individual: detalles).

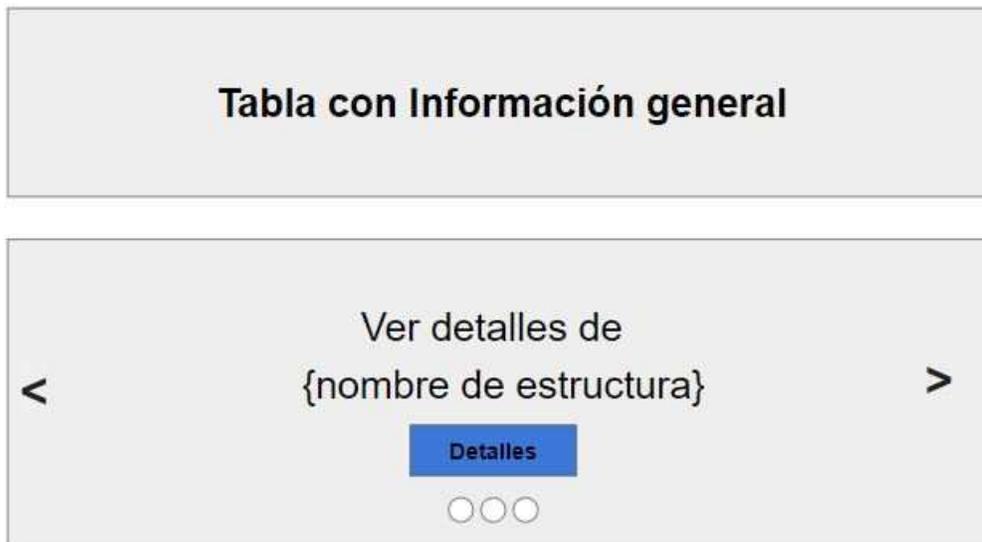


Figura 26 Mockup de ciclo individual: detalles

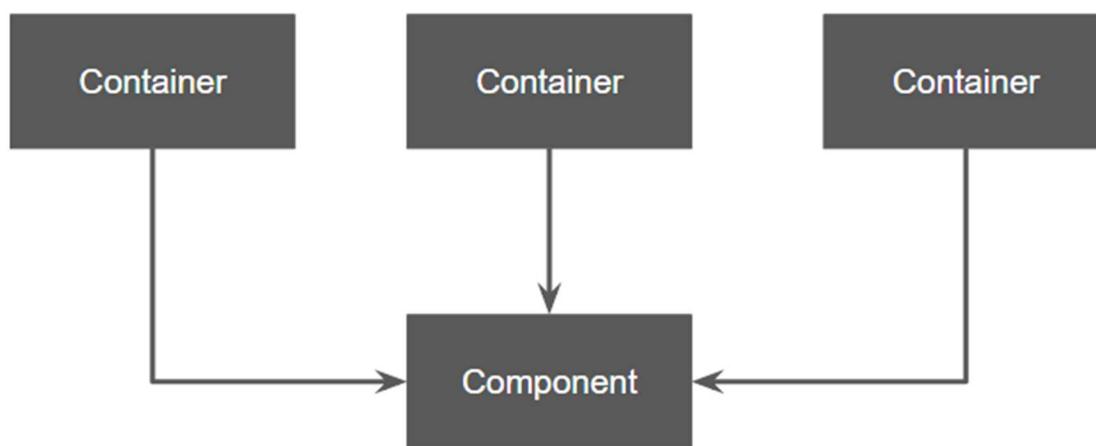
#### 4.4 Definición de arquitectura de Front-end

Para concretar los diseños expuestos en la sección 4.3 y poder brindar un medio de representación de los datos y comunicación entre el usuario y la lógica del sistema se debe de diseñar una arquitectura que soporte la parte gráfica (sección 4.3: definición de mockups) y la parte lógica (sección 4.2: definición de arquitectura del back-end).

Con la intención de facilitar la definición de dicha arquitectura, se parte de dos abstracciones base: “contenedor” y “componente”. Un contenedor corresponde a un archivo que permite manejar la parte lógica y las acciones de una clase, mientras que un componente hace referencia a una sección de código que permite renderizar una vista. Por lo tanto se tiene que el contenedor está relacionado con la recepción de los datos y la lógica que en ellos se aplica, mientras que el componente tiene una finalidad más gráfica puesto que permite moldear la manera en la que estos datos se ven.

Con el front-end teniendo como pilares las dos abstracciones recién comentadas, se permite realizar una estrategia que facilite la reutilización de código en la etapa de

desarrollo, lo cual se basa en el principio de no resolver el mismo problema dos veces y consiste en identificar factores comunes para extraer estos en una instancia aparte y posteriormente renderizar hacia ellos cuando se es necesario. Visualmente dicha estrategia se aprecia en Figura 27 Reutilización de componentes en arquitectura del front-end, en donde tres componentes diferentes renderizan sus datos a un mismo componente. Cabe destacar que el componente no es un archivo estático y dentro de él puede haber lógica que permita mostrar u ocultar datos de acuerdo a las condiciones establecidas.



**Figura 27 Reutilización de componentes en arquitectura del front-end**

El desarrollo de las etapas previas de la sección de Diseño de la plataforma software (definición del modelo entidad-relación, definición de arquitectura del back-end, definición de mockups) facilita la identificación de contenedores y componentes de la aplicación. En Tabla 14 Contenedores del front-end se aprecian los contenedores, mientras que en Tabla 15 Componentes del front-end, los componentes.

**Tabla 14 Contenedores del front-end**

Plano conceptual	Nombre de contenedor
Gestión del terreno	Finca

	Lote
	Bloque
Gestión de recursos	Insumos
	Equipo mecánico
	Equipo de seguridad
Gestión de aplicaciones	Ciclo de aplicación
	Grupo
	Cédula
	Receta
	Equipo mecánico en ciclo de aplicación
	Equipo de seguridad en ciclo de aplicación

Tabla 15 Componentes del front-end

Nombre del componente	Descripción
Tabla	Renderiza en una tabla los datos y las opciones a realizar de un componente
Tabla de seleccionar	Tabla que además de renderizar los datos de un componente es un formulario que se rellena seleccionando filas de la tabla mediante un checkbox. Cada fila de la tabla es un elemento del componente
Tabla acordeón	Tabla que al igual que las dos anteriores, renderiza datos de componentes. Sin embargo tiene la particularidad de desplegar información adicional de cada fila desplegando una fila nueva, simulando un acordeón.
Modal crear	Ventana que se antepone a la ventana principal de la aplicación, la cual contiene un formulario que permite crear elementos de algún contenedor

Modal editar	Ventana que se antepone a la ventana principal de la aplicación, la cual contiene un formulario que permite editar elementos de algún contenedor
Modal eliminar	Ventana que se antepone a la ventana principal de la aplicación, la cual permite confirmar o cancelar la eliminación de un elemento de un contenedor
Cargando	Componente que realiza una animación gráfica que simula un ciclo. Es utilizado mientras el sistema tarda en renderizar los datos de una consulta previa
Jumbotron	Componente de la vista que permite albergar en el otros componentes (tablas por ejemplo) para una renderización más clara de los datos
Carrusel	Es un mostrador de diapositivas, las cuales se repiten cíclicamente

#### 4.5 Análisis y selección de herramientas tecnológicas

Una vez realizado el diseño de la base de datos, del back-end y del front-end se procedió a seleccionar las herramientas tecnológicas a utilizar, lo cual se aprecia en Tabla 16 Justificación de herramientas tecnológicas.

**Tabla 16 Justificación de herramientas tecnológicas**

Nombre de herramienta	Justificación
MySQL	La lógica del problema incide en el uso de una base de datos relacional. Por su parte, ser una herramienta Open Source y su bajo nivel de concurrencia en la modificación de datos fueron factores claves para optar por dicho gestor de base de datos

React.js	React cuenta con una orientación a componentes, lo cual coincide con la arquitectura diseñada en la sección 4.4. De igual manera su buen rendimiento gracias al DOM virtual y el ciclo de vida que presentan los componentes en las aplicaciones React benefician la etapa de desarrollo y mantenimiento del programa
Redux	Su implementación incide positivamente en la etapa de desarrollo y mantenimiento puesto que facilita el manejo de los estados de los diversos componentes de la aplicación
Node.js	Es una herramienta Open Source que se encuentra optimizada con el motor V8 de Google, lo cual mejora su rendimiento. Además utiliza npm, gestor de paquetes más grande del mundo, lo cual garantiza soporte y variedad en las herramientas de apoyo
Docker	Recrear entorno de desarrollo y producción

Una vez que se definieron los detalles referentes a las tecnologías a utilizar (véase Tabla 16 Justificación de herramientas tecnológicas) y el diseño de los módulos estructurales (base de datos, back-end y front-end), se tuvo como resultado la arquitectura de la aplicación que se aprecia en Figura 28 Arquitectura de la aplicación.

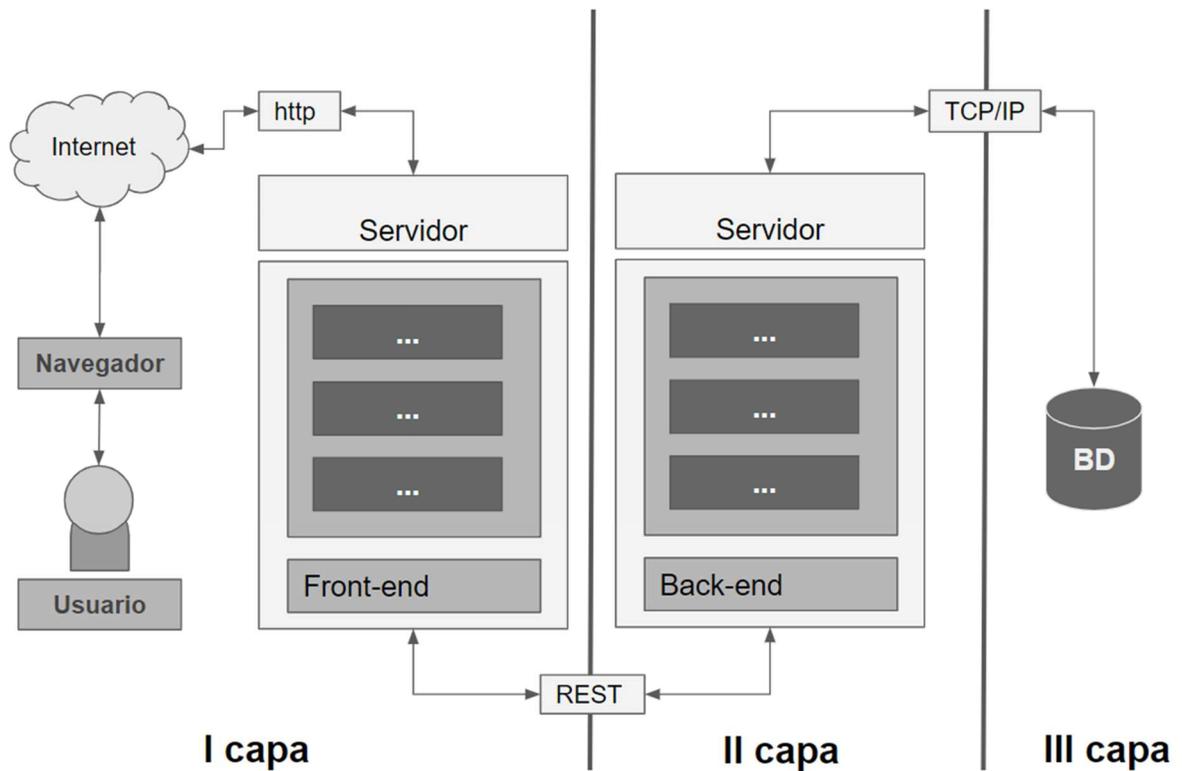


Figura 28 Arquitectura de la aplicación

La arquitectura de la aplicación corresponde a Cliente/Servidor con un total de tres capas, en donde la primera capa representa al usuario y la interfaz, la segunda capa está conformada por el back-end y la tercera por la base de datos. La primera capa se comunica mediante protocolo HTTP con la segunda capa, esta capa intermedia se comunica mediante REST con el back-end y el back-end implementa TCP/IP para realizar diversas operaciones en la base de datos

## 5 Desarrollo de la plataforma software

### 5.1 Base de datos

Atendiendo al trabajo realizado en la sección de diseño en donde se obtuvo el modelo relacional de la base de datos (Figura 13 Modelo entidad-relación de base de datos), se procedió a traducir este modelo al modelo relacional, el cual se observa en Figura 29 Modelo relacional de la base de datos.

