



Trabajo de Grado

APLICACIÓN WEB|MÓVIL PARA EL CONTROL DE PRODUCCIÓN AGRÍCOLA
DE UNA RED DE AGRICULTORES.

Por:

JUAN DANIEL GALARZA RODRÍGUEZ

Código 201323966

Directora:

BEATRIZ EUGENIA FLORIAN GAVIRIA, Ph.D.

Profesora Asistente EISC

UNIVERSIDAD DEL VALLE
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA DE SISTEMAS Y COMPUTACIÓN
PROGRAMA ACADÉMICO DE INGENIERÍA DE SISTEMAS
SANTIAGO DE CALI
Abril de 2019

Contenido

1	Planteamiento del problema	1
2	Justificación del proyecto	2
2.1	Justificación académica	2
2.2	Justificación económica	2
2.3	Justificación social	2
3	Objetivos	3
3.1	Objetivo general	3
3.2	Objetivos específicos	3
3.3	Productos obtenidos	3
4	Alcances de la propuesta	5
5	Marco Referencial	6
5.1	Marco teórico	6
5.1.1	Costos de producción y comercialización agrícola	6
5.1.2	Metodologías ágiles de desarrollo de software	9
5.1.2.1	SCRUM	9
5.1.3	Planeación y control de proyectos de desarrollo de software ágiles	11
5.1.3.1	Zube	11
5.1.4	Desarrollo de Aplicaciones Web	12
5.1.4.1	Arquitectura de una aplicación web	12
5.1.5	Herramienta One Touch SDK para integración biométrica	13
5.1.6	Diseño web responsive	13
5.1.6.1	Bootstrap	14
5.1.7	Visualización de datos	14
5.1.7.1	D3	15
5.1.8	Sistemas de Información Geográfico (SIG)	15
5.2	Antecedentes o estado del arte	16
5.2.1	Hecopaun	16
5.2.2	AgroWin	17
5.2.3	SynAgro	18
5.2.4	Agroptima	18

6 Metodología de desarrollo	21
6.1 Etapa de análisis.....	21
6.1.1 Artefactos y herramienta de planeación del proyecto	21
6.1.2 Herramientas de ambiente de desarrollo de software.....	24
6.2 Etapa de diseño	28
6.2.1 Arquitectura del software y tecnologías utilizadas	28
6.2.2 Modelo de datos	30
7 Desarrollo de un módulo para la gestión de usuarios y permisos ..	31
7.1 Análisis.....	31
7.2 Implementación.....	32
7.3 Pruebas.....	36
8 Desarrollo de un módulo para gestionar los lotes que pertenecen a los agricultores.....	40
8.1 Análisis.....	40
8.2 Implementación.....	41
8.3 Pruebas.....	44
9 Desarrollo de un módulo para la gestión de datos de los diferentes cultivos sembrados en los lotes.	48
9.1 Análisis.....	48
9.2 Implementación.....	49
9.3 Pruebas.....	51
10 Desarrollo de un módulo de gestión de datos de costos y gastos de producción agrícola correspondientes a los lotes.	55
10.1 Análisis.....	55
10.2 Implementación.....	57
10.3 Pruebas.....	60
11 Desarrollo de un módulo para la gestión de datos de producción que se presenta en los diferentes cultivos.	64
11.1 Análisis.....	64
11.2 Implementación.....	65
11.3 Pruebas.....	67
12 Desarrollo de un módulo de reportes que permita procesar y analizar los datos de producción agrícola para el agricultor y su entorno de red de agricultores.....	71
12.1 Análisis.....	71

12.2 Implementación.....	73
12.3 Pruebas.....	75
13 Conclusiones	79
13.1 Trabajo futuro.....	80
14 Referencias	81

Listado de tablas

Tabla 1 Productos esperados por objetivo específico	4
Tabla 2 Tabla comparativa de aplicaciones	19
Tabla 3 Funcionalidades por tipo de usuario para el módulo de gestión de usuarios y permisos.....	32
Tabla 4 Diseño y ejecución de pruebas funcionales para el módulo de gestión de usuarios.....	36
Tabla 5 Resumen de ejecución de casos de prueba para el módulo de gestión de usuarios.....	37
Tabla 6 Funcionalidades por tipo de usuario del módulo para gestionar los lotes que pertenecen a los agricultores	40
Tabla 7 Diseño y ejecución de pruebas funcionales para el módulo de gestión de lotes.....	44
Tabla 8 Resumen de ejecución de casos de prueba para el módulo de gestión de usuarios.....	45
Tabla 9 Funcionalidades por tipo de usuario del módulo para gestionar los diferentes cultivos sembrados en los lotes	48
Tabla 10 Diseño y ejecución de pruebas funcionales para el módulo de gestión de cultivos	52
Tabla 11 Resumen de ejecución de casos de prueba para el módulo de gestión de usuarios.....	52
Tabla 12 Funcionalidades por tipo de usuario del módulo para gestionar los datos de costos y gastos de producción agrícola.	55

Tabla 13 Diseño y ejecución de pruebas funcionales para el módulo de gestión de costos y gastos	60
Tabla 14 Resumen de ejecución de casos de prueba para el módulo de gestión de usuarios.....	61
Tabla 15 Funcionalidades por tipo de usuario del módulo para gestionar los datos de producción que se presenta en los diferentes cultivos.	64
Tabla 16 Diseño y ejecución de pruebas funcionales para el módulo de gestión de producción.....	67
Tabla 17 Resumen de ejecución de casos de prueba para el módulo de gestión de producción.....	68
Tabla 18 Funcionalidades por tipo de usuario del módulo de reportes.	71
Tabla 19 Diseño y ejecución de pruebas funcionales para el módulo de reportes	77

Listado de figuras

Ilustración 1 Actividades en el proceso de producción agrícola Fuente: Elaboración propia	7
Ilustración 2 Actividades en el proceso de comercialización agrícola	7
Ilustración 3 Costos en agricultura Fuente: Elaboración propia	8
Ilustración 5 Diseño web responsive	14
Ilustración 7 ArcGis, sistema de información geográfica	16
Ilustración 8 Hecopaun, herramienta móvil para el control de producción en el sector agrícola	17
Ilustración 9 AgroWin, software de administración y control para el agro	17
Ilustración 10 SynAgro, software agropecuario	18
Ilustración 11 Agroptima, software especializado en registro de labores agrícolas	18
Ilustración 12 Historia de usuario definida en Zube.....	22
Ilustración 13 Product Backlog definido en Zube	23
Ilustración 14 Sprints definidos en Zube	23

Ilustración 15 Sprint Backlog definido en Zube Fuente: Elaboración propia	24
Ilustración 16 Entorno de desarrollo integrado AWS Cloud9 Fuente: Elaboración propia	25
Ilustración 17 Entorno de desarrollo integrado NetBeans Fuente: Elaboración propia	25
Ilustración 18 pgAdmin, plataforma de desarrollo para PostgreSQL Fuente: Elaboración propia	26
Ilustración 19 Selenium IDE Fuente: Elaboración propia	27
Ilustración 20 Repositorio agros en GitHub Fuente: Elaboración propia	27
Ilustración 21 Vertabelo, herramienta para diseño de la base de datos Fuente: Elaboración propia	28
Ilustración 22 Arquitectura Modelo-Plantilla-Vista Fuente: http://www.maestrosdelweb.com	29
Ilustración 23 Tecnologías usadas por capa para el desarrollo de la aplicación Fuente: Elaboración propia	29
Ilustración 24 Modelo de la base de datos Fuente: Elaboración propia	30
Ilustración 25 Historias de usuario del módulo de gestión de usuarios y permisos Fuente: Elaboración propia	33
Ilustración 26 Funcionalidad de crear un agricultor Fuente: Elaboración propia ...	34
Ilustración 27 Funcionalidad de modificar agricultor Fuente: Elaboración propia..	34
Ilustración 28 Funcionalidad de listar agricultores Fuente: Elaboración propia	35
Ilustración 29 Funcionalidad de modificar perfil agricultor Fuente: Elaboración propia	35
Ilustración 30 Casos de prueba para la funcionalidad crear agricultor	36
Ilustración 31 Ejecución de casos de prueba para la funcionalidad crear agricultor	36
Ilustración 32 Historias de usuario del módulo de gestión de usuarios y permisos Fuente: Elaboración propia	42
Ilustración 33 Funcionalidad de crear una finca Fuente: Elaboración propia	43
Ilustración 34 Funcionalidad de detalles de finca Fuente: Elaboración propia	43

Ilustración 35 Casos de prueba para la funcionalidad modificar finca	44
Ilustración 36 Ejecución de casos de prueba para la funcionalidad modificar finca	44
Ilustración 37 Historias de usuario del módulo para la gestión de datos de los diferentes cultivos sembrados Fuente: Elaboración propia	49
Ilustración 38 Funcionalidad de registrar un tipo de cultivo Fuente: Elaboración propia	50
Ilustración 39 Funcionalidad de registrar un bloque Fuente: Elaboración propia ..	50
Ilustración 40 Funcionalidad de listar bloques Fuente: Elaboración propia	51
Ilustración 41 Casos de prueba para la funcionalidad de asignar una ubicación geográfica a un bloque	52
Ilustración 42 Ejecución de casos de prueba para la funcionalidad de modificar tipo de cultivo	52
Ilustración 43 Historias de usuario del módulo para la gestión de datos de costos y gastos de producción agrícola Fuente: Elaboración propia	57
Ilustración 44 Funcionalidad de registrar tipo de costo Fuente: Elaboración propia	58
Ilustración 45 Funcionalidad de listar registros de costos por producto Fuente: Elaboración propia	59
Ilustración 46 Funcionalidad de ingresar registro de costo por producto Fuente: Elaboración propia	59
Ilustración 47 Casos de prueba para la funcionalidad de modificar un registro de costo por producto	60
Ilustración 48 Ejecución de casos de prueba para la funcionalidad de modificar un registro de costo por producto	60
Ilustración 49 Historias de usuario del módulo para la gestión de datos de producción Fuente: Elaboración propia	65
Ilustración 50 listar registros de producción asociados a un bloque Fuente: Elaboración propia	66
Ilustración 51 Gestión de registros de ventas de un bloque Fuente: Elaboración propia	66