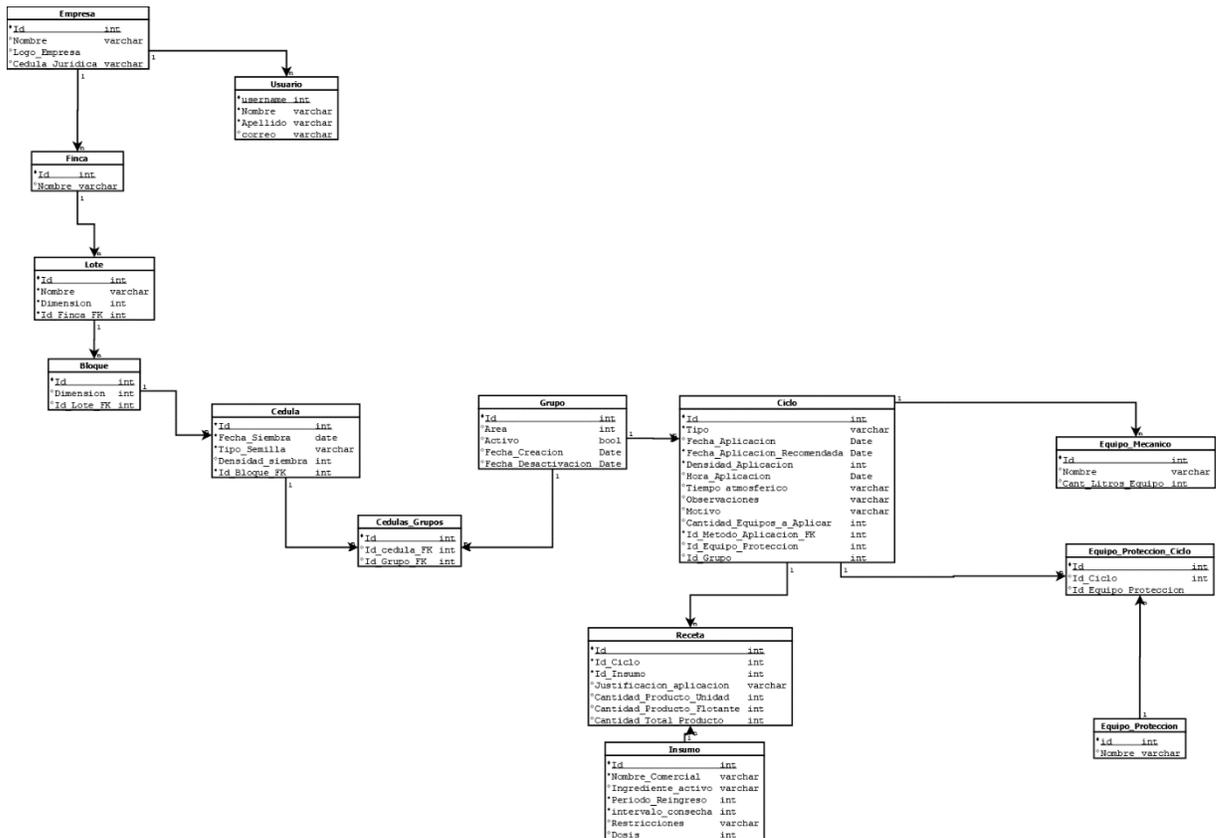


Figura 29 Modelo relacional de la base de datos

En total se crearon 14 tablas, entre las cuales hay 57 atributos y 3 tablas intermedias. Seguidamente se aprecia en Figura 30 Modelo Relacional - Modelo entidad relación: gestión del terreno, Figura 31 Modelo Relacional - Modelo entidad relación: gestión de recursos y Figura 32 Modelo Relacional - Modelo entidad relación: gestión de aplicaciones el proceso de

consolidación de las tablas desde el modelo entidad relación hasta el modelo relacional.

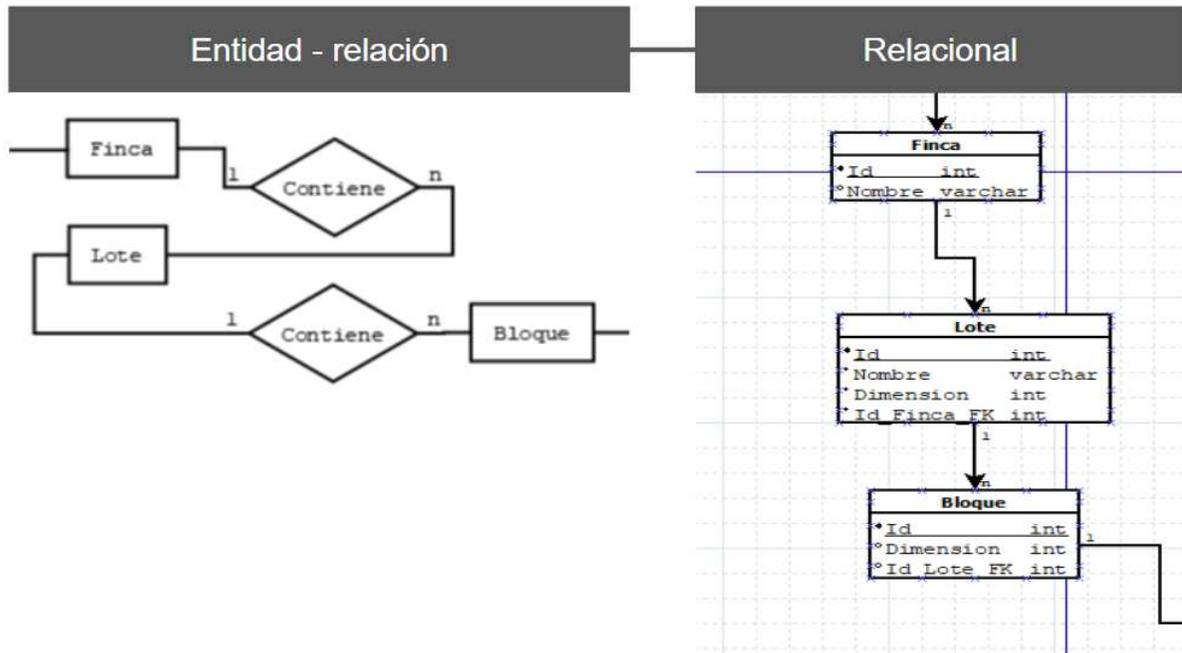


Figura 30 Modelo Relacional - Modelo entidad relación: gestión del terreno

La transición del modelo Entidad-Relación al Relacional (y posteriormente a código) en la gestión del terreno (Figura 30 Modelo Relacional - Modelo entidad relación: gestión del terreno) se da de manera habitual. Las cardinalidades se mantienen, por lo cual únicamente se crea la tabla y se agregan los atributos

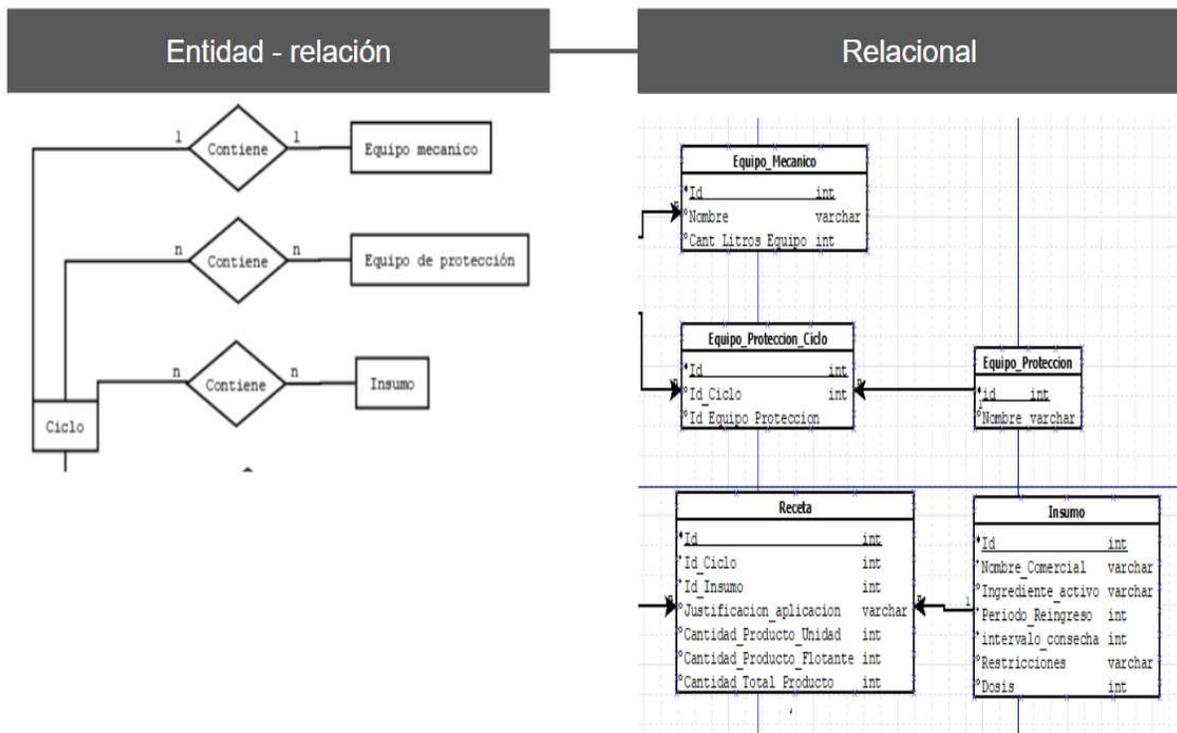


Figura 31 Modelo Relacional - Modelo entidad relación: gestión de recursos

La gestión de recursos (Figura 31 Modelo Relacional - Modelo entidad relación: gestión de recursos) presenta alteraciones en la traducción de los modelos, puesto que la relación muchos a muchos (n a n) en las tablas "Insumo" y "Equipo de protección" con "Ciclo" genera dos tablas nuevas en el modelo relacional: "Receta" (relación Ciclo-Insumo) y "Equipo de protección ciclo" (relación Ciclo-Equipo de protección). Estas tablas intermedias están compuestas por el identificador de cada tabla.

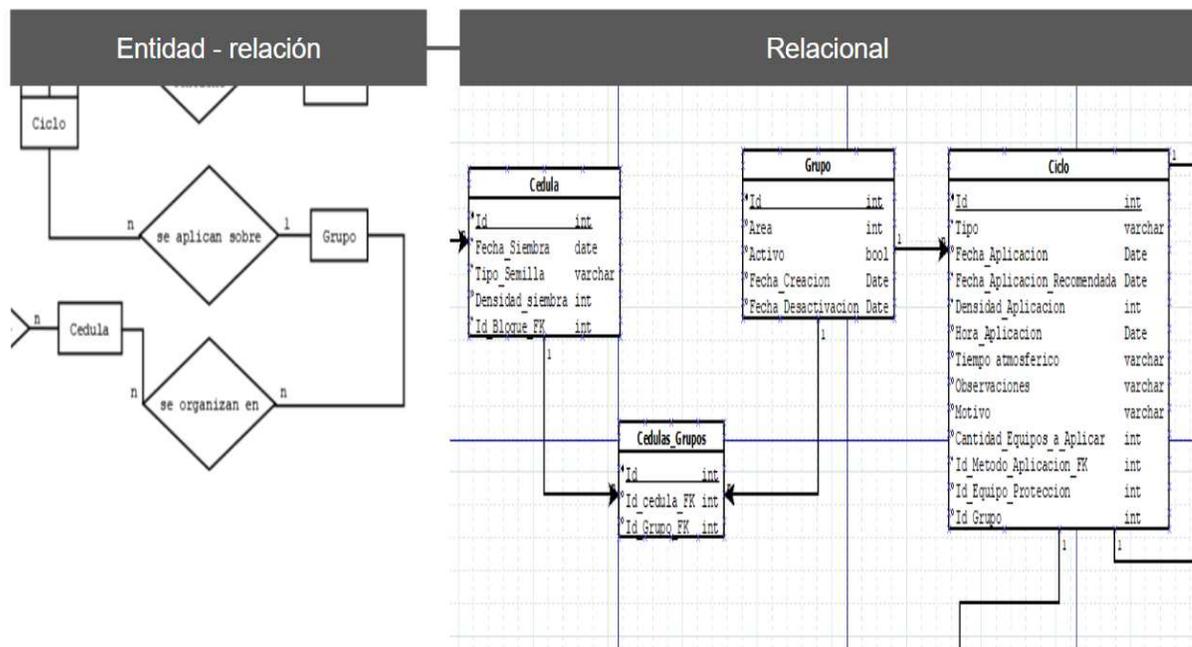


Figura 32 Modelo Relacional - Modelo entidad relación: gestión de aplicaciones

El módulo gestión de aplicaciones (Figura 32 Modelo Relacional - Modelo entidad relación: gestión de aplicaciones) presenta, igual que el caso anterior, la implementación de lo denominado tabla intermedia, debido a que una cédula pueden pertenecer a muchos grupos y un grupo puede contener muchas cédulas, lo cual da una relación n a n.

## 5.2 Desarrollo del Back-end

En la sección de diseño se comprendió la arquitectura del back-end y se especificaron las consultas a realizar a la base de datos. En esta sección se muestra el detalle de cada dirección del API con su controlador, el nombre del método y la acción que ejecuta, así como también el retorno de dicha ejecución. (Véase Tabla 17 Descripción de consulta del API: Gestión del terreno, Tabla 18 Descripción de consulta del API: Gestión de recursos, Tabla 19 Descripción de consulta del API: Gestión de aplicaciones).

Tabla 17 Descripción de consulta del API: Gestión del terreno

Controlador	Nombre del método	Acción del método	Retorno

Finca	getAll()	Get	Lista con objetos que representan los datos de la tabla finca
	getById()	Get	Objeto con todos los datos de la tabla finca
	getAllBloques()	Get	Lista de objeto con los lotes asociados a la finca
	create()	Post	Objeto creado
	delete()	Delete	Id del objeto a eliminar
	update()	Put	Objeto actualizado
Lote	getAll()	Get	Lista con objetos que representan los datos de la tabla lote
	create()	Post	Objeto creado
	delete()	Delete	Id del objeto a eliminar
	update()	Put	Objeto actualizado
Bloque	getAll()	Get	Lista con objetos que representan los datos de la tabla bloque
	create()	Post	Objeto creado
	delete()	Delete	Id del objeto a eliminar
	update()	Put	Objeto actualizado

Tabla 18 Descripción de consulta del API: Gestión de recursos

Controlador	Nombre del método	Acción del método	Retorno
Insumo	getAll()	Get	Lista con objetos que representan los datos de la tabla insumo
	getById()	Get	Objeto con todos los datos de la tabla finca
	create()	Post	Objeto creado
	delete()	Delete	Id del objeto a eliminar
	update()	Put	Objeto actualizado
Equipo mecánico	getAll()	Get	Lista con objetos que representan los datos de la tabla equipo mecánico
	getById()	Get	Objeto con todos los datos de la tabla equipo mecánico
	create()	Post	Objeto creado
	delete()	Delete	Id del objeto a eliminar
	update()	Put	Objeto actualizado
Equipo de seguridad	getAll()	Get	Lista con objetos que representan los datos de la tabla equipo de seguridad
	getById()	Get	Objeto con todos los datos de la tabla equipo de seguridad
	create()	Post	Objeto creado
	delete()	Delete	Id del objeto a eliminar
	update()	Put	Objeto actualizado

Tabla 19 Descripción de consulta del API: Gestión de aplicaciones

Controlador	Nombre del método	Acción del método	Retorno
Cédula	getAll()	Get	Lista con objetos que representan los datos de la tabla cedula
	create()	Post	Objeto creado
Grupo	getAll()	Get	Lista con objetos que representan los datos de la tabla grupo
	create()	Post	Objeto creado
Cedula_Grupo	getAll()	Get	Lista con objetos que representan los datos de la tabla cedula_grupo
	create()	Post	Objeto creado
Ciclo	getAll()	Get	Lista con objetos que representan los datos de la tabla ciclo
	create()	Post	Objeto creado

### 5.3 Desarrollo del Front-end

En la sección de diseño se estableció el comportamiento de las vistas del programa, lo cual permitió definir una arquitectura orientada a contenedores y componentes. De manera consecutiva, la especificación de la arquitectura del front-end permitió seleccionar las herramientas tecnológicas a utilizar, las cuales corresponde a React.js y Redux.

Recordando las características de dichas herramientas se tiene que React es una librería de código libre que permite la construcción de interfaces de usuario, mientras que Redux es una librería que facilita el manejo de los estados de los componentes

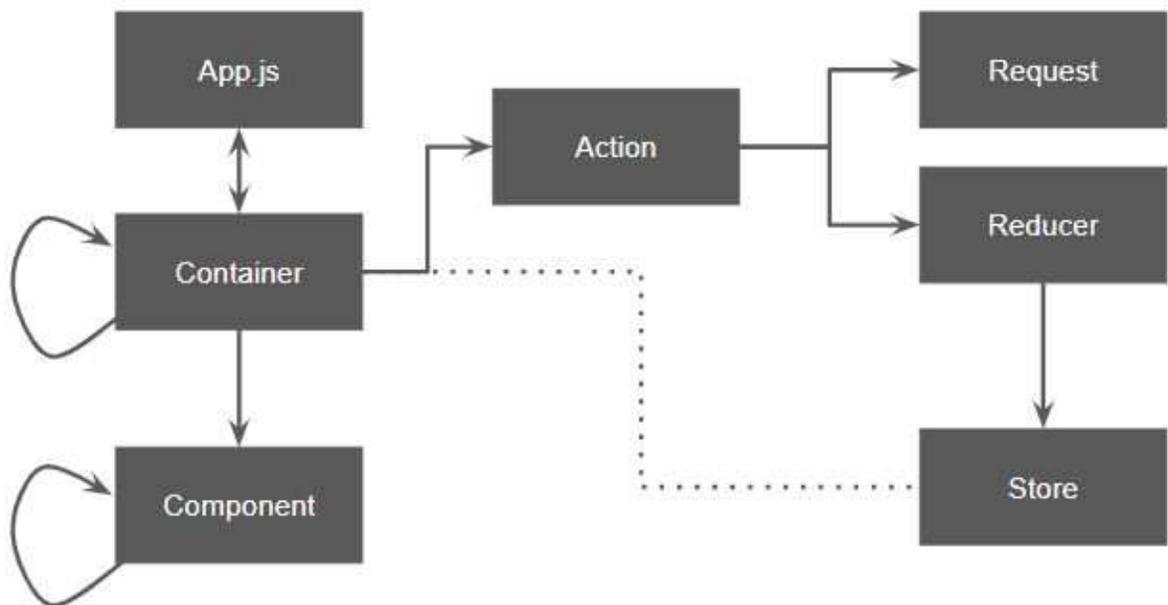


Figura 33 Comportamiento de la aplicación utilizando React-Redux

Como parte de sus características, en Figura 33 Comportamiento de la aplicación utilizando React-Redux se muestra el flujo del comportamiento de la aplicación utilizando React-Redux. En dicha figura se aprecia un archivo principal, App.js, el cual administra los contenedores a referenciar. Estos contenedores son los mismos de los cuales se hizo referencia en la sección de diseño del front-end, por lo tanto, su papel dentro del flujo de la aplicación es realizar la lógica de cada abstracción que se trate, como por ejemplo, realizar una acción, direccionar un componente o referenciar a otro contenedor.

Las acciones son aquellas instrucciones que repercuten en una consulta al back-end, este proceso genera un ciclo que se compone por dos secuencias: la consulta de los datos en primera instancia y la implementación de los mismos por consiguiente. La consulta realiza la petición al back-end, el cual se encarga de realizar las gestiones para satisfacer lo pedido. Por su parte, el reducer o reductor escucha la respuesta a la petición realizada para direccionar estos datos al almacén local del front-end, el cual mediante un seleccionador se conecta a sus respectivos contenedores y así tener los

datos de la consulta en el contenedor para direccionarlos a los componentes necesarios para las vistas



Figura 34 Arquitectura del front-end: ejemplo aplicado

En Figura 34 Arquitectura del front-end: ejemplo aplicado se aprecia la arquitectura del front-end combinada con la arquitectura de React-Redux, que recién se acaba de comentar. En la figura se muestra un contenedor, el cual corresponde a Lotes, que realiza el ciclo de consulta al back-end (realizar consulta, escuchar respuesta, almacenar datos y direccionarlos al contenedor) para obtener los datos necesarios que permitan implementar los componentes de la vista, en esta instancia, el componente Tabla.

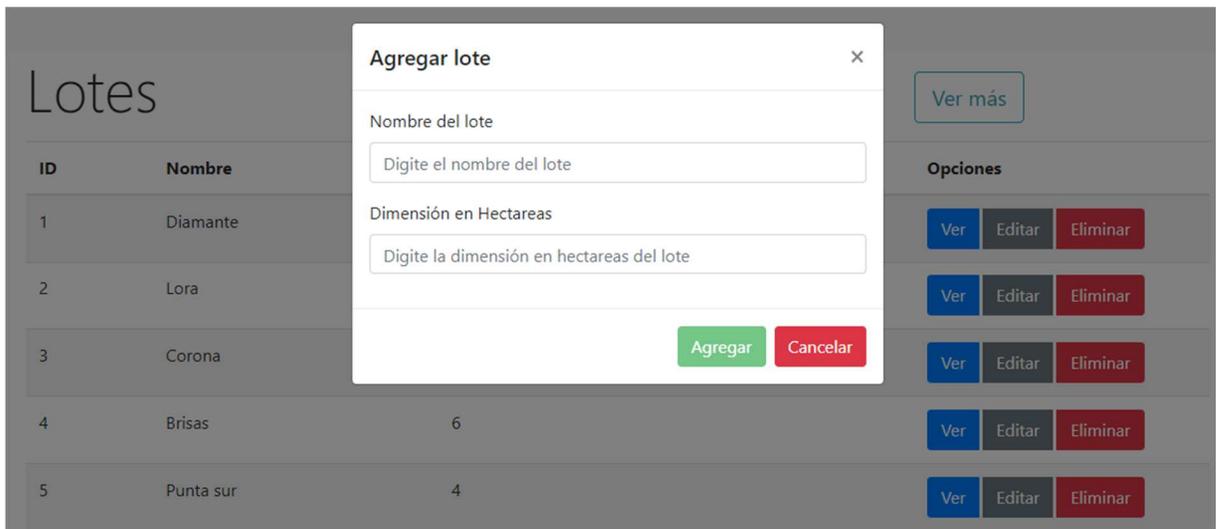


Figura 35 Modal crear

Como se mostró en Figura 33 Comportamiento de la aplicación utilizando React-Redux, los componentes y contenedores pueden redireccionar a otros componentes y contenedores, esto se puede ver implementado en Figura 35 Modal crear, donde desde un contenedor Lote se direcciona a un componente tabla y este a su vez se renderiza a otro componente modal que permite la funcionalidad de crear un elemento. Este proceso permitió el desarrollo de las diversas vistas del front-end.

## 6 Control de calidad

En la sección 2.6 del documento, Cronograma de trabajo, se comentó la distribución de las tareas a realizar en el proyecto. En dicha sección se mostró la segmentación de las etapas en tres secciones, mostrando un comportamiento incremental en el tipo de desarrollo de la aplicación; como cierre de cada una de estas etapas se realizó un proceso de control de calidad al segmento desarrollado.

El control de calidad resulta un proceso fundamental, y no opcional, en el desarrollo de productos software. Las pruebas de control de calidad que se aplican al finalizar cada etapa de desarrollo corresponden a pruebas funcionales, las cuales corresponden a pruebas exhaustivas de caja negra que permiten validar el cumplimiento de los requerimientos del programa.

Tabla 20 Plantilla para pruebas funcionales de control de calidad

Nombre de la prueba	Identificador de la prueba	Identificador de requerimiento involucrado	Módulo conceptual
<b>Precondiciones</b>	<ul style="list-style-type: none"><li>• .</li><li>• .</li><li>• .</li></ul>		
<b>Activador</b>			
<b>Descripción</b>			
<b>Caso</b>			
<b>Resultado</b>			
<b>Detalle</b>			
<b>Estado de la prueba</b>			

En tabla Tabla 20 Plantilla para pruebas funcionales de control de calidad se muestra el esquema del formato utilizado para documentar las pruebas, en donde se toma en consideración el nombre de la prueba, identificador de la misma, requerimiento en estudio y módulo conceptual al que pertenece tal requerimiento. Posteriormente se establecen cuáles

son las precondiciones de la prueba, se describe el camino feliz de la misma y se detalla cual es el activador que da pie a los casos de estudio. Seguidamente, se muestra el título del caso que se está probando, el resultado y alguna anotación correspondiente. Por último se anuncia el resultado de la prueba funcional. El resultado de dichas pruebas se aprecia en Anexo 2. Pruebas Funcionales del Sistema

## 7 Conclusiones

1. A lo largo de 16 semanas se implementaron los diversos procesos propios de un desarrollo ingenieril del software, los cuales fueron metódicos y orientados a la construcción de una solución para el sector agrícola comercial. Estos procesos corresponden a la propuesta de una solución, definición de los requerimientos, diseño de una plataforma software, desarrollo de la plataforma software y control de la calidad de la plataforma elaborada; los cuales fueron ejecutados en el mismo orden mencionado, presentando 3 iteraciones (una por cada módulo conceptual definido) en los procesos de desarrollo y control de calidad.
2. La propuesta de solución tomó como entrada del proceso un problema, el cual se estudió y se entendió con tal de brindar una propuesta que diera respuesta a las necesidades existentes. Dicha respuesta representó el parámetro de entrada para dar inicio con la definición de requerimientos, con el cual se identificó los módulos conceptuales de la solución planteada y se desglosó sus funcionalidades en tareas pequeñas con la finalidad de tener un listado de labores que permitieran la planificación del desarrollo de la solución.
3. El diseño de la plataforma tomó en consideración el listado de requerimientos para proponer una arquitectura que permitiera soportar los requisitos del programa y, a su vez, analizar los módulos conceptuales definidos previamente para plantear estructuras y bocetos que representaran las especificaciones de la solución, así que, su implementación dio como resultado la arquitectura del front-end, back-end y base de datos, los protocolos de comunicación entre cada módulo de la arquitectura, además del diseño del back-end y el front-end.
4. La etapa de desarrollo recibió los productos de las secciones previas, por lo tanto, a partir de las tareas a efectuar (requerimientos) y los diseños de cómo implementarlos, se procedió a realizarlos. La salida de esta sección correspondió a una pieza de código que solventara parte de la solución planteada.

5. Finalizando la secuencia de pasos realizados, se realizó el control de calidad, donde se acogió la lista de requerimientos y el resultado de la etapa correspondiente de desarrollo, y se comparó el resultado obtenido con el resultado esperado y de acuerdo a la calificación de dicha comparación se certificó el desarrollo con garantía de calidad.
6. El cumplimiento de los pasos propuestos en el proyecto permitió la construcción de una solución que permite gestionar los ciclos de aplicación de insumos a la plantación de las fincas agrícolas y facilita el proceso de auditoría para la exportación de la producción; dicha solución se implementó mediante la construcción de tres módulos, los cuales permiten recrear virtualmente la finca y su organización, digitalizar los recursos y sus características, y gestionar ambos actores (terreno y recursos) en el proceso de las aplicaciones.

## 7.1 Trabajos futuros

El desarrollo del presente trabajo corresponde a un primer paso en la construcción de un sistema escalable que permita integrar diversos módulos y mejoras, por consiguiente a continuación se comentan los trabajos a futuros a partir de esta primera aproximación.

### 7.1.1 Sistema de muestreo

Los muestreos corresponden a un proceso que infiere en la gestión de las cédulas de aplicación de insumos, el cual consiste en realizar una visita a campo y evaluar diversas variables de la plantación, el resultado de dichas mediciones permite tomar decisiones en cuanto a los insumos a aplicar en los ciclos y la frecuencia de estos. De tal manera que se establece una aplicación móvil que facilite la medición de tales variables mediante formulario como un trabajo que se debe de implementar a la plataforma a futuro.

### 7.1.2 C conteo de plantación mediante inteligencia artificial

Otro proceso existente en las fincas agrícolas comerciales que infieren de alguna manera en la gestión de cédulas de aplicación de insumos es el conteo de la plantación, puesto que teniendo dicho número se pueden realizar estimaciones más precisas. Actualmente dicho proceso se realiza manual, es decir, un colaborador de la empresa realiza el conteo en algunos bloques de diferentes lotes y posteriormente realiza un promedio para estimar el total de plantas sembradas. No obstante, el uso de tecnología de vanguardia como lo es la inteligencia artificial puede solventar dicha tarea mediante el análisis de imágenes aéreas y el reconocimiento de figuras en ellas.

### 7.1.3 Visualización del terreno mediante imágenes GIS

Actualmente el sistema abstrae la gestión del terreno en una composición de tres niveles, definiendo finca, lotes y bloques (mencionados en orden de jerarquía), dichas abstracciones se ven representadas textualmente en la interfaz del programa, más sin embargo no existe ninguna representación gráfica (mapas o imágenes) de dichas abstracciones. Con tal de brindar una guía al usuario que le permita ubicarse fácilmente en el contexto del terreno, se plantea agregar imágenes GIS para la visualización de la representación del terreno.

### 7.1.4 Ampliar el espectro de impacto de la aplicación

En esta primera etapa de desarrollo, el sistema da respuestas a planteamientos del área operacional. No obstante, los datos que dichas respuestas generan se pueden utilizar para ampliar la longitud de su inferencia al brindar datos que faciliten la toma de decisiones gerenciales.

## 8 Referencias

1. Di Palma, comunicación personal, 11 de Junio de 2018
2. Di Palma, comunicación personal, 24 de Agosto de 2018
3. Ministerio de Agricultura y Ganadería. (2010). Manual de Buenas Prácticas Agrícolas para la Producción de Piña (ananas Comosus L.) Recuperado el 30 de Julio de <http://www.mag.go.cr/bibliotecavirtual/F01-9646.PDF>
4. Reactjs.org. (2018). React – A JavaScript library for building user interfaces. [online] Disponible en: <https://reactjs.org/> [Consultado 7 Agosto. 2018].