Ilustración 52 Casos de prueba para la funcionalidad de modificar un registro de producción.....	67
Ilustración 53 Ejecución de casos de prueba para la funcionalidad de modificar un registro de producción.....	67
Ilustración 54 Historias de usuario del módulo de reportes Fuente: Elaboración propia Implementación	73
Ilustración 55 Funcionalidad de generar dashboard Fuente: Elaboración propia.	74
Ilustración 56 Funcionalidad de generar detalles generales de un bloque Fuente: Elaboración propia	75
Ilustración 57 Funcionalidad generar detalles de producción y ventas de un bloque Fuente: Elaboración propia	75
Ilustración 58 Casos de prueba para la funcionalidad de detalles generales del bloque	77
Ilustración 59 Ejecución de casos de prueba para la funcionalidad de detalles generales del bloque	77

Acrónimos y Glosario

Bloque: Nombre que se le da en el sector agrícola a los cultivos sembrados.

CSS: Es un lenguaje utilizado para describir la presentación que tienen las páginas web, incluidos los colores, el diseño y las fuentes. Una de las grandes ventajas que ofrece este lenguaje es la capacidad que tiene para permitir adaptar la presentación de la página a diferentes tipos de dispositivos, como pantallas grandes, pantallas pequeñas o impresoras. CSS es independiente de HTML y esta separación, facilita el mantenimiento de sitios, el intercambio de hojas de estilo entre páginas y la personalización de páginas en diferentes entornos (W3C, 2017).

Django: Es uno de los framework más conocido y posee una gran comunidad de apoyo que lo utiliza. Este es un entorno de desarrollo web de alto nivel con Python (un lenguaje de programación), que fomenta el desarrollo rápido y un diseño limpio y pragmático. Este entorno de desarrollo ha sido diseñado para ayudar a los programadores a llevar las aplicaciones desde el concepto hasta su completitud de una manera muy rápida, además ofrece una gran escalabilidad a las aplicaciones (Django Software Foundation, 2017b).

Entorno de desarrollo integrado (IDE)

Extreme Programing (XP)

Framework: Conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar.

HTML le da un valor agregado a un texto estándar. HyperText se refiere a enlaces que conectan una página Web con otra, ya sea dentro de una página web o entre diferentes sitios web. Los vínculos son un aspecto fundamental de la Web. Además de lo anterior, esta tecnología también permite el contenido de archivos multimedia como imágenes, sonido o videos dentro de la página (Mozilla Developer Network, 2017).

HTML: El HyperText Markup Language (HTML), es el elemento de construcción más básico de una página web y es usado para crear y representar visualmente una página web. Determina el contenido de la página web dentro de una estructuración, pero no determina su funcionalidad.

JavaScript: Es un lenguaje ligero e interpretado, orientado a objetos con funciones de primera clase, más conocido como el lenguaje de script para páginas web, pero también usado en muchos entornos sin navegador, tales como node.js o Apache CouchDB. Este lenguaje es utilizado para crear páginas web dinámicas, esto quiere decir que incorpora efectos de texto como aparecer y desaparecer, animaciones acciones que se activan al presionar botones y ventanas con mensajes de aviso al usuario (Mozilla Developer Network, 2017).

Kit de desarrollo de software (SDK)

Mapeo objeto-relaciona (ORM)

Modelo Vista Controlador (MVC)

Modelo Vista Template (MVT)

Para el correcto desarrollo de las aplicaciones web, se debe tener en cuenta que existen unas tecnologías y conceptos básicos, a continuación se darán a conocer algunos de estos:

Producto Interno Bruto (PIB)

Protocolo de transferencia de hipertexto (HTTP)

Python: Python es un lenguaje de programación de código abierto e interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible.

Responsive: Técnica de diseño web que busca la correcta visualización de una misma página en distintos dispositivos.

SCRUM: Metodología ágil de desarrollo de software.

Sistema de información geográfico (SIG)

Sprint: Bloque de tiempo de un mes o menos durante el cual se crea un incremento de producto “Terminado”, utilizable y potencialmente desplegable.

Agradecimientos

En primer lugar, quiero agradecer a Dios por las bendiciones que me han concedido en esta etapa de mi vida y en todos aquellos proyectos que he decidido emprender para mi crecimiento personal. Agradezco a mi familia por su apoyo incondicional, por brindarme su ayuda en momentos de necesidad, su ejemplo y su orgullo tras cada logro alcanzado. Gracias a los profesores que se quedarán en mi memoria por todas las enseñanzas transmitidas y a todas las demás personas que estuvieron presentes en mayor o menor medida durante este tiempo, creyendo en mí y dejando enseñanzas imborrables.

Agradezco a la profesora Beatriz Eugenia Florián Gaviria, Directora del proyecto, quien me acompañó y apoyó para el desarrollo exitoso del software, a través del seguimiento continuo durante el proceso del proyecto. Gracias por la confianza y por compartir su experiencia profesional durante este proceso.

A Rubén Darío Alvarado, ingeniero agrónomo por su tiempo y colaboración durante todo el proceso de desarrollo del proyecto, compartiendo su conocimiento en el sector agrícola.

A José David Palma, agricultor y a sus colaboradores por su disposición de tiempo para ayudar con la realización de pruebas de campo con datos reales.

A la Universidad del Valle por ofrecerme las instalaciones físicas necesarias para el desarrollo de este proyecto y por brindarme todos los recursos humanos y físicos que hicieron de este proceso de formación como ingeniero el mejor posible.

Resumen

El sector agrícola en Colombia representa un papel muy importante para el desarrollo económico del país, sin embargo para los pequeños y medianos agricultores existe un problema debido a que estos muchas veces no llevan a cabo el correcto seguimiento de planeación y proceso de producción, haciendo registros manuales no muy exactos de su producción. Además, al no ser rigurosos con el registro de sus costos y gastos diarios no pueden hacer análisis de su rentabilidad ni proyecciones a futuro. Por último no pueden establecer redes de agricultores para compartir información sobre sus procesos.

Este proyecto entrega una aplicación web/móvil que les permite a las personas que tienen extensiones de tierra y se desempeñan en el sector agrícola llevar un control de planeación y producción de sus cultivos, en particular gestionando los registros diarios de los gastos y costos que se van presentando durante toda la etapa de producción. Para esto, en el proyecto se desarrollaron seis módulos, los cuales son: gestión de usuarios, gestión de lotes, gestión de cultivos, gestión de gastos y costos, gestión de información producción y un módulo para análisis de datos y reportes.

Este documento tiene la siguiente estructura: En el Capítulo 1 se presenta la Definición del problema, en el Capítulo 2 la justificación de la propuesta, en el Capítulo 3 los objetivos del trabajo de grado, en el Capítulo 4 Alcances de la Propuesta, Capítulo 5 marco referencial, en el Capítulo 6 la metodología del trabajo de , después en los capítulos 7, 8, 9, 10, 11, 12 está el desarrollo de los módulos planteados en los objetivos específicos y por último en el capítulo 13 están las conclusiones.

1 Planteamiento del problema

Colombia es un país que se encuentra muy bien ubicado geográficamente ya que tiene diversidad en cuanto a climas, flora, fauna, cuencas hidrográficas y recursos naturales. Debido a estas características, la agricultura colombiana posee una gran variedad. El sector agrícola en Colombia representa un papel muy importante para el desarrollo económico del país, pues constituye la principal fuente de ingresos de su área rural. El producto interno bruto (PIB) de Colombia entre 2011 y 2015 tuvo una participación del 6.3% por parte de la agricultura (Rojas, 2016).

En el sector agrícola como en los otros sectores económicos del país, es importante llevar las cuentas de los gastos y costos que se tienen durante los procesos de producción, ya que por medio de estos se puede determinar cuáles labores son más costosas y las alternativas disponibles para disminuir los costos, se puede analizar la factibilidad de la siembra de un cultivo en ciertas zonas del país y también se puede realizar un balance financiero que determine qué tanto afectan los gastos y costos a las ganancias (EL TIEMPO, 2000). Es en este aspecto donde se presenta un problema para los pequeños y medianos agricultores, quienes muchas veces deciden no usar hojas de cálculo o informes contables y no registran sus egresos, debido a que no poseen herramientas para realizar un correcto análisis de la información que esto representa, esto lleva a que no se tengan en cuenta los gastos y costos que se realizan diariamente en el proceso de producción y provocan una idea errónea de las ganancias que se tienen o incluso puede hasta ocultar pérdidas.