## 9 Anexos

### 9.1 Anexo 1. Requerimientos del Sistema

Identificación	RF-01	Nombre	Creación fincas	Plano conceptual	Gestión del terreno
Descripción	El usuario del sistema podrá crear fincas con su respectivo nombre.				

Identificación	RF-02	Nombre	Visualización de fincas	Plano conceptual	Gestión del terreno
Descripción	El usuario del sistema podrá ver la identificación y nombre de sus fincas asociadas				

Identificación	RF-03	Nombre	Editar fincas	Plano conceptual	Gestión del terreno
Descripción	El usuario del sistema podrá editar el nombre de sus fincas asociadas				

Identificación	RF-04	Nombre	Eliminar fincas	Plano conceptual	Gestión del terreno
Descripción	El usuario del sistema podrá eliminar sus fincas asociadas				

Identificación	RF-05	Nombre	Creación lotes	Plano conceptual	Gestión del terreno
Descripción	El usuario del sistema podrá crear lotes asociados a una finca especificando el nombre y la dimensión en hectáreas.				

Identificación	RF-06	Nombre	Visualización de fincas	Plano conceptual	Gestión del terreno
Descripción	El usuario del sistema podrá ver el identificador, nombre y dimensión de los lotes asociados a sus fincas				

Identificación	RF-07	Nombre	Editar lotes	Plano conceptual	Gestión del terreno
----------------	-------	--------	--------------	------------------	---------------------

Descripción	El usuario del sistema podrá editar el nombre y dimensión de sus lotes asociados a sus fincas				
-------------	---	--	--	--	--

Identificación	RF-08	Nombre	Eliminar lotes	Plano conceptual	Gestión del terreno
Descripción	El usuario del sistema podrá eliminar lotes asociados a sus fincas				

Identificación	RF-09	Nombre	Creación bloques	Plano conceptual	Gestión del terreno
Descripción	El usuario del sistema podrá crear bloques asociados a un lote con su respectiva dimensión				

Identificación	RF-10	Nombre	Visualización de bloques	Plano conceptual	Gestión del terreno
Descripción	El usuario del sistema podrá ver el identificador y la dimensión de los bloques asociados a sus lotes				

Identificación	RF-11	Nombre	Editar bloques	Plano conceptual	Gestión del terreno
Descripción	El usuario del sistema podrá editar la dimensión de los bloques asociados a sus lotes				

Identificación	RF-12	Nombre	Eliminar bloques	Plano conceptual	Gestión del terreno
Descripción	El usuario del sistema podrá eliminar bloques asociados a sus lotes				

Identificación	RF-13	Nombre	Creación equipo mecánico	Plano conceptual	Gestión de recursos
Descripción	El usuario del sistema podrá crear equipos mecánicos especificando el nombre y la capacidad en litros del mismo				

Identificación	RF-14	Nombre	Visualización de equipo mecánico	Plano conceptual	Gestión de recursos
Descripción	El usuario del sistema podrá ver el identificador, el nombre y la capacidad en litros de los equipos mecánicos				

Identificación	RF-15	Nombre	Editar equipo mecánico	Plano conceptual	Gestión de recursos
Descripción	El usuario del sistema podrá editar el nombre y la capacidad en litros de los equipos mecánicos				

Identificación	RF-16	Nombre	Eliminar equipo mecánico	Plano conceptual	Gestión de recursos
Descripción	El usuario del sistema podrá eliminar equipos mecánicos				

Identificación	RF-17	Nombre	Creación equipo de protección	Plano conceptual	Gestión de recursos
Descripción	El usuario del sistema podrá crear un equipo de protección especificando su nombre respectivo				

Identificación	RF-18	Nombre	Visualización equipo de protección	Plano conceptual	Gestión de recursos
Descripción	El usuario del sistema podrá ver el identificador y el nombre de los equipos de protección				

Identificación	RF-19	Nombre	Editar equipo de protección	Plano conceptual	Gestión de recursos
Descripción	El usuario del sistema podrá editar el nombre de los equipos de protección				

Identificación	RF-20	Nombre	Eliminar equipo de protección	Plano conceptual	Gestión de recursos
Descripción	El usuario del sistema podrá eliminar equipos de protección				

Identificación	RF-21	Nombre	Creación insumos	Plano conceptual	Gestión de recursos
Descripción	El usuario del sistema podrá crear insumos detallando el nombre comercial, ingrediente activo, periodo de reingreso, intervalo de cosecha, restricciones y dosis recomendada				

Identificación	RF-22	Nombre	Visualización de insumos	Plano conceptual	Gestión de recursos
Descripción	El usuario del sistema podrá ver los siguientes datos de los insumos: identificador, nombre comercial, ingrediente activo, periodo de reingreso, intervalo de cosecha, restricciones y dosis recomendada.				

Identificación	RF-23	Nombre	Editar insumos	Plano conceptual	Gestión de recursos
Descripción	El usuario del sistema podrá editar los siguientes datos de los insumos: nombre comercial, ingrediente activo, periodo de reingreso, intervalo de cosecha, restricciones y dosis recomendada				

Identificación	RF-24	Nombre	Eliminar insumos	Plano conceptual	Gestión de recursos
Descripción	El usuario del sistema podrá eliminar insumos				

Identificación	RF-25	Nombre	Crear cédula	Plano conceptual	Gestión de aplicaciones
Descripción	El usuario del sistema podrá crear cédulas detallando la fecha de siembra, el tipo de semilla utilizada, la densidad de siembra y el identificador del bloque sembrado.				

Identificación	RF-26	Nombre	Crear grupos	Plano conceptual	Gestión de aplicaciones
Descripción	El usuario del sistema podrá crear un grupo que contenga todas aquellas cédulas representantes de bloques que pertenezcan a una misma temporada de cosecha. El grupo deberá de tener los siguientes datos: sumatoria del área de los bloques, estado (activo o desactivado), fecha de creación y fecha de desactivación.				

Identificación	RF-27	Nombre	Ver grupos	Plano conceptual	Gestión de aplicaciones
Descripción	El usuario del sistema podrá ver el estado (activo o inactivo), sumatoria del área de los bloques pertenecientes al grupo, fecha de creación y fecha de desactivación de los grupos de una finca				

Identificación	RF-28	Nombre	Bifurcar grupos	Plano conceptual	Gestión de aplicaciones
Descripción	El usuario del sistema podrá bifurcar un grupo al cambiar el estado del grupo (de activo a desactivado) e insertar la fecha de desactivación. Misma instrucción requiere de la selección de las cédulas para dividir el grupo y poder crear dos nuevos grupos.				

Identificación	RF-29	Nombre	Crear ciclos	Plano conceptual	Gestión de aplicaciones
Descripción	El usuario del sistema podrá crear ciclos detallando los siguientes datos: tipo, fecha de aplicación recomendada, fecha de aplicación, densidad de aplicación, hora de aplicación, tiempo atmosférico a la hora de aplicación, observaciones de aplicación y motivo de aplicación.				

Identificación	RF-30	Nombre	Ver ciclos	Plano conceptual	Gestión de aplicaciones
Descripción	El usuario del sistema podrá ver los ciclos con los siguientes datos: tipo, fecha de aplicación recomendada, fecha de aplicación, densidad				

	de aplicación, hora de aplicación, tiempo atmosférico a la hora de aplicación, observaciones de aplicación y motivo de aplicación				
--	---	--	--	--	--

Identificación	RF-31	Nombre	Editar ciclos	Plano conceptual	Gestión de aplicaciones
Descripción	El usuario del sistema podrá editar los siguientes datos de los ciclos: fecha de aplicación, densidad de aplicación, hora de aplicación, tiempo atmosférico a la hora de aplicación, observaciones de aplicación y motivo de aplicación				

Identificación	RF-32	Nombre	Eliminar ciclos	Plano conceptual	Gestión de aplicaciones
Descripción	El usuario del sistema podrá eliminar ciclos				

Identificación	RF-33	Nombre	Crear receta	Plano conceptual	Gestión de aplicaciones
Descripción	El usuario del sistema podrá crear la receta de insumos a aplicar en un ciclo en específico con la lista de insumos a aplicar y la justificación de la utilización de los mismos.				

Identificación	RF-34	Nombre	Ver receta	Plano conceptual	Gestión de aplicaciones
Descripción	El usuario del sistema podrá ver la lista de insumos a aplicar en el ciclo con su respectivo nombre y la cantidad total de producto a aplicar				

Identificación	RF-35	Nombre	Editar receta	Plano conceptual	Gestión de aplicaciones
Descripción	El usuario del sistema puede agregar o eliminar un insumo a la receta				

Identificación	RF-36	Nombre	Eliminar receta	Plano conceptual	Gestión de aplicaciones
Descripción	El usuario del sistema puede eliminar la receta del ciclo				

Identificación	RF-37	Nombre	Crear equipos mecánicos utilizados en aplicación	Plano conceptual	Gestión de aplicaciones
Descripción	El usuario del sistema podrá crear el equipo mecánico a utilizar en un ciclo de aplicación seleccionando el equipos mecánico desea en la aplicación				

Identificación	RF-38	Nombre	Ver equipos mecánicos utilizados en aplicación	Plano conceptual	Gestión de aplicaciones
Descripción	El usuario del sistema podrá ver el nombre de los equipos de protección que han sido asignados a un ciclo				

Identificación	RF-39	Nombre	Eliminar equipos mecánicos utilizados en aplicación	Plano conceptual	Gestión de aplicaciones
Descripción	El usuario del sistema podrá eliminar el equipo mecánico asignado a un ciclo				

Identificación	RF-40	Nombre	Crear equipos de protección	Plano conceptual	Gestión de aplicaciones
----------------	-------	--------	-----------------------------	------------------	-------------------------

			utilizados en aplicación		
Descripción	El usuario del sistema podrá crear los equipos de protección utilizados en un ciclo de aplicación seleccionando cuales equipos de protección desea en la aplicación				

Identificación	RF-41	Nombre	Ver equipos de protección utilizados en aplicación	Plano conceptual	Gestión de aplicaciones
Descripción	El usuario del sistema podrá ver el nombre de los equipos de protección que han sido asignados a un ciclo				

Identificación	RF-42	Nombre	Editar equipos de protección utilizados en aplicación	Plano conceptual	Gestión de aplicaciones
Descripción	El usuario del sistema podrá eliminar o agregar equipos de protección utilizados en una aplicación				

Identificación	RF-43	Nombre	Eliminar equipos de protección utilizados en aplicación	Plano conceptual	Gestión de aplicaciones
Descripción	El usuario del sistema podrá eliminar la totalidad de equipos de protección utilizados en una aplicación				



## 9.2 Anexo 2. Pruebas Funcionales del Sistema

Nombre de la prueba	Identificador de la prueba	Identificador de requerimiento involucrado	Módulo conceptual
Crear finca	QA-Func-01	RF-01	Gestión del terreno
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>• Tabla de Finca creada en base de datos</li> <li>• Interfaz de la aplicación del programa (API) que permita la consulta de crear Finca</li> <li>• Vista con botón que permita activar la instrucción</li> <li>• Vista con formulario que permita ingresar los datos requeridos para crear finca</li> <li>• Vista con botón que permita enviar formulario</li> <li>• Vista con botón que permita cancelar instrucción</li> </ul>		
<b>Activador</b>	Click en el botón con la leyenda (“crear nueva finca”)		
<b>Descripción</b>	El usuario del sistema realiza click sobre el botón activador de la instrucción. Seguidamente se despliega un formulario que permite al usuario ingresar la información requerida. Una vez el usuario haya rellenado los datos solicitados, al dar click al botón “aceptar”, el elemento se creará en la base de datos.		
<b>Caso</b>	<b>ingresar datos correctamente y dar click sobre botón agregar</b>		
<b>Resultado</b>	Exitoso		
<b>Detalle</b>	Finca creada exitosamente. Nuevo elemento se aprecia en la interfaz gráfica del programa.		
<b>Caso</b>	<b>Ingresar datos correctamente y dar click sobre botón cancelar</b>		
<b>Resultado</b>	Exitoso		
<b>Detalle</b>	El sistema devuelve al usuario a la vista previa antes de presionar el botón “crear nueva finca”		
<b>Caso</b>	<b>Click en enviar sin haber ingresado ningún dato en el formulario</b>		
<b>Resultado</b>	Exitoso		
<b>Detalle</b>	El sistema impide la acción al desactivar el botón puesto que el formulario no contiene valores		
<b>Caso</b>	<b>Click en enviar con un espacio del formulario en blanco</b>		
<b>Resultado</b>	Exitoso		
<b>Detalle</b>	El sistema impide la creación del elemento al desactivar el botón puesto que el formulario contiene al menos un valor en blanco.		
<b>Estado de la prueba</b>	Aprobada		

Nombre de la prueba	Identificador de la prueba	Identificador de requerimiento involucrado	Módulo conceptual
Ver finca	QA-Func-02	RF-02	Gestión del terreno

<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>• Tabla de Finca creada en base de datos</li> <li>• Al menos una fila de Finca creada en la base de datos</li> <li>• Interfaz de la aplicación del programa (API) que permita la consulta de ver Finca</li> <li>• Vista con tabla que permita la renderización de datos</li> </ul>
<b>Activador</b>	Acceso a ruta correspondiente a la estructura
<b>Descripción</b>	Al acceder a la ruta, el sistema renderiza en una tabla algunos de los datos correspondientes a la estructura. A su misma vez dispone de las opciones a realizar sobre cada fila de la tabla.
<b>Caso</b>	<b>Acceder a la ruta</b>
<b>Resultado</b>	Exitoso
<b>Detalle</b>	Datos de la tabla renderizados correctamente. De igual manera los botones con las acciones a realizar sobre cada fila de la tabla.
<b>Estado de la prueba</b>	Aprobada

<b>Nombre de la prueba</b>	<b>Identificador de la prueba</b>	<b>Identificador de requerimiento involucrado</b>	<b>Módulo conceptual</b>
Editar finca	QA-Func-03	RF-03	Gestión del terreno
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>• Tabla de Finca creada en base de datos</li> <li>• Al menos una fila de Finca creada en la base de datos</li> <li>• Interfaz de la aplicación del programa (API) que permita la consulta de editar Finca</li> <li>• Vista con botón que permita activar la instrucción</li> <li>• Vista con formulario que permita ingresar los datos requeridos para editar finca</li> <li>• Vista con botón que permita enviar formulario</li> <li>• Vista con botón que permita cancelar instrucción</li> </ul>		
<b>Activador</b>	Click en el botón con la leyenda (“editar”)		
<b>Descripción</b>	El usuario del sistema realiza click sobre el botón activador de la instrucción. Seguidamente se despliega un formulario que permite al usuario ingresar la información requerida. Una vez el usuario haya editados los datos deseados, al dar click al botón “editar”, el elemento se actualizará en la base de datos.		
<b>Caso</b>	<b>Ingresar datos correctamente y dar click sobre botón agregar</b>		
<b>Resultado</b>	Exitoso		
<b>Detalle</b>	Elemento de finca actualizado exitosamente. Elemento actualizado se aprecia en la interfaz gráfica del programa.		
<b>Caso</b>	<b>Ingresar datos correctamente y dar click sobre botón cancelar</b>		
<b>Resultado</b>	Exitoso		

<b>Detalle</b>	El sistema devuelve al usuario a la vista previa antes de presionar el botón “editar”
<b>Caso</b>	<b>Click en actualizar sin haber cambiado ningún dato en el formulario</b>
<b>Resultado</b>	Exitoso
<b>Detalle</b>	El sistema impide la acción puesto que el formulario no contiene modificaciones y actualización resulta innecesaria
<b>Caso</b>	<b>Click en editar con un espacio del formulario en blanco</b>
<b>Resultado</b>	Exitoso
<b>Detalle</b>	El sistema impide la actualización del elemento puesto que el formulario contiene al menos un valor en blanco. Se le indica al usuario por medio de un mensaje que el espacio en blanco que dejó en el formulario es requerido para la actualización del elemento.
<b>Estado de la prueba</b>	Aprobada

<b>Nombre de la prueba</b>	<b>Identificador de la prueba</b>	<b>Identificador de requerimiento involucrado</b>	<b>Módulo conceptual</b>
Eliminar finca	QA-Func-04	RF-04	Gestión del terreno
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>• Tabla de Finca creada en base de datos</li> <li>• Al menos una fila de Finca creada en la base de datos</li> <li>• Interfaz de la aplicación del programa (API) que permita la consulta de eliminar Finca</li> <li>• Vista con botón que permita activar la instrucción</li> <li>• Vista con botón que permita confirmar la instrucción</li> <li>• Vista con botón que permita cancelar instrucción</li> </ul>		
<b>Activador</b>	Click en el botón con la leyenda (“Eliminar”)		
<b>Descripción</b>	El usuario del sistema realiza click sobre el botón activador de la instrucción. Seguidamente se despliega un mensaje que notifica al usuario de la instrucción a realizar y le pide confirmar su decisión. Una vez el usuario haya confirmado su decisión, el elemento se eliminará en la base de datos.		
<b>Caso</b>	Confirmar decisión		
<b>Resultado</b>	Exitoso		
<b>Detalle</b>	Elemento de finca eliminado exitosamente. Elemento eliminado no se aprecia en la interfaz gráfica del programa.		
<b>Caso</b>	Cancelar decisión		
<b>Resultado</b>	Exitoso		
<b>Detalle</b>	El sistema devuelve al usuario a la vista previa antes de presionar el botón “Eliminar”		

Estado de la prueba	Aprobada
---------------------	----------

Nombre de la prueba	Identificador de la prueba	Identificador de requerimiento involucrado	Módulo conceptual
Crear lote	QA-Func-05	RF-05	Gestión del terreno
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>• Tabla de Lote creada en base de datos</li> <li>• Interfaz de la aplicación del programa (API) que permita la consulta de crear Lote</li> <li>• Vista con botón que permita activar la instrucción</li> <li>• Vista con formulario que permita ingresar los datos requeridos para crear lote</li> <li>• Vista con botón que permita enviar formulario</li> <li>• Vista con botón que permita cancelar instrucción</li> </ul>		
<b>Activador</b>	Click en el botón con la leyenda (“crear nuevo lote”)		
<b>Descripción</b>	El usuario del sistema realiza click sobre el botón activador de la instrucción. Seguidamente se despliega un formulario que permite al usuario ingresar los datos requeridos. Una vez el usuario haya rellenado los datos solicitados, al dar click al botón “aceptar”, el elemento se creará en la base de datos.		
<b>Caso</b>	<b>ingresar datos correctamente y dar click sobre botón agregar</b>		
<b>Resultado</b>	Exitoso		
<b>Detalle</b>	Lote creado exitosamente. Nuevo elemento se aprecia en la interfaz gráfica del programa.		
<b>Caso</b>	<b>Ingresar datos correctamente y dar click sobre botón cancelar</b>		
<b>Resultado</b>	Exitoso		
<b>Detalle</b>	El sistema devuelve al usuario a la vista previa antes de presionar el botón “crear nuevo lote”		
<b>Caso</b>	<b>Click en enviar sin haber ingresado ningún dato en el formulario</b>		
<b>Resultado</b>	Exitoso		
<b>Detalle</b>	El sistema impide la acción al desactivar el botón puesto que el formulario no contiene valores		
<b>Caso</b>	<b>Click en enviar con un espacio del formulario en blanco</b>		
<b>Resultado</b>	Exitoso		
<b>Detalle</b>	El sistema impide la creación del elemento al desactivar el botón puesto que el formulario contiene al menos un valor en blanco.		
Estado de la prueba	Aprobada		

Nombre de la prueba	Identificador de la prueba	Identificador de requerimiento involucrado	Módulo conceptual
Ver lote	QA-Func-06	RF-06	Gestión del terreno
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>• Tabla de Lote creada en base de datos</li> <li>• Al menos una fila de Lote creada en la base de datos</li> <li>• Interfaz de la aplicación del programa (API) que permita la consulta de ver Lote</li> <li>• Vista con tabla que permita la renderización de datos</li> </ul>		
<b>Activador</b>	Click en el botón con la leyenda (“Lotes”)		
<b>Descripción</b>	El usuario del sistema realiza click sobre el botón activador de la instrucción. Seguidamente el sistema renderiza en una tabla algunos de los datos correspondientes a la estructura. A su misma vez dispone de las opciones a realizar sobre cada fila de la tabla.		
<b>Caso</b>	<b>Click en el botón con la leyenda “Lotes”</b>		
<b>Resultado</b>	Exitoso		
<b>Detalle</b>	Datos de la tabla renderizados correctamente. De igual manera los botones con las acciones a realizar sobre cada fila de la tabla.		
<b>Estado de la prueba</b>	Aprobada		

Nombre de la prueba	Identificador de la prueba	Identificador de requerimiento involucrado	Módulo conceptual
Editar lotes	QA-Func-07	RF-07	Gestión del terreno
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>• Tabla de Lotes creada en base de datos</li> <li>• Al menos una fila de Lotes creada en la base de datos</li> <li>• Interfaz de la aplicación del programa (API) que permita la consulta de editar Finca</li> <li>• Vista con botón que permita activar la instrucción</li> <li>• Vista con formulario que permita ingresar los datos requeridos para editar lotes</li> <li>• Vista con botón que permita enviar formulario</li> <li>• Vista con botón que permita cancelar instrucción</li> </ul>		
<b>Activador</b>	Click en el botón con la leyenda (“editar”)		
<b>Descripción</b>	El usuario del sistema realiza click sobre el botón activador de la instrucción. Seguidamente se despliega un formulario que permite al usuario ingresar la información requerida. Una vez el usuario haya editados los datos deseados, al dar click al botón “editar”, el elemento se actualizará en la base de datos.		
<b>Caso</b>	<b>Ingresar datos correctamente y dar click sobre botón agregar</b>		
<b>Resultado</b>	Exitoso		

<b>Detalle</b>	Elemento de lote actualizado exitosamente. Elemento actualizado se aprecia en la interfaz gráfica del programa.
<b>Caso</b>	<b>Ingresar datos correctamente y dar click sobre botón cancelar</b>
<b>Resultado</b>	Exitoso
<b>Detalle</b>	El sistema devuelve al usuario a la vista previa antes de presionar el botón “editar”
<b>Caso</b>	<b>Click en actualizar sin haber cambiado ningún dato en el formulario</b>
<b>Resultado</b>	Exitoso
<b>Detalle</b>	El sistema impide la acción puesto que el formulario no contiene modificaciones y actualización resulta innecesaria
<b>Caso</b>	<b>Click en editar con un espacio del formulario en blanco</b>
<b>Resultado</b>	Exitoso
<b>Detalle</b>	El sistema impide la actualización del elemento puesto que el formulario contiene al menos un valor en blanco. Se le indica al usuario por medio de un mensaje que el espacio en blanco que dejó en el formulario es requerido para la actualización del elemento.
<b>Estado de la prueba</b>	Aprobada

<b>Nombre de la prueba</b>	<b>Identificador de la prueba</b>	<b>Identificador de requerimiento involucrado</b>	<b>Módulo conceptual</b>
Eliminar lote	QA-Func-08	RF-08	Gestión del terreno
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>• Tabla de Lote creada en base de datos</li> <li>• Al menos una fila de Lote creada en la base de datos</li> <li>• Interfaz de la aplicación del programa (API) que permita la consulta de eliminar Finca</li> <li>• Vista con botón que permita activar la instrucción</li> <li>• Vista con botón que permita confirmar la instrucción</li> <li>• Vista con botón que permita cancelar instrucción</li> </ul>		
<b>Activador</b>	Click en el botón con la leyenda (“Eliminar”)		
<b>Descripción</b>	El usuario del sistema realiza click sobre el botón activador de la instrucción. Seguidamente se despliega un mensaje que notifica al usuario de la instrucción a realizar y le pide confirmar su decisión. Una vez el usuario haya confirmado su decisión, el elemento se eliminará en la base de datos.		
<b>Caso</b>	Confirmar decisión		
<b>Resultado</b>	Exitoso		
<b>Detalle</b>	Elemento de lote eliminado exitosamente. Elemento eliminado no se aprecia en la interfaz gráfica del programa.		
<b>Caso</b>	Cancelar decisión		

<b>Resultado</b>	Exitoso
<b>Detalle</b>	El sistema devuelve al usuario a la vista previa antes de presionar el botón “Eliminar”
<b>Estado de la prueba</b>	Aprobada

<b>Nombre de la prueba</b>	<b>Identificador de la prueba</b>	<b>Identificador de requerimiento involucrado</b>	<b>Módulo conceptual</b>
Crear bloque	QA-Func-09	RF-09	Gestión del terreno
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>• Tabla de Bloque creada en base de datos</li> <li>• Interfaz de la aplicación del programa (API) que permita la consulta de crear Bloque</li> <li>• Vista con botón que permita activar la instrucción</li> <li>• Vista con formulario que permita ingresar los datos requeridos para crear bloque</li> <li>• Vista con botón que permita enviar formulario</li> <li>• Vista con botón que permita cancelar instrucción</li> </ul>		
<b>Activador</b>	Click en el botón con la leyenda (“crear nuevo bloque”)		
<b>Descripción</b>	El usuario del sistema realiza click sobre el botón activador de la instrucción. Seguidamente se despliega un formulario que permite al usuario ingresar los datos requeridos. Una vez el usuario haya rellenado los datos solicitados, al dar click al botón “aceptar”, el elemento se creará en la base de datos.		
<b>Caso</b>	<b>ingresar datos correctamente y dar click sobre botón agregar</b>		
<b>Resultado</b>	Exitoso		
<b>Detalle</b>	Bloque creado exitosamente. Nuevo elemento se aprecia en la interfaz gráfica del programa.		
<b>Caso</b>	<b>Ingresar datos correctamente y dar click sobre botón cancelar</b>		
<b>Resultado</b>	Exitoso		
<b>Detalle</b>	El sistema devuelve al usuario a la vista previa antes de presionar el botón “crear nuevo bloque”		
<b>Caso</b>	<b>Click en enviar sin haber ingresado ningún dato en el formulario</b>		
<b>Resultado</b>	Exitoso		
<b>Detalle</b>	El sistema impide la acción al desactivar el botón puesto que el formulario no contiene valores		
<b>Caso</b>	<b>Click en enviar con un espacio del formulario en blanco</b>		
<b>Resultado</b>	Exitoso		
<b>Detalle</b>	El sistema impide la creación del elemento al desactivar el botón puesto que el formulario contiene al menos un valor en blanco.		