En la actualidad, existen herramientas como “AgroWin®” que permiten llevar estos registros y generar reportes que dejan dar una visión general al agricultor de los gastos y costos que genera un lote. “AgroWin®” es un sistema diseñado para ayudar al agricultor con la gestión, planeación y seguimiento de su empresa y recursos, este permite llevar un control de producción y ventas, mano de obra y costos de desarrollo y producción de cada cultivo (InSoft, 2017). Sin embargo, la utilización de herramientas como esta se ve limitada porque la mayoría están pensadas para grandes productores ya que requieren inversiones considerables de dinero, además de personal entrenado en las terminales de recolección y procesamiento de los datos.

Esta problemática afecta un gran número de agricultores de Colombia, generando la necesidad de una herramienta que permita llevar los registros de gastos y costos que estos tienen en sus lotes sin tener que invertir mucho dinero o contratar personal para utilizarla, además, permitir actualizar la información desde cualquier lugar debido a que la forma de trabajo de un agricultor lo mantiene en constante movimiento.

Debido a todo esto, se propuso una aplicación web/móvil que permitiera hacer el seguimiento del proceso de planeación y producción agrícola de una red de agricultores. En específico, la aplicación gestiona los gastos y costos así como también genera reportes que le permiten al agricultor tener una visión crítica del comportamiento de la producción en sus propios lotes y de los que se encuentran en sus cercanías que pertenezcan a la red de agricultores.

2 Justificación del proyecto

A continuación se plantea las justificaciones económicas, sociales y académicas por los cuales se realiza el proyecto.

2.1 Justificación académica

El desarrollo del proyecto me permitirá aplicar los diversos conocimientos adquiridos a lo largo de mi carrera, principalmente en los cursos orientados al desarrollo de proyectos de software como lo son desarrollo de software I, desarrollo de software II, aplicaciones en la web y redes inalámbricas, sistemas de información y técnicas de pruebas de software. Además de permitirme adquirir nuevos conocimientos en las áreas de visualización de datos, desarrollo móvil y tecnologías de la información.

2.2 Justificación económica

El desarrollo de la aplicación web que permita la recolección, procesamiento y visualización de los datos de costos y gastos de los lotes pertenecientes a los agricultores, contribuiría positivamente a la parte económica, ya que se les permitirá tener un control total sobre sus egresos y realizar un balance financiero que determine qué tanto afectan los gastos y costos a las ganancias, además de poder determinar cuáles labores son más costosas y buscar las alternativas disponibles para disminuir los costos. Adicionalmente se reducirían otros costos como en papelería y gastos necesarios para la toma de datos por medio de un formato físico.

2.3 Justificación social

La aplicación está pensada de forma que los agricultores hagan parte de una red, esto les permitirá compartir información con su entorno y de esta manera, darse cuenta de aspectos como que producto se está usando para tratar ciertos cultivos o que cultivos se están sembrando en ciertas zonas del país.

Esta acercamiento con su entorno que tendrán los agricultores por medio de la red, permitirá mejorar las condiciones de la comunidad porque por medio de esta se podrá optimizar la fase de producción ya sea ahorrando costos de productos innecesarios o que pueden ser remplazados y también mejorando algunas prácticas debido a las recomendaciones que se hagan observando los otros cultivos.

3 Objetivos

En esta sección del documento se muestran los objetivos del proyecto, primero se expondrá el objetivo general y posteriormente los seis objetivos específicos que aseguran el cumplimiento del objetivo general planteado.

3.1 Objetivo general

Desarrollar una aplicación web/móvil para el control de producción agrícola de una red de agricultores.

3.2 Objetivos específicos

1. Desarrollar un módulo para la gestión de usuarios que tendrán acceso al sistema dependiendo del rol que desempeñen dentro de la aplicación.
2. Desarrollar un módulo para gestionar los lotes que pertenecen a los agricultores.
3. Desarrollar un módulo para la gestión de datos de los diferentes cultivos sembrados en los lotes.
4. Desarrollar un módulo de gestión de datos de costos y gastos de producción agrícola correspondientes a los lotes.
5. Desarrollar un módulo para la gestión de datos de producción que se presenta en los diferentes cultivos.
6. Desarrollar un módulo de reportes que permita procesar y analizar los datos de producción agrícola para el agricultor y su entorno de red de agricultores.

3.3. Productos obtenidos

La siguiente tabla especifica los productos obtenidos asociados a cada objetivo específico. Se destacan los productos extra resultados de este trabajo que no fueron prometidos en el anteproyecto pero que se pudieron realizar dentro del marco de este trabajo de grado.

Tabla 1 Productos esperados por objetivo específico

Objetivo	Productos	Evidencias de los productos dentro de este documento
1	Análisis y diseño (HU, Diseños iniciales de las interfaces web/móvil, Diseños de pruebas funcionales, esquema inicial E-R) Implementación (Código) Pruebas (Reporte de ejecución de pruebas funcionales)	Capítulo 7 Anexos: Diseño de pruebas funcionales-Modulo de usuarios, MRP-Modulo de usuarios, Modelo de datos.
2	Análisis y diseño (HU, Diseños de pruebas funcionales) Implementación (Código) Pruebas (Reporte de ejecución de pruebas funcionales)	Capítulo 8 Anexos: Diseño de pruebas funcionales-Modulo de lotes, MRP-Modulo de lotes. <i>Software biométrico auxiliar. Anexo: BioAgros.mp4</i>
3	Análisis y diseño (HU, Diseños de pruebas funcionales) Implementación (Código) Pruebas (Reporte de ejecución de pruebas funcionales)	Capítulo 9 Anexos: Diseño de pruebas funcionales-Modulo de cultivos, MRP-Modulo de cultivos.
4	Análisis y diseño (HU, Diseños de pruebas funcionales) Implementación (Código) Pruebas (Reporte de ejecución de pruebas funcionales)	Capítulo 10 Anexos: Diseño de pruebas funcionales-Modulo de costos, MRP-Modulo de costos.
5	Análisis y diseño (HU, Diseños de pruebas funcionales) Implementación (Código) Pruebas (Reporte de ejecución de pruebas funcionales)C	Capítulo 11 Anexos: Diseño de pruebas funcionales-Modulo de producción, MRP-Modulo de producción.
6	Análisis y diseño (HU, Diseños de pruebas funcionales) Implementación (Código) Pruebas (Reporte de ejecución de pruebas funcionales)	Capítulo 12 Anexos: Diseño de pruebas funcionales-Modulo de reportes, MRP-Modulo de reportes.
Nota: Los productos en <i>verde</i> corresponden a productos extra resultado de este trabajo de grado que no fueron prometidos inicialmente en el anteproyecto		

4 Alcances de la propuesta

El proyecto se definió como una aplicación web/móvil, que permite a los agricultores llevar un control de planeación y producción agrícola en la fase de cultivo, por lo tanto, no se registraran datos diferentes a los de producción, gastos y costos que generen los lotes, los informes y reportes no usaran ninguna otra información diferente de la mencionada anteriormente. La aplicación no ofrece funcionalidades para llevar una contabilidad completa de las fincas o lotes, tampoco permite tener un manejo de nómina y no tiene algún modulo para realizar el manejo de inventario.

Para el aspecto móvil de la aplicación no se prometió desarrollar una aplicación diferente, el diseño de la aplicación web es *responsive* para que pueda ser visualizada adecuadamente en los dispositivos móviles como teléfonos inteligentes o tabletas.

5 Marco Referencial

En esta sección del documento se mostrará el marco referencial, constituido por el marco teórico y los antecedentes o estado del arte.

5.1 Marco teórico

En esta sección encontraremos el marco teórico, que es una síntesis de las teorías y conceptos que se utilizaron en el desarrollo de este proyecto, podemos encontrar conceptos relacionados con producción y comercialización agrícola, gestión y planificación de proyectos, metodologías de desarrollo de software, temas sobre desarrollo web y móvil, conceptos sobre lenguajes de programación y visualización de la información.

5.1.1 Costos de producción y comercialización agrícola

El concepto de producción agrícola se utiliza en el ámbito económico para hacer referencia a los diferentes tipos de productos y beneficios que puede generar la actividad agrícola. Cuando se usa el término de producción agrícola, se está haciendo referencia todo aquello que es resultado de la agricultura. La agricultura, es decir, el cultivo de cereales, granos y vegetales, es una de las actividades más importante para la subsistencia del ser humano, debido a esto su producción siempre es una parte relevante en las economías de la mayoría de regiones en el planeta.

Para quienes trabajan en el sector de la agricultura, la producción agrícola es un aspecto que deben tener muy en cuenta a la hora de pensar en los beneficios que pueden obtener. Esto se debe a que la producción agrícola debe ser controlada y organizada de la manera más apropiada, se deben conocer los ciclos naturales y de los productos que se desea cultivar, así como también los factores climáticos que muchas veces pueden afectar y hacer perder todo el trabajo invertido. Por último, para que la producción agrícola sea rentable, la misma debe permitir recuperar todas las inversiones que se han realizado, además de superarlas con el fin de generar ganancias (EcuRed, 2018).

La Ilustración 1 se muestra las principales actividades realizadas durante el proceso de producción agrícola. El primer paso para llevar a cabo la producción es la preparación del suelo, para este proceso se deben hacer el barbecho, que es una técnica de la agricultura por la cual la tierra de cultivo se deja sin sembrar durante uno o varios ciclos vegetativos. También se debe hacer un subsuelo, que se usa para suelos profundos en donde se busca romper capas endurecidas por tráfico para una mayor fertilidad del suelo y humedad. Por último para preparar el suelo se debe hacer un rastreo, una técnica usada para preparar una buena cama de siembra de por lo menos 10 cm de tierra mullida, lo anterior se logra con uno o dos pasos de rastra y ayuda a eliminar la primera generación de maleza.