<b>Estado de la prueba</b>	Aprobada
----------------------------	----------

<b>Nombre de la prueba</b>	<b>Identificador de la prueba</b>	<b>Identificador de requerimiento involucrado</b>	<b>Módulo conceptual</b>
Ver bloque	QA-Func-10	RF-10	Gestión del terreno
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>• Tabla de Bloque creada en base de datos</li> <li>• Al menos una fila de Bloque creada en la base de datos</li> <li>• Interfaz de la aplicación del programa (API) que permita la consulta de ver Bloque</li> <li>• Vista con tabla que permita la renderización de datos</li> </ul>		
<b>Activador</b>	Click en el botón con la leyenda (“Bloques”)		
<b>Descripción</b>	El usuario del sistema realiza click sobre el botón activador de la instrucción. Seguidamente el sistema renderiza en una tabla algunos de los datos correspondientes a la estructura. A su misma vez dispone de las opciones a realizar sobre cada fila de la tabla.		
<b>Caso</b>	<b>Click en el botón con la leyenda “Bloques”</b>		
<b>Resultado</b>	Exitoso		
<b>Detalle</b>	Datos de la tabla renderizados correctamente. De igual manera los botones con las acciones a realizar sobre cada fila de la tabla.		
<b>Estado de la prueba</b>	Aprobada		

<b>Nombre de la prueba</b>	<b>Identificador de la prueba</b>	<b>Identificador de requerimiento involucrado</b>	<b>Módulo conceptual</b>
Editar bloques	QA-Func-11	RF-11	Gestión del terreno
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>• Tabla de Bloque creada en base de datos</li> <li>• Al menos una fila de Bloque creada en la base de datos</li> <li>• Interfaz de la aplicación del programa (API) que permita la consulta de editar Bloque</li> <li>• Vista con botón que permita activar la instrucción</li> <li>• Vista con formulario que permita ingresar los datos requeridos para editar bloques</li> <li>• Vista con botón que permita enviar formulario</li> <li>• Vista con botón que permita cancelar instrucción</li> </ul>		
<b>Activador</b>	Click en el botón con la leyenda (“editar”)		
<b>Descripción</b>	El usuario del sistema realiza click sobre el botón activador de la instrucción. Seguidamente se despliega un formulario que permite al usuario ingresar la información requerida. Una vez el usuario haya		

	editados los datos deseados, al dar click al botón “editar”, el elemento se actualizará en la base de datos.
<b>Caso</b>	<b>Ingresar datos correctamente y dar click sobre botón agregar</b>
<b>Resultado</b>	Exitoso
<b>Detalle</b>	Elemento de bloque actualizado exitosamente. Elemento actualizado se aprecia en la interfaz gráfica del programa.
<b>Caso</b>	<b>Ingresar datos correctamente y dar click sobre botón cancelar</b>
<b>Resultado</b>	Exitoso
<b>Detalle</b>	El sistema devuelve al usuario a la vista previa antes de presionar el botón “editar”
<b>Caso</b>	<b>Click en actualizar sin haber cambiado ningún dato en el formulario</b>
<b>Resultado</b>	Exitoso
<b>Detalle</b>	El sistema impide la acción puesto que el formulario no contiene modificaciones y actualización resulta innecesaria
<b>Caso</b>	<b>Click en editar con un espacio del formulario en blanco</b>
<b>Resultado</b>	Exitoso
<b>Detalle</b>	El sistema impide la actualización del elemento puesto que el formulario contiene al menos un valor en blanco. Se le indica al usuario por medio de un mensaje que el espacio en blanco que dejó en el formulario es requerido para la actualización del elemento.
<b>Estado de la prueba</b>	Aprobada

<b>Nombre de la prueba</b>	<b>Identificador de la prueba</b>	<b>Identificador de requerimiento involucrado</b>	<b>Módulo conceptual</b>
Eliminar bloque	QA-Func-12	RF-12	Gestión del terreno
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>• Tabla de Bloque creada en base de datos</li> <li>• Al menos una fila de Bloque creada en la base de datos</li> <li>• Interfaz de la aplicación del programa (API) que permita la consulta de eliminar Bloque</li> <li>• Vista con botón que permita activar la instrucción</li> <li>• Vista con botón que permita confirmar la instrucción</li> <li>• Vista con botón que permita cancelar instrucción</li> </ul>		
<b>Activador</b>	Click en el botón con la leyenda (“Eliminar”)		
<b>Descripción</b>	El usuario del sistema realiza click sobre el botón activador de la instrucción. Seguidamente se despliega un mensaje que notifica al usuario de la instrucción a realizar y le pide confirmar su decisión. Una vez el usuario haya confirmado su decisión, el elemento se eliminará en la base de datos.		
<b>Caso</b>	Confirmar decisión		

<b>Resultado</b>	Exitoso
<b>Detalle</b>	Elemento de bloque eliminado exitosamente. Elemento eliminado no se aprecia en la interfaz gráfica del programa.
<b>Caso</b>	Cancelar decisión
<b>Resultado</b>	Exitoso
<b>Detalle</b>	El sistema devuelve al usuario a la vista previa antes de presionar el botón “Eliminar”
<b>Estado de la prueba</b>	Aprobada

<b>Nombre de la prueba</b>	<b>Identificador de la prueba</b>	<b>Identificador de requerimiento involucrado</b>	<b>Módulo conceptual</b>
Crear equipo mecánico	QA-Func-13	RF-13	Gestión de recursos
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>• Tabla de Equipo mecánico creada en base de datos</li> <li>• Interfaz de la aplicación del programa (API) que permita la consulta de crear Equipo mecánico</li> <li>• Vista con botón que permita activar la instrucción</li> <li>• Vista con formulario que permita ingresar los datos requeridos para crear Equipo mecánico</li> <li>• Vista con botón que permita enviar formulario</li> <li>• Vista con botón que permita cancelar instrucción</li> </ul>		
<b>Activador</b>	Click en el botón con la leyenda (“crear nuevo equipo mecánico”)		
<b>Descripción</b>	El usuario del sistema realiza click sobre el botón activador de la instrucción. Seguidamente se despliega un formulario que permite al usuario ingresar los datos requeridos. Una vez el usuario haya rellenado los datos solicitados, al dar click al botón “aceptar”, el elemento se creará en la base de datos.		
<b>Caso</b>	<b>ingresar datos correctamente y dar click sobre botón agregar</b>		
<b>Resultado</b>	Exitoso		
<b>Detalle</b>	Equipo mecánico creado exitosamente. Nuevo elemento se aprecia en la interfaz gráfica del programa.		
<b>Caso</b>	<b>Ingresar datos correctamente y dar click sobre botón cancelar</b>		
<b>Resultado</b>	Exitoso		
<b>Detalle</b>	El sistema devuelve al usuario a la vista previa antes de presionar el botón “crear nuevo equipo mecánico”		
<b>Caso</b>	<b>Click en enviar sin haber ingresado ningún dato en el formulario</b>		
<b>Resultado</b>	Exitoso		
<b>Detalle</b>	El sistema impide la acción al desactivar el botón puesto que el formulario no contiene valores		

<b>Caso</b>	<b>Click en enviar con un espacio del formulario en blanco</b>
<b>Resultado</b>	Exitoso
<b>Detalle</b>	El sistema impide la creación del elemento al desactivar el botón puesto que el formulario contiene al menos un valor en blanco.
<b>Estado de la prueba</b>	Aprobada

<b>Nombre de la prueba</b>	<b>Identificador de la prueba</b>	<b>Identificador de requerimiento involucrado</b>	<b>Módulo conceptual</b>
Ver equipo mecánico	QA-Func-14	RF-14	Gestión de recursos
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>• Tabla de Equipo mecánico creada en base de datos</li> <li>• Al menos una fila de Equipo mecánico creada en la base de datos</li> <li>• Interfaz de la aplicación del programa (API) que permita la consulta de ver Equipo mecánico</li> <li>• Vista con tabla que permita la renderización de datos</li> </ul>		
<b>Activador</b>	Click en el link con la leyenda (“Equipo mecánico”)		
<b>Descripción</b>	El usuario del sistema realiza click sobre el lick activador de la instrucción. Seguidamente el sistema renderiza en una tabla algunos de los datos correspondientes a la estructura. A su misma vez dispone de las opciones a realizar sobre cada fila de la tabla.		
<b>Caso</b>	<b>Click en el link con la leyenda “Equipos mecánico”</b>		
<b>Resultado</b>	Exitoso		
<b>Detalle</b>	Datos de la tabla renderizados correctamente. De igual manera los botones con las acciones a realizar sobre cada fila de la tabla.		
<b>Estado de la prueba</b>	Aprobada		

<b>Nombre de la prueba</b>	<b>Identificador de la prueba</b>	<b>Identificador de requerimiento involucrado</b>	<b>Módulo conceptual</b>
Editar equipo de protección	QA-Func-15	RF-15	Gestión de recursos
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>• Tabla de Equipo mecánico creada en base de datos</li> <li>• Al menos una fila de Equipo mecánico creada en la base de datos</li> <li>• Interfaz de la aplicación del programa (API) que permita la consulta de editar Equipo mecánico</li> <li>• Vista con botón que permita activar la instrucción</li> <li>• Vista con formulario que permita ingresar los datos requeridos para editar Equipo mecánico</li> <li>• Vista con botón que permita enviar formulario</li> </ul>		

	<ul style="list-style-type: none"> <li>Vista con botón que permita cancelar instrucción</li> </ul>
<b>Activador</b>	Click en el botón con la leyenda (“editar”)
<b>Descripción</b>	El usuario del sistema realiza click sobre el botón activador de la instrucción. Seguidamente se despliega un formulario que permite al usuario ingresar la información requerida. Una vez el usuario haya editados los datos deseados, al dar click al botón “editar”, el elemento se actualizará en la base de datos.
<b>Caso</b>	<b>Ingresar datos correctamente y dar click sobre botón agregar</b>
<b>Resultado</b>	Exitoso
<b>Detalle</b>	Elemento de Equipo mecánico actualizado exitosamente. Elemento actualizado se aprecia en la interfaz gráfica del programa.
<b>Caso</b>	<b>Ingresar datos correctamente y dar click sobre botón cancelar</b>
<b>Resultado</b>	Exitoso
<b>Detalle</b>	El sistema devuelve al usuario a la vista previa antes de presionar el botón “editar”
<b>Caso</b>	<b>Click en actualizar sin haber cambiado ningún dato en el formulario</b>
<b>Resultado</b>	Exitoso
<b>Detalle</b>	El sistema impide la acción puesto que el formulario no contiene modificaciones y actualización resulta innecesaria
<b>Caso</b>	<b>Click en editar con un espacio del formulario en blanco</b>
<b>Resultado</b>	Exitoso
<b>Detalle</b>	El sistema impide la actualización del elemento puesto que el formulario contiene al menos un valor en blanco. Se le indica al usuario por medio de un mensaje que el espacio en blanco que dejó en el formulario es requerido para la actualización del elemento.
<b>Estado de la prueba</b>	Aprobada

<b>Nombre de la prueba</b>	<b>Identificador de la prueba</b>	<b>Identificador de requerimiento involucrado</b>	<b>Módulo conceptual</b>
Eliminar equipo de protección	QA-Func-16	RF-16	Gestión de recursos
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>Tabla de Equipo mecánico creada en base de datos</li> <li>Al menos una fila de Equipo mecánico creada en la base de datos</li> <li>Interfaz de la aplicación del programa (API) que permita la consulta de eliminar Equipo mecánico</li> <li>Vista con botón que permita activar la instrucción</li> <li>Vista con botón que permita confirmar la instrucción</li> <li>Vista con botón que permita cancelar instrucción</li> </ul>		

<b>Activador</b>	Click en el botón con la leyenda (“Eliminar”)
<b>Descripción</b>	El usuario del sistema realiza click sobre el botón activador de la instrucción. Seguidamente se despliega un mensaje que notifica al usuario de la instrucción a realizar y le pide confirmar su decisión. Una vez el usuario haya confirmado su decisión, el elemento se eliminará en la base de datos.
<b>Caso</b>	Confirmar decisión
<b>Resultado</b>	Exitoso
<b>Detalle</b>	Elemento de Equipo mecánico eliminado exitosamente. Elemento eliminado no se aprecia en la interfaz gráfica del programa.
<b>Caso</b>	Cancelar decisión
<b>Resultado</b>	Exitoso
<b>Detalle</b>	El sistema devuelve al usuario a la vista previa antes de presionar el botón “Eliminar”
<b>Estado de la prueba</b>	Aprobada

<b>Nombre de la prueba</b>	<b>Identificador de la prueba</b>	<b>Identificador de requerimiento involucrado</b>	<b>Módulo conceptual</b>
Crear equipo de protección	QA-Func-17	RF-17	Gestión de recursos
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>• Tabla de Equipo de protección creada en base de datos</li> <li>• Interfaz de la aplicación del programa (API) que permita la consulta de crear Equipo de protección</li> <li>• Vista con botón que permita activar la instrucción</li> <li>• Vista con formulario que permita ingresar los datos requeridos para crear Equipo de protección</li> <li>• Vista con botón que permita enviar formulario</li> <li>• Vista con botón que permita cancelar instrucción</li> </ul>		
<b>Activador</b>	Click en el botón con la leyenda (“crear nuevo equipo de protección”)		
<b>Descripción</b>	El usuario del sistema realiza click sobre el botón activador de la instrucción. Seguidamente se despliega un formulario que permite al usuario ingresar los datos requeridos. Una vez el usuario haya rellenado los datos solicitados, al dar click al botón “aceptar”, el elemento se creará en la base de datos.		
<b>Caso</b>	<b>ingresar datos correctamente y dar click sobre botón agregar</b>		
<b>Resultado</b>	Exitoso		
<b>Detalle</b>	Equipo de protección creado exitosamente. Nuevo elemento se aprecia en la interfaz gráfica del programa.		
<b>Caso</b>	<b>Ingresar datos correctamente y dar click sobre botón cancelar</b>		

<b>Resultado</b>	Exitoso
<b>Detalle</b>	El sistema devuelve al usuario a la vista previa antes de presionar el botón “crear nuevo equipo de protección”
<b>Caso</b>	<b>Click en enviar sin haber ingresado ningún dato en el formulario</b>
<b>Resultado</b>	Exitoso
<b>Detalle</b>	El sistema impide la acción al desactivar el botón puesto que el formulario no contiene valores
<b>Caso</b>	<b>Click en enviar con un espacio del formulario en blanco</b>
<b>Resultado</b>	Exitoso
<b>Detalle</b>	El sistema impide la creación del elemento al desactivar el botón puesto que el formulario contiene al menos un valor en blanco.
<b>Estado de la prueba</b>	Aprobada

Nombre de la prueba	Identificador de la prueba	Identificador de requerimiento involucrado	Módulo conceptual
Ver equipo de protección	QA-Func-18	RF-18	Gestión de recursos
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>• Tabla de Equipo de protección creada en base de datos</li> <li>• Al menos una fila de Equipo de protección creada en la base de datos</li> <li>• Interfaz de la aplicación del programa (API) que permita la consulta de ver Equipo de protección</li> <li>• Vista con tabla que permita la renderización de datos</li> </ul>		
<b>Activador</b>	Click en el link con la leyenda (“Equipos de protección”)		
<b>Descripción</b>	El usuario del sistema realiza click sobre el link activador de la instrucción. Seguidamente el sistema renderiza en una tabla algunos de los datos correspondientes a la estructura. A su misma vez dispone de las opciones a realizar sobre cada fila de la tabla.		
<b>Caso</b>	<b>Click en el link con la leyenda “Equipos de protección”</b>		
<b>Resultado</b>	Exitoso		
<b>Detalle</b>	Datos de la tabla renderizados correctamente. De igual manera los botones con las acciones a realizar sobre cada fila de la tabla.		
<b>Estado de la prueba</b>	Aprobada		

Nombre de la prueba	Identificador de la prueba	Identificador de requerimiento involucrado	Módulo conceptual
Editar equipo de protección	QA-Func-19	RF-19	Gestión de recursos
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>• Tabla de Equipo de protección creada en base de datos</li> </ul>		

	<ul style="list-style-type: none"> <li>• Al menos una fila de Equipo de protección creada en la base de datos</li> <li>• Interfaz de la aplicación del programa (API) que permita la consulta de editar Equipo de protección</li> <li>• Vista con botón que permita activar la instrucción</li> <li>• Vista con formulario que permita ingresar los datos requeridos para editar Equipo de protección</li> <li>• Vista con botón que permita enviar formulario</li> <li>• Vista con botón que permita cancelar instrucción</li> </ul>
<b>Activador</b>	Click en el botón con la leyenda (“editar”)
<b>Descripción</b>	El usuario del sistema realiza click sobre el botón activador de la instrucción. Seguidamente se despliega un formulario que permite al usuario ingresar la información requerida. Una vez el usuario haya editados los datos deseados, al dar click al botón “editar”, el elemento se actualizará en la base de datos.
<b>Caso</b>	<b>Ingresar datos correctamente y dar click sobre botón agregar</b>
<b>Resultado</b>	Exitoso
<b>Detalle</b>	Elemento de Equipo de protección actualizado exitosamente. Elemento actualizado se aprecia en la interfaz gráfica del programa.
<b>Caso</b>	<b>Ingresar datos correctamente y dar click sobre botón cancelar</b>
<b>Resultado</b>	Exitoso
<b>Detalle</b>	El sistema devuelve al usuario a la vista previa antes de presionar el botón “editar”
<b>Caso</b>	<b>Click en actualizar sin haber cambiado ningún dato en el formulario</b>
<b>Resultado</b>	Exitoso
<b>Detalle</b>	El sistema impide la acción puesto que el formulario no contiene modificaciones y actualización resulta innecesaria
<b>Caso</b>	<b>Click en editar con un espacio del formulario en blanco</b>
<b>Resultado</b>	Exitoso
<b>Detalle</b>	El sistema impide la actualización del elemento puesto que el formulario contiene al menos un valor en blanco. Se le indica al usuario por medio de un mensaje que el espacio en blanco que dejó en el formulario es requerido para la actualización del elemento.
<b>Estado de la prueba</b>	Aprobada

Nombre de la prueba	Identificador de la prueba	Identificador de requerimiento involucrado	Módulo conceptual
Eliminar equipo de protección	QA-Func-20	RF-20	Gestión de recursos

<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>• Tabla de Equipo de protección creada en base de datos</li> <li>• Al menos una fila de Equipo de protección creada en la base de datos</li> <li>• Interfaz de la aplicación del programa (API) que permita la consulta de eliminar Equipo de protección</li> <li>• Vista con botón que permita activar la instrucción</li> <li>• Vista con botón que permita confirmar la instrucción</li> <li>• Vista con botón que permita cancelar instrucción</li> </ul>
<b>Activador</b>	Click en el botón con la leyenda (“Eliminar”)
<b>Descripción</b>	El usuario del sistema realiza click sobre el botón activador de la instrucción. Seguidamente se despliega un mensaje que notifica al usuario de la instrucción a realizar y le pide confirmar su decisión. Una vez el usuario haya confirmado su decisión, el elemento se eliminará en la base de datos.
<b>Caso</b>	Confirmar decisión
<b>Resultado</b>	Exitoso
<b>Detalle</b>	Elemento de Equipo de protección eliminado exitosamente. Elemento eliminado no se aprecia en la interfaz gráfica del programa.
<b>Caso</b>	Cancelar decisión
<b>Resultado</b>	Exitoso
<b>Detalle</b>	El sistema devuelve al usuario a la vista previa antes de presionar el botón “Eliminar”
<b>Estado de la prueba</b>	Aprobada

#### Insumos

<b>Nombre de la prueba</b>	<b>Identificador de la prueba</b>	<b>Identificador de requerimiento involucrado</b>	<b>Módulo conceptual</b>
Crear Insumos	QA-Func-21	RF-21	Gestión de recursos
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>• Tabla de Insumo creada en base de datos</li> <li>• Interfaz de la aplicación del programa (API) que permita la consulta de crear Insumo</li> <li>• Vista con botón que permita activar la instrucción</li> <li>• Vista con formulario que permita ingresar los datos requeridos para crear Insumo</li> <li>• Vista con botón que permita enviar formulario</li> <li>• Vista con botón que permita cancelar instrucción</li> </ul>		
<b>Activador</b>	Click en el botón con la leyenda (“crear nuevo insumo”)		
<b>Descripción</b>	El usuario del sistema realiza click sobre el botón activador de la instrucción. Seguidamente se despliega un formulario que permite al usuario ingresar los datos requeridos. Una vez el usuario haya		

	rellenado los datos solicitados, al dar click al botón “aceptar”, el elemento se creará en la base de datos.
<b>Caso</b>	<b>ingresar datos correctamente y dar click sobre botón agregar</b>
<b>Resultado</b>	Exitoso
<b>Detalle</b>	Insumo creado exitosamente. Nuevo elemento se aprecia en la interfaz gráfica del programa.
<b>Caso</b>	<b>Ingresar datos correctamente y dar click sobre botón cancelar</b>
<b>Resultado</b>	Exitoso
<b>Detalle</b>	El sistema devuelve al usuario a la vista previa antes de presionar el botón “crear nuevo insumo”
<b>Caso</b>	<b>Click en enviar sin haber ingresado ningún dato en el formulario</b>
<b>Resultado</b>	Exitoso
<b>Detalle</b>	El sistema impide la acción al desactivar el botón puesto que el formulario no contiene valores
<b>Caso</b>	<b>Click en enviar con un espacio del formulario en blanco</b>
<b>Resultado</b>	Exitoso
<b>Detalle</b>	El sistema impide la creación del elemento al desactivar el botón puesto que el formulario contiene al menos un valor en blanco.
<b>Estado de la prueba</b>	Aprobada

Nombre de la prueba	Identificador de la prueba	Identificador de requerimiento involucrado	Módulo conceptual
Ver Insumo	QA-Func-22	RF-22	Gestión de recursos
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>• Tabla de Insumo creada en base de datos</li> <li>• Al menos una fila de Insumo creada en la base de datos</li> <li>• Interfaz de la aplicación del programa (API) que permita la consulta de ver Insumo</li> <li>• Vista con tabla que permita la renderización de datos</li> </ul>		
<b>Activador</b>	Click en el link con la leyenda (“Insumo”)		
<b>Descripción</b>	El usuario del sistema realiza click sobre el lick activador de la instrucción. Seguidamente el sistema renderiza en una tabla algunos de los datos correspondientes a la estructura. A su misma vez dispone de las opciones a realizar sobre cada fila de la tabla.		
<b>Caso</b>	<b>Click en el link con la leyenda “Insumo”</b>		
<b>Resultado</b>	Exitoso		
<b>Detalle</b>	Datos de la tabla renderizados correctamente. De igual manera los botones con las acciones a realizar sobre cada fila de la tabla.		
<b>Estado de la prueba</b>	Aprobada		

Nombre de la prueba	Identificador de la prueba	Identificador de requerimiento involucrado	Módulo conceptual
Editar Insumo	QA-Func-23	RF-23	Gestión de recursos
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>• Tabla de Insumo creada en base de datos</li> <li>• Al menos una fila Insumo creada en la base de datos</li> <li>• Interfaz de la aplicación del programa (API) que permita la consulta de editar Insumo</li> <li>• Vista con botón que permita activar la instrucción</li> <li>• Vista con formulario que permita ingresar los datos requeridos para editar Insumo</li> <li>• Vista con botón que permita enviar formulario</li> <li>• Vista con botón que permita cancelar instrucción</li> </ul>		
<b>Activador</b>	Click en el botón con la leyenda (“editar”)		
<b>Descripción</b>	El usuario del sistema realiza click sobre el botón activador de la instrucción. Seguidamente se despliega un formulario que permite al usuario ingresar la información requerida. Una vez el usuario haya editados los datos deseados, al dar click al botón “editar”, el elemento se actualizará en la base de datos.		
<b>Caso</b>	<b>Ingresar datos correctamente y dar click sobre botón agregar</b>		
<b>Resultado</b>	Exitoso		
<b>Detalle</b>	Elemento de Insumo actualizado exitosamente. Elemento actualizado se aprecia en la interfaz gráfica del programa.		
<b>Caso</b>	<b>Ingresar datos correctamente y dar click sobre botón cancelar</b>		
<b>Resultado</b>	Exitoso		
<b>Detalle</b>	El sistema devuelve al usuario a la vista previa antes de presionar el botón “editar”		
<b>Caso</b>	<b>Click en actualizar sin haber cambiado ningún dato en el formulario</b>		
<b>Resultado</b>	Exitoso		
<b>Detalle</b>	El sistema impide la acción puesto que el formulario no contiene modificaciones y actualización resulta innecesaria		
<b>Caso</b>	<b>Click en editar con un espacio del formulario en blanco</b>		
<b>Resultado</b>	Exitoso		
<b>Detalle</b>	El sistema impide la actualización del elemento puesto que el formulario contiene al menos un valor en blanco. Se le indica al usuario por medio de un mensaje que el espacio en blanco que dejó en el formulario es requerido para la actualización del elemento.		
<b>Estado de la prueba</b>	Aprobada		

Nombre de la prueba	Identificador de la prueba	Identificador de requerimiento involucrado	Módulo conceptual
Eliminar Insumo	QA-Func-24	RF-24	Gestión de recursos
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>• Tabla de Insumo creada en base de datos</li> <li>• Al menos una fila de Insumo creada en la base de datos</li> <li>• Interfaz de la aplicación del programa (API) que permita la consulta de eliminar Insumo</li> <li>• Vista con botón que permita activar la instrucción</li> <li>• Vista con botón que permita confirmar la instrucción</li> <li>• Vista con botón que permita cancelar instrucción</li> </ul>		
<b>Activador</b>	Click en el botón con la leyenda (“Eliminar”)		
<b>Descripción</b>	El usuario del sistema realiza click sobre el botón activador de la instrucción. Seguidamente se despliega un mensaje que notifica al usuario de la instrucción a realizar y le pide confirmar su decisión. Una vez el usuario haya confirmado su decisión, el elemento se eliminará en la base de datos.		
<b>Caso</b>	Confirmar decisión		
<b>Resultado</b>	Exitoso		
<b>Detalle</b>	Elemento de Insumo eliminado exitosamente. Elemento eliminado no se aprecia en la interfaz gráfica del programa.		
<b>Caso</b>	Cancelar decisión		
<b>Resultado</b>	Exitoso		
<b>Detalle</b>	El sistema devuelve al usuario a la vista previa antes de presionar el botón “Eliminar”		
<b>Estado de la prueba</b>	Aprobada		

Nombre de la prueba	Identificador de la prueba	Identificador de requerimiento involucrado	Módulo conceptual
Crear Receta	QA-Func-33	RF-33	Gestión de aplicaciones
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>• Tabla del receta creada en base de datos</li> <li>• Interfaz de la aplicación del programa (API) que permita la consulta de crear receta</li> <li>• Vista con botón que permita activar la instrucción</li> <li>• Vista con formulario que permita seleccionar los datos requeridos para crear receta</li> <li>• Vista con botón que permita enviar formulario</li> <li>• Vista con botón que permita cancelar instrucción</li> </ul>		

<b>Activador</b>	Click en el botón con la leyenda (“crear”)
<b>Descripción</b>	El usuario del sistema realiza click sobre el botón activador de la instrucción. Seguidamente se despliega un formulario que permite al usuario seleccionar mediante un checkbox las opciones deseadas. Una vez el usuario haya seleccionado al menos una opción, al dar click al botón “aceptar”, el elemento se creará en la base de datos.
<b>Caso</b>	<b>Selección de datos correctamente y dar click sobre botón agregar</b>
<b>Resultado</b>	Exitoso
<b>Detalle</b>	Receta creada exitosamente.
<b>Caso</b>	<b>Selección de datos correctamente y dar click sobre botón cancelar</b>
<b>Resultado</b>	Exitoso
<b>Detalle</b>	El sistema devuelve al usuario a la vista previa antes de presionar el botón “crear”
<b>Caso</b>	<b>Click en enviar sin haber seleccionado ningún dato en el formulario</b>
<b>Resultado</b>	Exitoso
<b>Detalle</b>	El sistema impide la acción al desactivar el botón puesto que el formulario no contiene valores
<b>Estado de la prueba</b>	Aprobada

<b>Nombre de la prueba</b>	<b>Identificador de la prueba</b>	<b>Identificador de requerimiento involucrado</b>	<b>Módulo conceptual</b>
Ver receta	QA-Func-34	RF-34	Gestión de aplicaciones
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>• Tabla de receta creada en base de datos</li> <li>• Al menos una fila de receta creada en la base de datos</li> <li>• Interfaz de la aplicación del programa (API) que permita la consulta de ver receta</li> <li>• Vista con tabla que permita la renderización de datos</li> </ul>		
<b>Activador</b>	Click en el link con la leyenda (“receta”)		
<b>Descripción</b>	El usuario del sistema realiza click sobre el link activador de la instrucción. Seguidamente el sistema renderiza en una tabla algunos de los datos correspondientes a la estructura.		
<b>Caso</b>	<b>Click en el link con la leyenda “receta”</b>		
<b>Resultado</b>	Exitoso		
<b>Detalle</b>	Datos de la tabla renderizados correctamente.		
<b>Estado de la prueba</b>	Aprobada		