Después de realizar la preparación del suelo se procede a hacer la siembra de la fruta o verdura seleccionada, posteriormente se debe regar y fertilizar. Después se procede a realizar las labores culturales, también conocidas como las labores, como lo son las escardas (labores para el control de malezas), poda, entutorado (se usan para dar soporte al crecimiento de determinadas plantas) y polinización.

También se debe realizar una sanidad del cultivo, es decir realizar un control de plagas y enfermedades ya sea usando productos o realizando labores sobre el cultivo. Por último una vez completado todo el proceso de producción se realiza la cosecha de la fruta o el vegetal que se había sembrado.

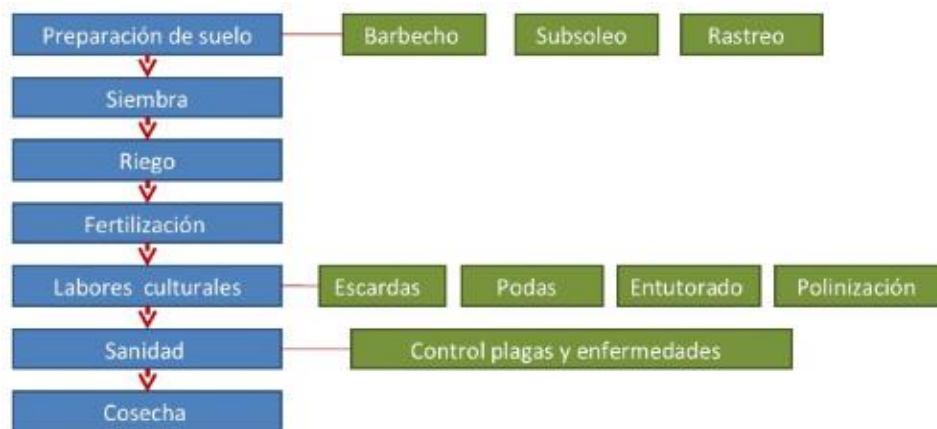


Ilustración 1 Actividades en el proceso de producción agrícola
Fuente: Elaboración propia

Una vez terminado todo el proceso de producción agrícola, se lleva a cabo el proceso de comercialización agrícola. Durante este proceso se realiza el secado junto con la adición de pesticidas o fungicidas a la producción obtenida. Además la producción debe ser almacenada y mantenida en stock hasta realizar su venta. Después de vender la producción se debe hacer el transporte de esta hasta su lugar de destino. La Ilustración 2 se muestra las actividades que se realizan durante el proceso de comercialización agrícola.

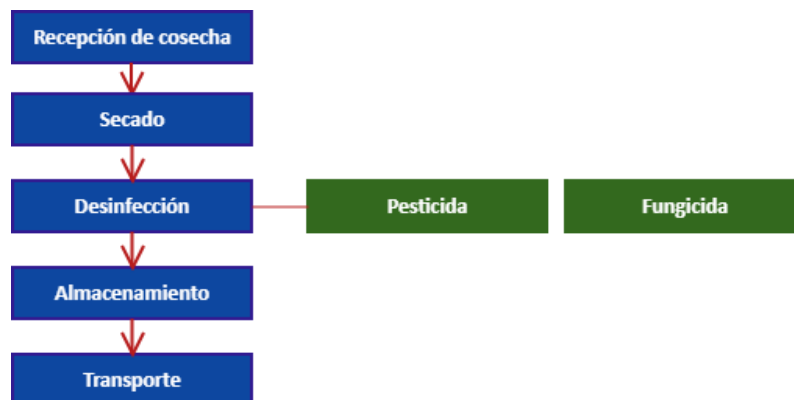


Ilustración 2 Actividades en el proceso de comercialización agrícola

Para hablar de costos en la producción y comercialización agrícola, debemos tener en cuenta qué actividades se considerarán en el proceso de producción agrícola, la Ilustración 1 y la Ilustración 2 nos resumen cuales son actividades propias de estos procesos. Otro aspecto a tomar en cuenta es la magnitud de esfuerzo necesario dentro de estas actividades y sus tareas para alcanzar los objetivos de producción y comercialización.

Podemos clasificar los costos de producción agrícola en dos grandes factores. El primero, son los costos de producción, que son aquellos en los que es necesario incurrir desde la preparación del terreno hasta la cosecha incluida. El segundo, son los costos de comercialización, que son en los que se incurre una vez efectuada la cosecha y hasta la generación de los ingresos por la venta del producto, una vez que se ha optado por una modalidad dada de comercialización.

Dentro de los costos de producción podemos encontrar los costos fijos, variables y de cosecha. En los costos fijos tenemos los costos generados por el factor tierra, los costos de laboreo como preparación y siembra, y los costos de trabajo realizado por terceros. Los costos variables corresponden a aquellos que en su cuantía total varían en función de la producción. Los costos de cosecha son los que permiten convertir al cultivo en un producto comercializable y también los costos generados por maquinaria y mano de obra.

Dentro de los costos de comercialización podemos encontrar unos costos variables y unos costos fijos. Los costos variables son dependientes del volumen de producción comercializada y varían en función del precio de venta o de los volúmenes comercializados. Los costos fijos se vinculan con los generados por instalaciones propias para mantenimiento del grano o su tratamiento que luego, en función de los volúmenes mantenidos o tratados se considerarán como un costo (Osorio, 1995).

En la Ilustración 3 se muestra un gráfico resumiendo la clasificación de los costos que se presentan en la agricultura. En color azul se encuentran las categorías y en color verde algunos de los costos correspondientes a las categorías.

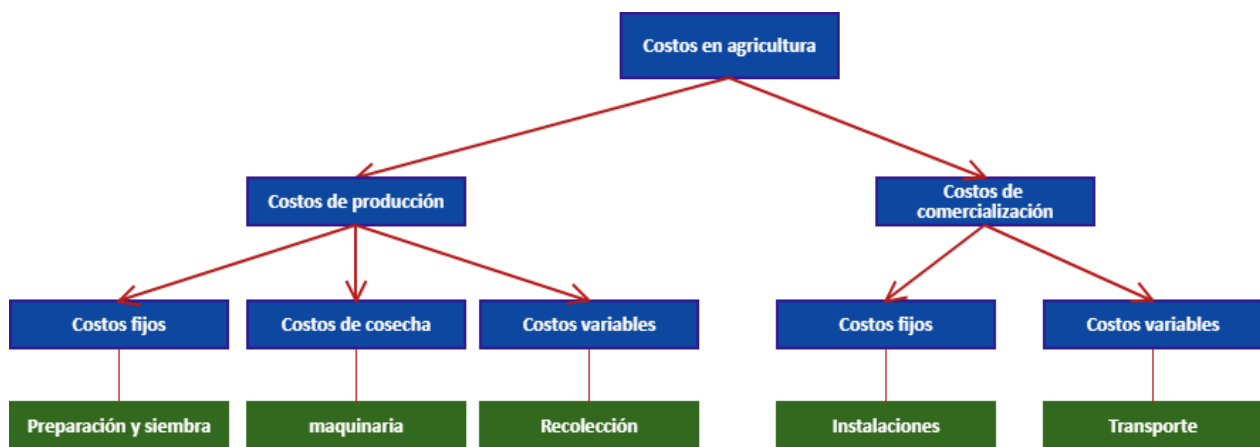


Ilustración 3 Costos en agricultura
Fuente: Elaboración propia

La aplicación que se desarrolló en este trabajo de grado inicialmente estaba enfocado en apoyar a los agricultores a llevar un control de los costos que se generaban durante la producción agrícola, sin embargo con el apoyo de la documentación descrita anteriormente, se presentó la posibilidad de apoyar al agricultor también con el control de los costos que se generaban durante el proceso de comercialización. De esta manera además de permitirle al agricultor llevar un control sobre los costos de los productos y labores que requerían un cultivo, también se le permitiría llevar un control de los costos de comercialización como el generado por el transporte.

5.1.2 Metodologías ágiles de desarrollo de software

Las metodologías ágiles de desarrollo de software surgieron como alternativa a las metodologías de desarrollo tradicionales que se basaban en tener una documentación exhaustiva del proyecto e involucrar poco al cliente durante el proceso de desarrollo. Por el contrario, las metodologías ágiles dejan de centrarse en la documentación, para dar paso a un proceso que desarrolle el software de manera más rápida y poder responder eficientemente a los cambios que se pueden presentar a lo largo del proyecto, además de enfatizar en tener una estrecha relación entre el equipo de desarrollo y las partes interesadas en el negocio.

En 2001 se celebró una reunión en Utah – EEUU, donde participaron 17 profesionales en el área del desarrollo de software, en esta reunión se discutieron ideas compartidas y se expusieron varios enfoques para el desarrollo de software. La colección de valores y principios presentados en esta reunión, fueron expresados en el “Manifiesto for Agile Software Development” con sus correspondientes 12 principios (Beck, Beedle, & Bennekum, 2001). Poco después de esta reunión se formó un grupo de soporte para las personas que quieran explorar y aplicar los valores ágiles, sus principio y prácticas para construir software, este grupo fue llamado Agile Alliance (Agile Alliance, 2017).