Nombre de la prueba	Identificador de la prueba	Identificador de requerimiento involucrado	Módulo conceptual
Editar receta	QA-Func-35	RF-35	Gestión de aplicaciones
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>• Tabla de receta creada en base de datos</li> <li>• Al menos una fila equipo de receta creada en la base de datos</li> <li>• Interfaz de la aplicación del programa (API) que permita la consulta de editar receta</li> <li>• Vista con botón que permita activar la instrucción</li> <li>• Vista con formulario que permita seleccionar los datos requeridos para editar receta</li> <li>• Vista con botón que permita enviar formulario</li> <li>• Vista con botón que permita cancelar instrucción</li> </ul>		
<b>Activador</b>	Click en el botón con la leyenda (“editar”)		
<b>Descripción</b>	El usuario del sistema realiza click sobre el botón activador de la instrucción. Seguidamente se despliega un formulario que permite al usuario seleccionar la información requerida. Una vez el usuario haya editados los datos deseados, al dar click al botón “editar”, el elemento se actualizará en la base de datos.		
<b>Caso</b>	<b>Seleccionar datos correctamente y dar click sobre botón agregar</b>		
<b>Resultado</b>	Exitoso		
<b>Detalle</b>	Elemento de receta actualizado exitosamente.		
<b>Caso</b>	<b>Seleccionar datos correctamente y dar click sobre botón cancelar</b>		
<b>Resultado</b>	Exitoso		
<b>Detalle</b>	El sistema devuelve al usuario a la vista previa antes de presionar el botón “editar”		
<b>Caso</b>	<b>Click en actualizar sin haber cambiado ningún dato en el formulario</b>		
<b>Resultado</b>	Exitoso		
<b>Detalle</b>	El sistema impide la acción puesto que el formulario no contiene modificaciones y actualización resulta innecesaria		
<b>Estado de la prueba</b>	Aprobada		

Nombre de la prueba	Identificador de la prueba	Identificador de requerimiento involucrado	Módulo conceptual
Eliminar receta	QA-Func-36	RF-36	Gestión de aplicaciones
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>• Tabla de receta creada en base de datos</li> <li>• Al menos una fila de receta creada en la base de datos</li> <li>• Interfaz de la aplicación del programa (API) que permita la consulta de eliminar receta</li> </ul>		

	<ul style="list-style-type: none"> <li>• Vista con botón que permita activar la instrucción</li> <li>• Vista con botón que permita confirmar la instrucción</li> <li>• Vista con botón que permita cancelar instrucción</li> </ul>
<b>Activador</b>	Click en el botón con la leyenda (“Eliminar”)
<b>Descripción</b>	El usuario del sistema realiza click sobre el botón activador de la instrucción. Seguidamente se despliega un mensaje que notifica al usuario de la instrucción a realizar y le pide confirmar su decisión. Una vez el usuario haya confirmado su decisión, el elemento se eliminará en la base de datos.
<b>Caso</b>	Confirmar decisión
<b>Resultado</b>	Exitoso
<b>Detalle</b>	Elemento del receta eliminado exitosamente. Elemento eliminado no se aprecia en la interfaz gráfica del programa.
<b>Caso</b>	Cancelar decisión
<b>Resultado</b>	Exitoso
<b>Detalle</b>	El sistema devuelve al usuario a la vista previa antes de presionar el botón “Eliminar”
<b>Estado de la prueba</b>	Aprobada

<b>Nombre de la prueba</b>	<b>Identificador de la prueba</b>	<b>Identificador de requerimiento involucrado</b>	<b>Módulo conceptual</b>
Crear equipo mecánico utilizado en aplicación	QA-Func-37	RF-37	Gestión de aplicaciones
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>• Tabla del equipo mecánico utilizado en aplicación creada en base de datos</li> <li>• Interfaz de la aplicación del programa (API) que permita la consulta de crear equipo mecánico utilizado en aplicación</li> <li>• Vista con botón que permita activar la instrucción</li> <li>• Vista con formulario que permita seleccionar los datos requeridos para crear equipo mecánico utilizado en aplicación</li> <li>• Vista con botón que permita enviar formulario</li> <li>• Vista con botón que permita cancelar instrucción</li> </ul>		
<b>Activador</b>	Click en el botón con la leyenda (“crear”)		
<b>Descripción</b>	El usuario del sistema realiza click sobre el botón activador de la instrucción. Seguidamente se despliega un formulario que permite al usuario seleccionar mediante un checkbox las opciones deseadas. Una vez el usuario haya seleccionado al menos una opción, al dar click al botón “aceptar”, el elemento se creará en la base de datos.		

<b>Caso</b>	<b>Selección de datos correctamente y dar click sobre botón agregar</b>
<b>Resultado</b>	Exitoso
<b>Detalle</b>	Equipo mecánico utilizado en aplicación creado exitosamente.
<b>Caso</b>	<b>Selección de datos correctamente y dar click sobre botón cancelar</b>
<b>Resultado</b>	Exitoso
<b>Detalle</b>	El sistema devuelve al usuario a la vista previa antes de presionar el botón "crear"
<b>Caso</b>	<b>Click en enviar sin haber seleccionado ningún dato en el formulario</b>
<b>Resultado</b>	Exitoso
<b>Detalle</b>	El sistema impide la acción al desactivar el botón puesto que el formulario no contiene valores
<b>Estado de la prueba</b>	Aprobada

<b>Nombre de la prueba</b>	<b>Identificador de la prueba</b>	<b>Identificador de requerimiento involucrado</b>	<b>Módulo conceptual</b>
Ver equipo mecánico utilizado en aplicación	QA-Func-38	RF-38	Gestión de aplicaciones
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>• Tabla de equipo mecánico utilizado en aplicación creada en base de datos</li> <li>• Al menos una fila de equipo mecánico utilizado en aplicación creada en la base de datos</li> <li>• Interfaz de la aplicación del programa (API) que permita la consulta de ver equipo mecánico utilizado en aplicación</li> <li>• Vista con tabla que permita la renderización de datos</li> </ul>		
<b>Activador</b>	Click en el link con la leyenda ("equipo mecánico utilizado en aplicación")		
<b>Descripción</b>	El usuario del sistema realiza click sobre el link activador de la instrucción. Seguidamente el sistema renderiza en una tabla algunos de los datos correspondientes a la estructura		
<b>Caso</b>	<b>Click en el link con la leyenda "equipo mecánico utilizado en aplicación"</b>		
<b>Resultado</b>	Exitoso		
<b>Detalle</b>	Datos de la tabla renderizados correctamente.		
<b>Estado de la prueba</b>	Aprobada		

Nombre de la prueba	Identificador de la prueba	Identificador de requerimiento involucrado	Módulo conceptual
Eliminar equipo mecánico utilizado en aplicación	QA-Func-39	RF-39	Gestión de aplicaciones
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>• Tabla de equipo mecánico utilizado en aplicación creada en base de datos</li> <li>• Al menos una fila de equipo mecánico utilizado en aplicación creada en la base de datos</li> <li>• Interfaz de la aplicación del programa (API) que permita la consulta de eliminar Insumo</li> <li>• Vista con botón que permita activar la instrucción</li> <li>• Vista con botón que permita confirmar la instrucción</li> <li>• Vista con botón que permita cancelar instrucción</li> </ul>		
<b>Activador</b>	Click en el botón con la leyenda (“Eliminar”)		
<b>Descripción</b>	El usuario del sistema realiza click sobre el botón activador de la instrucción. Seguidamente se despliega un mensaje que notifica al usuario de la instrucción a realizar y le pide confirmar su decisión. Una vez el usuario haya confirmado su decisión, el elemento se eliminará en la base de datos.		
<b>Caso</b>	Confirmar decisión		
<b>Resultado</b>	Exitoso		
<b>Detalle</b>	Elemento del equipo mecánico utilizado en aplicación eliminado exitosamente. Elemento eliminado no se aprecia en la interfaz gráfica del programa.		
<b>Caso</b>	Cancelar decisión		
<b>Resultado</b>	Exitoso		
<b>Detalle</b>	El sistema devuelve al usuario a la vista previa antes de presionar el botón “Eliminar”		
<b>Estado de la prueba</b>	Aprobada		

Nombre de la prueba	Identificador de la prueba	Identificador de requerimiento involucrado	Módulo conceptual
Crear Equipo de protección utilizados en aplicación	QA-Func-40	RF-40	Gestión de aplicaciones
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>• Tabla del Equipo de protección utilizados en aplicación creada en base de datos</li> <li>• Interfaz de la aplicación del programa (API) que permita la consulta de crear Equipo de protección utilizados en aplicación</li> <li>• Vista con botón que permita activar la instrucción</li> </ul>		

	<ul style="list-style-type: none"> <li>• Vista con formulario que permita seleccionar los datos requeridos para crear Equipo de protección utilizados en aplicación</li> <li>• Vista con botón que permita enviar formulario</li> <li>• Vista con botón que permita cancelar instrucción</li> </ul>
<b>Activador</b>	Click en el botón con la leyenda (“crear”)
<b>Descripción</b>	El usuario del sistema realiza click sobre el botón activador de la instrucción. Seguidamente se despliega un formulario que permite al usuario seleccionar mediante un checkbox las opciones deseadas. Una vez el usuario haya seleccionado al menos una opción, al dar click al botón “aceptar”, el elemento se creará en la base de datos.
<b>Caso</b>	<b>Selección de datos correctamente y dar click sobre botón agregar</b>
<b>Resultado</b>	Exitoso
<b>Detalle</b>	Equipo de protección utilizados en aplicación creado exitosamente.
<b>Caso</b>	<b>Selección de datos correctamente y dar click sobre botón cancelar</b>
<b>Resultado</b>	Exitoso
<b>Detalle</b>	El sistema devuelve al usuario a la vista previa antes de presionar el botón “crear”
<b>Caso</b>	<b>Click en enviar sin haber seleccionado ningún dato en el formulario</b>
<b>Resultado</b>	Exitoso
<b>Detalle</b>	El sistema impide la acción al desactivar el botón puesto que el formulario no contiene valores
<b>Estado de la prueba</b>	Aprobada

<b>Nombre de la prueba</b>	<b>Identificador de la prueba</b>	<b>Identificador de requerimiento involucrado</b>	<b>Módulo conceptual</b>
Ver equipo de protección utilizados en aplicación	QA-Func-41	RF-41	Gestión de aplicaciones
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>• Tabla de equipo de protección utilizados en aplicación creada en base de datos</li> <li>• Al menos una fila de equipo de protección utilizados en aplicación creada en la base de datos</li> <li>• Interfaz de la aplicación del programa (API) que permita la consulta de ver equipo de protección utilizados en aplicación</li> <li>• Vista con tabla que permita la renderización de datos</li> </ul>		

<b>Activador</b>	Click en el link con la leyenda (“equipo de protección utilizados en aplicación”)
<b>Descripción</b>	El usuario del sistema realiza click sobre el link activador de la instrucción. Seguidamente el sistema renderiza en una tabla algunos de los datos correspondientes a la estructura.
<b>Caso</b>	<b>Click en el link con la leyenda “equipo de protección utilizados en aplicación”</b>
<b>Resultado</b>	Exitoso
<b>Detalle</b>	Datos de la tabla renderizados correctamente.
<b>Estado de la prueba</b>	Aprobada

<b>Nombre de la prueba</b>	<b>Identificador de la prueba</b>	<b>Identificador de requerimiento involucrado</b>	<b>Módulo conceptual</b>
Editar equipo de protección utilizados en aplicación	QA-Func-42	RF-42	Gestión de aplicaciones
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>• Tabla de equipo de protección utilizados en aplicación creada en base de datos</li> <li>• Al menos una fila equipo de protección utilizados en aplicación creada en la base de datos</li> <li>• Interfaz de la aplicación del programa (API) que permita la consulta de editar equipo de protección utilizados en aplicación</li> <li>• Vista con botón que permita activar la instrucción</li> <li>• Vista con formulario que permita seleccionar los datos requeridos para editar Insumo</li> <li>• Vista con botón que permita enviar formulario</li> <li>• Vista con botón que permita cancelar instrucción</li> </ul>		
<b>Activador</b>	Click en el botón con la leyenda (“editar”)		
<b>Descripción</b>	El usuario del sistema realiza click sobre el botón activador de la instrucción. Seguidamente se despliega un formulario que permite al usuario seleccionar la información requerida. Una vez el usuario haya editados los datos deseados, al dar click al botón “editar”, el elemento se actualizará en la base de datos.		
<b>Caso</b>	<b>Seleccionar datos correctamente y dar click sobre botón agregar</b>		
<b>Resultado</b>	Exitoso		
<b>Detalle</b>	Elemento de equipo de protección utilizados en aplicación actualizado exitosamente. Elemento actualizado se aprecia en la interfaz gráfica del programa.		
<b>Caso</b>	<b>Seleccionar datos correctamente y dar click sobre botón cancelar</b>		
<b>Resultado</b>	Exitoso		

<b>Detalle</b>	El sistema devuelve al usuario a la vista previa antes de presionar el botón “editar”
<b>Caso</b>	<b>Click en actualizar sin haber cambiado ningún dato en el formulario</b>
<b>Resultado</b>	Exitoso
<b>Detalle</b>	El sistema impide la acción puesto que el formulario no contiene modificaciones y actualización resulta innecesaria
<b>Estado de la prueba</b>	Aprobada

<b>Nombre de la prueba</b>	<b>Identificador de la prueba</b>	<b>Identificador de requerimiento involucrado</b>	<b>Módulo conceptual</b>
Eliminar Equipo de protección utilizados en aplicación	QA-Func-43	RF-43	Gestión de aplicaciones
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>• Tabla de equipo de protección utilizados en aplicación creada en base de datos</li> <li>• Al menos una fila de equipo de protección utilizados en aplicación creada en la base de datos</li> <li>• Interfaz de la aplicación del programa (API) que permita la consulta de eliminar equipo de protección utilizados en aplicación</li> <li>• Vista con botón que permita activar la instrucción</li> <li>• Vista con botón que permita confirmar la instrucción</li> <li>• Vista con botón que permita cancelar instrucción</li> </ul>		
<b>Activador</b>	Click en el botón con la leyenda (“Eliminar”)		
<b>Descripción</b>	El usuario del sistema realiza click sobre el botón activador de la instrucción. Seguidamente se despliega un mensaje que notifica al usuario de la instrucción a realizar y le pide confirmar su decisión. Una vez el usuario haya confirmado su decisión, el elemento se eliminará en la base de datos.		
<b>Caso</b>	Confirmar decisión		
<b>Resultado</b>	Exitoso		
<b>Detalle</b>	Elemento del equipo de protección utilizados en aplicación eliminado exitosamente. Elemento eliminado no se aprecia en la interfaz gráfica del programa.		
<b>Caso</b>	Cancelar decisión		
<b>Resultado</b>	Exitoso		
<b>Detalle</b>	El sistema devuelve al usuario a la vista previa antes de presionar el botón “Eliminar”		

<b>Estado de la prueba</b>	Aprobada
----------------------------	----------