Las metodologías ágiles han tenido un gran crecimiento en los últimos años, siendo adoptadas por muchas grandes empresas de reconocimiento a nivel mundial. Actualmente existen diversas metodologías ágiles, pero es importante resaltar que entre las más reconocidas encontramos **Scrum** y **Extreme Programing (XP)**. Hay prácticas que sobresalen de cada una, pues realizan actividades de una mejor manera que las otras prácticas.

5.1.2.1 SCRUM

Scrum es un **framework** diseñado para construir productos de software. **Scrum** es un proceso de equipo que tiene tres roles principales:

- **Product Owner:** tiene la responsabilidad de decidir qué trabajo deberá ser realizado.
- **ScrumMaster:** actúa como líder servicial, ayudando al equipo y a la organización a hacer el mejor uso de Scrum.

- Miembros del Equipo de Desarrollo: son los encargados de construir el producto en forma incremental, en una serie de períodos cortos de tiempo llamados **Sprints**.

Un **Sprint** es un bloque de tiempo de un mes o menos durante el cual se crea un incremento de producto “Terminado”, utilizable y potencialmente desplegable. Cada nuevo Sprint comienza inmediatamente después de la finalización del Sprint previo.

Durante cada **Sprint** el Equipo **Scrum** construirá y entregará un Incremento de Producto. Cada incremento es una mejora del producto que se encuentre en estado operativo y este correcto visualmente, esta mejora debe alcanzar los criterios de aceptación establecidos y está con un nivel de calidad mínimo.

Scrum incluye tres artefactos esenciales:

- **Product Backlog**: es una lista ordenada las actividades que se van a realizar durante el proyecto, y es la única fuente de requisitos para cualquier cambio a realizarse en el producto (Schwaber & Sutherland, 2013).
- **Sprint Backlog**: es el conjunto de elementos del **Product Backlog** seleccionados para un **Sprint**, más un plan para entregar el Incremento de producto y conseguir el objetivo del **Sprint**. En este se especifica que funcionalidad formará parte del próximo Incremento y del trabajo necesario para entregar esa funcionalidad en un Incremento “Terminado”.
- Incremento de Producto: es un resultado requerido de cada **Sprint**. Es una versión integrada del producto, mantenida en un nivel de calidad lo suficientemente alto como para poder ser lanzado si así lo decidiera el **Product Owner**.

Scrum incluye cinco Actividades o reuniones:

- Refinamiento del **Product Backlog**: esta actividad se realiza constantemente a lo largo del desarrollo de un proyecto **Scrum**. Esta actividad incluye:
 - Mantener el Product Backlog ordenado;
 - Eliminar o degradar ítems que ya no sean importantes;
 - Agregar o promover ítems que surgen o se vuelven importantes;
 - Dividir ítems en ítems más pequeños;
 - Unir ítems en ítems más grandes;
 - Estimar ítems.
- Planificación del **Sprint**: en esta reunión el Equipo **Scrum** colabora para seleccionar y comprender el trabajo que será realizado en el **Sprint** que está por comenzar. Esta reunión se realiza justo antes de empezar cada **Sprint**.
- **Scrum** Diario: esta actividad se realiza para asegurarse que todo marcha bien en pos de lograr el Objetivo del **Sprint**. La reunión tiene lugar a la misma hora y en el mismo lugar todos los días. En esta reunión cada miembro del equipo responde tres preguntas principales: ¿Qué hice ayer?, ¿Qué voy a hacer hoy? Y ¿Si tengo o no algún bloqueo?
- Revisión del **Sprint**: esta actividad se realiza al final de cada **Sprint**, donde el Equipo **Scrum** y las partes interesadas revisan el resultado del **Sprint**. Es en esta

actividad cuando usualmente el equipo de trabajo muestra el incremento logrado durante el **Sprint**.

- Retrospectiva del **Sprint**: Esta actividad se lleva a cabo al final de cada **Sprint** y tiene como propósito revisar cómo fueron las cosas respecto al proceso, la relación entre las personas y las herramientas utilizadas. El equipo identifica qué salió bien y qué no tan bien, e identifica potenciales mejoras.

Para la realización de este trabajo de grado, se tomó la decisión de usar esta metodología para el desarrollo de la aplicación. Para más información consultar la sección 6.1.2, que se encuentra en la página 21.

5.1.3 Planeación y control de proyectos de desarrollo de software ágiles

La gestión de proyectos es un proceso continuo que requiere de una estrategia global, apoyada por herramientas y prácticas de trabajo que incrementen la productividad. El propósito de planificar y controlar es proveer una propuesta uniforme para el desarrollo y la administración de los proyectos.

A través del ciclo de vida de un proyecto, se presentan dos categorías de actividades a realizar y que se encuentran directamente relacionadas: las actividades de gestión y las actividades de desarrollo del sistema: Las actividades de gestión son aquellas relacionadas con la administración de las organizaciones, personas, sistemas y procedimientos involucrados en el proceso de planificación y construcción del sistema. Las actividades de desarrollo del sistema se centran en el desarrollo mismo. Las metodologías de desarrollo están típicamente organizadas en distintas fases, agrupadas en áreas funcionales de análisis, diseño e implementación, basadas en una estructura de partición del trabajo.

Es tan grande la importancia de llevar una buena gestión del proyecto, que se estima que un 70% de los fracasos expresados en atrasos y en la no estimación de los costos reales de un proyecto, se deben a la no incorporación de prácticas esenciales durante su planeación y gestión (González, 2005).

5.1.3.1 Zube

Zube es una herramienta web que permite llevar una planificación y control de proyectos de desarrollo de software que implementan prácticas ágiles y especialmente la metodología **Scrum**. Dentro de las características de esta herramienta se encuentra la posibilidad de administrar el equipo de trabajo, permitiendo asignar responsabilidades específicas para cada miembro. Además de permitir llevar una planificación del proyecto implementando prácticas ágiles como las historias de usuario con su respectivo responsable, organizadas por sprints.

Esta herramienta ofrece un tablero de Kanban para la visualización de la planificación, esto nos da la posibilidad de conocer el avance de nuestro proyecto de una forma gráfica

y muy fácil de entender. Además de esto, la herramienta tiene una sección de analíticas en donde se presenta información gráfica para ver cuánto falta para terminar un sprint, cual es el progreso sobre un sprint o gráficos de eficiencia para ver cuánto tiempo le toma a su grupo de trabajo terminar una determinada tarea (“Zube,” 2018).

5.1.4 Desarrollo de Aplicaciones Web

Una aplicación web (web-based application) es un tipo especial de aplicación cliente/servidor, donde tanto el cliente (el navegador, explorador o visualizador) como el servidor (el servidor web) y el protocolo mediante el que se comunican (HTTP) están estandarizados y no han de ser creados por el programador de aplicaciones (Mora, 2013).

Las aplicaciones web son aquellas herramientas que los usuarios pueden utilizar accediendo a un servidor web a través de internet o de una intranet mediante un navegador. En otras palabras, es un programa que se codifica en un lenguaje interpretable por los navegadores web en la que se confía la ejecución al navegador. Entre las aplicaciones web más comunes encontramos tiendas online como MercadoLibre o Amazon, plataformas de aprendizaje como moodle, y sitios multimedia para presentación de videos como YouTube, entre otros.

5.1.4.1 Arquitectura de una aplicación web

Para conocer acerca de la arquitectura de una aplicación web, debemos hablar del patrón de arquitectura de software conocido como Modelo-Vista-Controlador (MVC), que intenta separar los datos y la lógica, de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Este patrón ha tenido un gran crecimiento desde la llegada del internet porque es una de las mejores formas de crear aplicaciones cliente-servidor, donde un cliente hace peticiones y el servidor se encarga de responderle (Django Software Foundation, 2017b).

Ahora explicaremos un poco en que consiste este patrón de diseño MVC:

- El *modelo (M)* es un modelo o representación de los datos. Este modelo permite extraer datos de su base de datos sin conocer las complejidades de la base de datos subyacente.
- La *vista (V)* es lo que se ve. Es la capa de representación para el modelo. En el computador, la vista es lo que ve en el navegador para una aplicación web o la interfaz de usuario para una aplicación de escritorio.
- El *controlador (C)* controla el flujo de información entre el modelo y la vista. Utiliza la lógica programada para decidir qué información se extrae de la base de datos a través del modelo y qué información se pasa a la vista. También obtiene información del usuario a través de la vista e implementa la lógica empresarial: ya sea al cambiar la vista, o al modificar datos a través del modelo, o ambos (Django Software Foundation, 2017b).

Las aplicaciones web dividen su diseño entre una capa de presentación y una capa de acceso a los datos. La capa de presentación es conocida como *Front-End*, en esta se utilizan todas aquellas tecnologías que funcionan dentro del navegador web como HTML, CSS y JavaScript. El objetivo de esta capa es el de estilizar las paginas para que estas sean cómodas de observar para el usuario, además de diseñar las interacciones que este puede tener con las páginas. La capa de acceso a los datos es conocida como Back-End, en esta se utilizan todas aquellas tecnologías que funcionan del lado del servidor, ya sea para procesar o manipular los datos y su lógica, dentro de estas tecnologías encontramos los lenguajes como Python, PHP, .NET y Java.

5.1.5 Herramienta One Touch SDK para integración biométrica

Java es un lenguaje de programación que puede ser usado en diferentes tipos de dispositivos, concurrente, es decir que puede ejecutar tareas con cierta simultaneidad por medio de procesos e hijos de ejecución; otra característica es que es orientado a objetos. Fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una única vez y lo ejecuten en cualquier dispositivo, lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra. Java es, a partir de 2012, uno de los lenguajes de programación más populares en uso, particularmente para aplicaciones de cliente-servidor de web, con unos diez millones de usuarios reportados (Oracle Corporation, 2018a).

Java posee grupos de herramientas que se reúnen para permitir programar usando diferentes tipos de hardware, estos llevan el nombre de “Software Development Kit” (Kit de desarrollo de software), también conocidos como SDK. Tal es el caso del **One Touch SDK**, esta es una herramienta de desarrollo de software para integrar la biométrica de huellas dactilares en un amplio conjunto de aplicaciones, servicios y productos basados en Java. Permite a los desarrolladores realizar operaciones biométricas básicas de huellas dactilares: capturar una huella dactilar de un lector de huellas dactilares DigitalPersona, extraer las características distintivas de la muestra de huellas dactilares capturadas y almacenar los datos resultantes en una plantilla para una comparación posterior de una huella dactilar enviada y una plantilla de huellas dactilares existente (DigitalPersona Inc, 2010).

5.1.6 Diseño web responsive

El diseño web responsive o adaptativo es una técnica de diseño web que busca la correcta visualización de una misma página en distintos dispositivos. Desde ordenadores de escritorio a tabletas y móviles.

Actualmente accedemos a sitios web desde diferentes tipos de dispositivos; ordenador, tableta, Smartphone, entre otros, por lo que, cada vez más surge la necesidad de que nuestra página web se adapte a los diferentes tamaños de los mismos.

El diseño responsive trata de redimensionar y colocar los elementos de la web de forma que se adapten al ancho de cada dispositivo permitiendo una correcta visualización y una mejor experiencia de usuario. Se caracteriza porque los layouts (contenidos) e imágenes son fluidos y se usa código media-queries de CSS3 (Socialmood, 2017).



Ilustración 4 Diseño web responsive

A continuación mencionaremos una tecnología usada actualmente para realizar diseños responsive en aplicaciones web:

5.1.6.1 Bootstrap

Bootstrap es un framework de código abierto que simplifica el proceso de creación de diseños web combinando CSS y JavaScript. Ha sido desarrollado por Twitter que recientemente liberó su versión 2.0. La mayor ventaja es que podemos crear interfaces que se adapten a los distintos navegadores (responsive design) apoyándonos en un framework potente con numerosos componentes webs que nos ahorrarán mucho esfuerzo y tiempo (Bootstrap, 2017).

Entre las principales características de Bootstrap tenemos (Bootstrap, 2017):

- Permite crear interfaces que se adapten a los diferentes navegadores, tanto de escritorio como tabletas y móviles a distintas escalas y resoluciones.
- Se integra perfectamente con las principales librerías JavaScript, por ejemplo JQuery.
- Dispone de distintos layout predefinidos con estructuras fijas a 940 píxeles de distintas columnas o diseños fluidos.

5.1.7 Visualización de datos

Actualmente vivimos en un mundo en donde la información crece de manera exponencial, cada vez se produce más información y cada vez tenemos más capacidad de acceso a dicha información. El entender y aprovechar esta información depende de nuestra capacidad de procesarla y transformarla más que en solo datos.

No obstante, los datos, entendidos como registros aislados, no tienen ningún significado específico. Sólo cuando realizamos un acercamiento y les aplicamos una interpretación, estos cobran sentido y se transforman en conocimiento. Hablando un poco del aspecto tecnológico, el cual ha permitido que el beneficio que obtenemos de los datos haya evolucionado en las últimas décadas para diseñar mecanismos de interpretación cada vez más robustos y asequibles. Y, entre estos mecanismos de explotación, el más importante es la visualización de datos (Gobierno de España, 2016).

“La visualización de datos es la presentación gráfica de información con dos propósitos. Por un lado, la interpretación y construcción de significado a partir de los datos (es decir, el análisis); y por otro lado, la comunicación.” (Gobierno de España, 2016).

La visualización es una herramienta muy poderosa para descubrir y comprender la lógica que se encuentra detrás de un conjunto de datos, así como para compartir esta interpretación con otras personas desde un punto de vista objetivo.

Con la importancia que tiene actualmente la visualización de datos, es indispensable elegir las herramientas correctas que nos permitan cumplir nuestros objetivos de la mejor manera, a continuación se presenta una librería que nos puede ayudar a presentar de mejor manera nuestra información:

5.1.7.1 D3

D3 es una librería de código abierto creada por Mike Bostock, esta se ha convertido en una de las herramientas dominantes para crear visualizaciones basadas en gráficos de vectores SVG. La ventaja de este tipo de gráficos es que no se ven alterados por la resolución de las pantallas, sin importar el zoom que se haga, las imágenes nunca se pixelan.

Entre los distintos gráficos que podemos generar con esta librería se encuentran los gráficos de redes, árboles, mapas o burbujas, además de los gráficos usuales de barras o dispersión. Además de soportar la visualización de mapas de manera interactiva, esta es una de las grandes ventajas que presenta esta librería.

5.1.8 Sistemas de Información Geográfico (SIG)

Un sistema de información geográfica (también conocido con los acrónimos SIG en español o GIS en inglés) es un sistema de información capaz de integrar, almacenar, editar, analizar, compartir y mostrar la información geográficamente referenciada. Los SIG son herramientas que permiten a los usuarios crear consultas interactivas, analizar la información espacial, editar datos, mapas y presentar los resultados de todas estas operaciones.

Los sistemas de información geográfica son un conjunto de herramientas que integra y relaciona diversos componentes (usuarios, hardware, software, procesos) que permiten la organización, almacenamiento, manipulación, análisis y modelización de grandes

cantidades de datos procedentes del mundo real que están vinculados a una referencia espacial (Zhao et al., 2010).

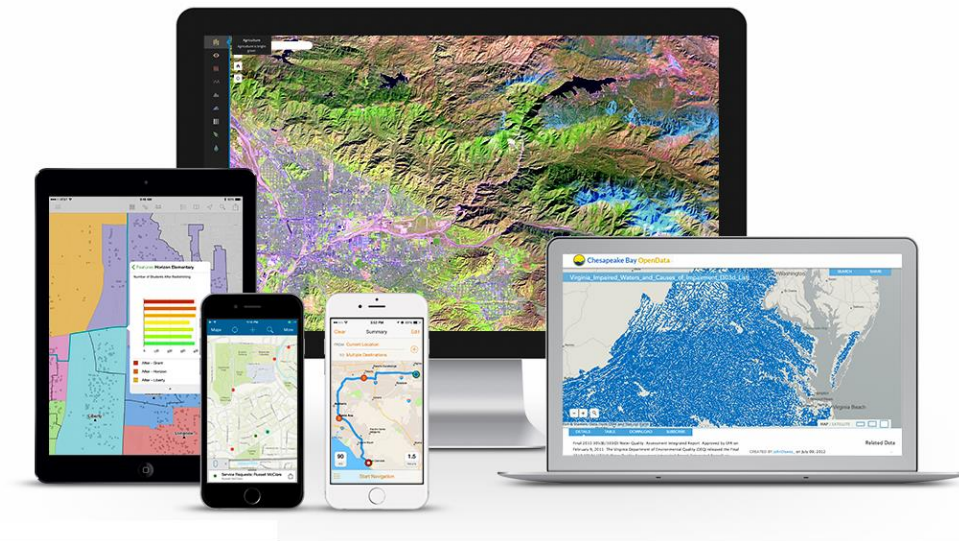


Ilustración 5 ArcGis, sistema de información geográfica

Para el caso específico de Django, esta cuenta con una expansión llamada GeoDjango, esta es utilizada en el desarrollo web con el framework Django y añade funcionalidades específicas que permiten almacenar y manipular datos geográficos. Su objetivo es facilitar lo más posible la creación de aplicaciones web SIG y aprovechar la potencia de los datos espacialmente habilitados (Django Software Foundation, 2017).

5.2 Antecedentes o estado del arte

Hasta el año 2018 existen diversos tipos de aplicaciones o software, podemos encontrar redes sociales, plataformas de aprendizaje o aplicaciones para realizar gestiones administrativas. Para entrar a analizar aplicaciones relacionadas con este proyecto, debemos tener en cuenta que estas deben presentar módulos para realizar el seguimiento de la producción y procesar esa información para generar reportes.

A continuación presentamos algunas aplicaciones que prestan los servicios antes mencionados:

5.2.1 Hecopaun

Es una herramienta para el control y programación de la producción en el sector agrícola. Fue presentada en el año 2011 a la Universidad Nacional de Colombia por Sandra Vélez como tesis de maestría en ingeniería industrial. Utiliza la plataforma Java 2 Micro Edition (J2ME), especial para dispositivos con recursos restringidos, como los teléfonos móviles de gama baja y media, esta aplicación fue pensada para que los agricultores pudieran llevar un control sobre su producción y además administrar la información de sus cultivos,

todo esto desde su teléfono móvil para llevar un mejor control de sus registros (Veléz Quiroga, 2011).



Ilustración 6 Hecopaun, herramienta móvil para el control de producción en el sector agrícola

5.2.2 AgroWin

Es un sistema diseñado para ayudar al empresario agrícola en la gestión, planeación y seguimiento de la empresa, fue creado por InSoft SAS. Entre los principales servicios que ofrece el programa se encuentran el de llevar un control de los contratos de labores y mano de obra, control de ventas y producción, manejo de inventarios y facturación, llevar registros para múltiples fincas y lotes, y generar informes agrícolas de ciclos de producción e informes financieros (InSoft, 2017).

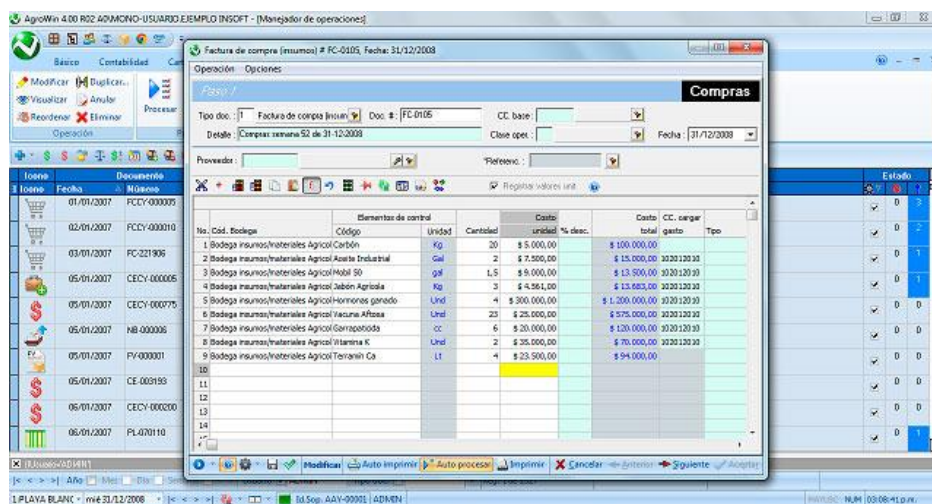


Ilustración 7 AgroWin, software de administración y control para el agro

5.2.3 SynAgro

Es un software agropecuario, que busca ayudar a las empresas agropecuarias a organizar los procesos de toma de datos y análisis de la información, este integra la información Técnica, Económica, Financiera, Administrativa, Contable e Impositiva. Esta herramienta nace en Tucumán, argentina en el año 1993 con el objetivo de desarrollar un programa informático específico para el agro (SynAgro, 2016).

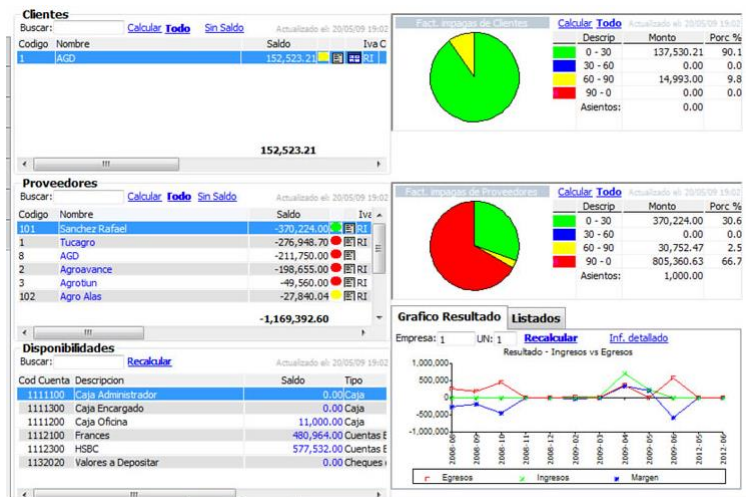


Ilustración 8 SynAgro, software agropecuario

5.2.4 Agroptima

Es un software agropecuario que permite anotar las labores agrícolas diarias y de esta manera lograr que el agricultor logre conocer sus costos. Esta herramienta no solo funciona en computadores sino también en dispositivos móviles, lo cual representa una gran ventaja para el agricultor a la hora de llevar sus registros, pues este puede hacerlas desde cualquier lugar. Agroptima proporciona una forma muy fácil e intuitiva para controlar los costos de producción agrícola y así asegurar un mejoramiento en los márgenes de ganancia (Agroptima, 2017).



Ilustración 9 Agroptima, software especializado en registro de labores agrícolas

A continuación se presenta una tabla comparativa donde describiremos las principales cualidades que poseen las aplicaciones antes mencionadas:

Tabla 2 Tabla comparativa de aplicaciones

Aplicación	Características web	Características móviles	Módulos disponibles	Costo
Hecopaun	Es una aplicación pensada para teléfonos con recursos restringidos, por lo tanto no posee ninguna característica web.	Es una aplicación que solo funciona en teléfonos de gama baja y media.	Módulo de programación y control de producción y administración de información en la fase de cultivo agrícola.	No tiene ningún costo.
AgroWin	Es una aplicación de escritorio, por lo tanto no posee ninguna característica web.	Es una aplicación que no posee características móviles, solo es de escritorio.	Módulo de Costos de producción y manufactura. Módulo de Actividades y labores y cronogramas de ejecución. Módulo de Seguimiento y control de Materiales e insumos. Módulo de Control total de las transacciones con clientes y proveedores. Módulo de Contabilidad agrícola. Módulo de Cartera y proveedores. Módulo de Indicadores, gráficos de gestión y mapas personalizables	Costos dependiendo de los módulos adquiridos que van desde \$3.385.000 + IVA, hasta \$6.665.000 + IVA
SynAgro	Es una aplicación de escritorio, por lo tanto no posee ninguna característica web.	Es una aplicación que no posee características móviles, solo es de escritorio.	Módulo contable que ofrece gestión comercial, financiera e información contable. Modulo productivo que ofrece gestión de información económica, técnica y financiera.	Costos dependiendo de los módulos adquiridos que van desde \$2.500.000 + IVA, hasta \$4.600.000 + IVA
Agroptima	Es una aplicación web, lo que quiere decir que puede ser accedida desde cualquier lugar con un navegador y conexión a internet.	Es una aplicación que funciona totalmente en dispositivos móviles o Smartphone con acceso a internet.	Módulo de anotación y consulta de labores. Módulo de anotación de ingresos y gastos. Módulo de costos y rendimiento por lote, cultivo, hectárea. Módulo de control de costos de maquinaria.	Costos dependiendo de los módulos adquiridos que van desde 249 USD/año hasta 498 USD/año, y se pueden añadir hasta 4 trabajadores extra por un costo de 100 USD extra/año

De la anterior tabla comparativa, podemos observar que de ellas solo una posee características web, esto es una desventaja para las otras, ya que la idea es que los agricultores puedan acceder a sus datos desde cualquier lugar. También podemos observar que de ellas solo dos poseen características móviles, la cual debe ser indispensable si se piensa en que los agricultores puedan registrar su información desde cualquier lugar o dispositivo. Todas las herramientas anteriores poseen módulos para la gestión de producción y aunque algunos ofrecen módulos de más como manejo de inventario o control de maquinaria, todos cumplen con gestionar la información de costos. La gran mayoría de las aplicaciones tienen costos que podrían parecer altos para los pequeños y medianos agricultores, a pesar de que hay una que no tiene ningún costo que es HecoPaun, pero esta tiene una desventaja tecnológica ya que solo está pensada para teléfonos celulares de gamas baja y media del año 2011, no está adecuada a las tecnologías que tenemos hoy en día.

Por estas razones, se observó la oportunidad de crear una aplicación web/móvil que pueda ser accedida desde cualquier lugar, que posea características móviles y que tengan los módulos básicos para la gestión de producción agrícola. Además de gestionar la información de los agricultores que forman una red, para así permitir que estos puedan conocer cómo se está comportando su entorno durante este proceso de producción. Por último resaltar que esta oportunidad se ve apoyada por que podría representar menores costos a los agricultores, pues como observábamos anteriormente, las aplicaciones existentes tienen costos que podrían parecer altos para los pequeños y medianos agricultores.

6 Metodología de desarrollo

En esta sección se realiza una descripción de la metodología de desarrollo de software describiendo los artefactos y herramientas de gestión que se usaron para el desarrollo de este proyecto, así como también las prácticas y tecnologías que se utilizaron.

6.1 Etapa de análisis

Para garantizar el éxito de un proyecto de desarrollo de software, es de suma importancia que antes de empezar a codificar e implementar el software, se lleve a cabo una correcta planificación del proyecto y se haga la identificación de los requerimientos que debe cumplir el mismo. A este proceso se le denomina etapa de análisis. Durante esta etapa, se debe hacer la planificación del proyecto seleccionando la metodología y las herramientas sobre las que se va a apoyar este proceso, así como la correcta identificación de las funcionalidades del sistema y criterios de calidad.

6.1.1 Artefactos y herramienta de planeación del proyecto

Durante la etapa de análisis de la aplicación, se tomó la decisión de utilizar la metodología de desarrollo de software ágil **Scrum** debido a que es un marco de trabajo que ofrece diferentes beneficios para quienes lo utilizan, como lo es llevar una gestión regular de las expectativas del cliente por medio de una lista de requerimientos priorizada, además ofrece la posibilidad de tener una mejor capacidad de flexibilidad y adaptación debido a que la planificación de iteraciones son cortas. Por otra parte, permite tener una mejor mitigación de riesgos ya que plantea un desarrollo iterativo e incremental y ofrece una mayor productividad y calidad con la mejora continua lograda con las retroalimentaciones después de cada iteración. Adicionalmente se tomó la decisión de usar la aplicación en línea **Zube**, la cual permite llevar una gestión y planificación de proyectos de software que se rigen bajo metodología ágil **Scrum**.

Esta metodología implementa el concepto de **Sprint**, que es un bloque de tiempo de un mes o menos durante el cual se crea un incremento de producto “Terminado”, utilizable y potencialmente desplegable. Cada nuevo Sprint comienza inmediatamente después de la finalización del Sprint previo.

Uno de los artefactos usados fueron las **historias de usuario**, que son la forma de especificar los requerimientos funcionales en esta metodología. Las **historias de usuario** se detallan con un título, una descripción, una prioridad, una cantidad de puntos funcionales, cuáles son los datos necesarios para poder operar y también sus criterios de aceptación.

La Ilustración 10 muestra una **historia de usuario** para crear un usuario administrador definida en **Zube**, esta cuenta con las características antes descritas.

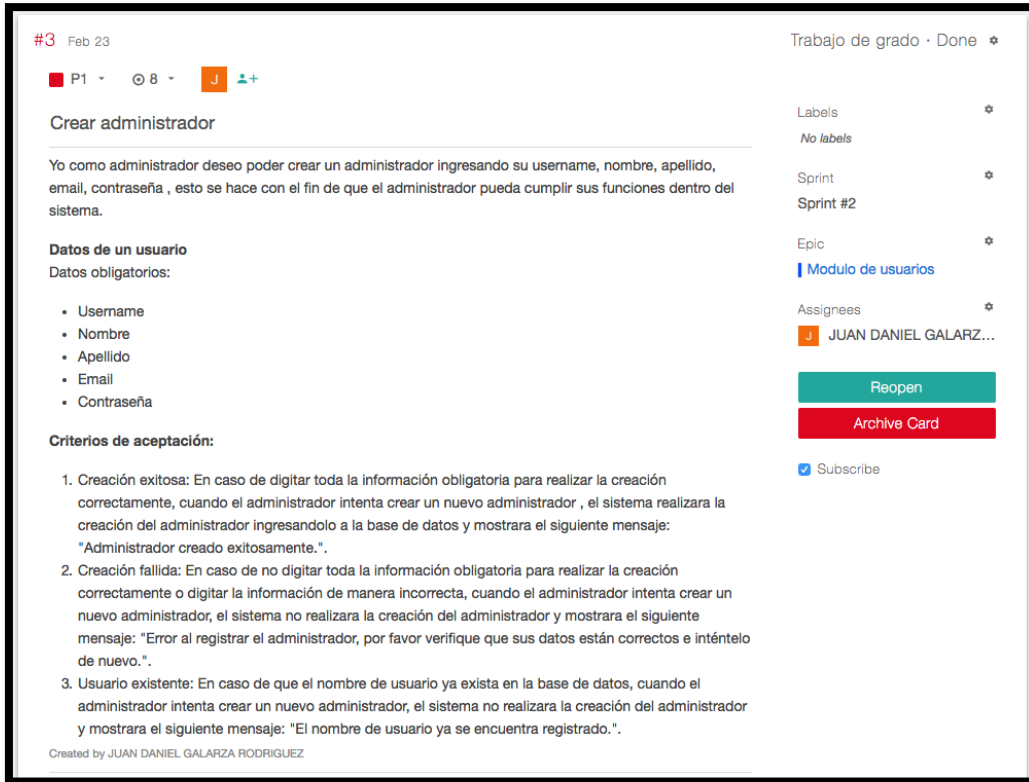


Ilustración 10 Historia de usuario definida en Zube

La metodología seleccionada proporciona otros artefactos de análisis que se utilizaron para este proyecto, como es el caso del **Product Backlog**, el cual es una lista ordenada y priorizada de las historias de usuario definidas y necesarias para el desarrollo del proyecto. La Ilustración 11 se muestra el **Product Backlog** definido en **Zube**, se puede observar las historias de usuario con su título, a qué modulo pertenecen, el responsable de desarrollar esa historia, entre otras características.

La Ilustración 12 muestra los dos primeros **Sprints** definidos en **Zube**, estos tienen su nombre, fecha de inicio, fecha de terminación, cuántas historias de usuario se asignaron para este periodo de tiempo y su completitud, así como también los puntos funcionales asignados y su progreso de desarrollo.

Además, también se usó el artefacto conocido como **Sprint Backlog**, que es un plan detallado de las **historias de usuario** que se van a desarrollar durante un **Sprint**. La Ilustración 13 muestra un **Sprint Backlog** definido en **Zube**, este se puede ver como un **Tablero de Kanban** en donde las historias pueden estar en diferentes estados.

Además de los artefactos mencionados anteriormente también se usó el Incremento de Producto, que es el resultado de cada **Sprint**. Donde cada incremento es una mejora del producto que se encuentre en estado operativo.

Trabajo de grado

Search by title or number New Card

Filters

- Open & Closed
- Assignee
- Creator
- Epic
- Label
- Priority
- Status
- Sprint
- Category

Export JSON

ID	Title	Progress	Assignee	Created	Updated	Closed
#1	Ingreso al sistema	100%	JUAN DANIEL GALARZA RODRIGUEZ	Feb 23	Updated: Apr 3	Closed: Apr 3
#2	Salida del sistema	100%	JUAN DANIEL GALARZA RODRIGUEZ	Feb 23	Updated: Apr 3	Closed: Apr 3
#3	Crear administrador	100%	JUAN DANIEL GALARZA RODRIGUEZ	Feb 23	Updated: Apr 17	Closed: Apr 17
#4	Modificar administrador	100%	JUAN DANIEL GALARZA RODRIGUEZ	Feb 23	Updated: Apr 17	Closed: Apr 17
#5	Consultar administradores	100%	JUAN DANIEL GALARZA RODRIGUEZ	Feb 23	Updated: Apr 17	Closed: Apr 17
#6	Deshabilitar administrador	100%	JUAN DANIEL GALARZA RODRIGUEZ	Feb 23	Updated: Apr 17	Closed: Apr 17
#7	Crear agricultor admin	100%	JUAN DANIEL GALARZA RODRIGUEZ	Feb 26	Updated: Apr 3	Closed: Apr 3
#8	Crear agricultor usuario publico	100%	JUAN DANIEL GALARZA RODRIGUEZ	Feb 26	Updated: Apr 3	Closed: Apr 3
#9	Modificar agricultor admin	100%	JUAN DANIEL GALARZA RODRIGUEZ	Feb 26	Updated: Apr 3	Closed: Apr 3
#10	Modificar perfil agricultor	100%	JUAN DANIEL GALARZA RODRIGUEZ	Feb 26	Updated: Apr 3	Closed: Apr 3
#11	Consultar agricultores	100%	JUAN DANIEL GALARZA RODRIGUEZ	Feb 26	Updated: Apr 3	Closed: Apr 3
#12	Cambiar estado agricultor	100%	JUAN DANIEL GALARZA RODRIGUEZ	Feb 26	Updated: Apr 3	Closed: Apr 3
#13	Crear Lote	100%	JUAN DANIEL GALARZA RODRIGUEZ	Feb 26	Updated: May 16	Closed: May 16
#14	Modificar lote	100%	JUAN DANIEL GALARZA RODRIGUEZ	Feb 26	Updated: May 16	Closed: May 16
#15	Consultar lotes	100%	JUAN DANIEL GALARZA RODRIGUEZ	Feb 26	Updated: May 16	Closed: May 16
#16	Filtrar lotes	100%	JUAN DANIEL GALARZA RODRIGUEZ	Feb 26	Updated: May 16	Closed: May 16
#17	Crear Finca	100%	JUAN DANIEL GALARZA RODRIGUEZ	Feb 26	Updated: Apr 23	Closed: Apr 23
#18	Modificar finca	100%	JUAN DANIEL GALARZA RODRIGUEZ	Feb 26	Updated: Apr 23	Closed: Apr 23
#19	Consultar fincas	100%	JUAN DANIEL GALARZA RODRIGUEZ	Feb 26	Updated: Apr 23	Closed: Apr 23
#20	Filtrar fincas	100%	JUAN DANIEL GALARZA RODRIGUEZ	Feb 26	Updated: Apr 23	Closed: Apr 23
#21	Filtrar usuarios	100%	JUAN DANIEL GALARZA RODRIGUEZ	Feb 26	Updated: Apr 17	Closed: Apr 17
#22	Filtrar agricultores	100%	JUAN DANIEL GALARZA RODRIGUEZ	Feb 26	Updated: Apr 3	Closed: Apr 3
#23	Asignar ubicacion geografica de una finca	100%	JUAN DANIEL GALARZA RODRIGUEZ	Feb 26	Updated: Apr 23	Closed: Apr 23
#24	Modificar ubicacion geografica de una finca	100%	JUAN DANIEL GALARZA RODRIGUEZ	Feb 26	Updated: Apr 23	Closed: Apr 23

Ilustración 11 Product Backlog definido en Zube

Sprints

OPEN CLOSED New Sprint

Sort Start date

Sprint #1

Start date March 12, 2018 **Past Due: April 1, 2018** Updated 6 months ago

In Progress Changes

22 Cards added 18 Cards removed

Timeline

Points: Open 0 Closed 59 Complete 100%

Cards: Open 0 Closed 9 Complete 100%

Edit Close Delete

Sprint #2

Start date April 2, 2018 **Past Due: April 15, 2018** Updated 6 months ago

Timeline

Points: Open 0 Closed 28 Complete 100%

Cards: Open 0 Closed 6 Complete 100%

Edit Close Delete

Ilustración 12 Sprints definidos en Zube

Scrum también propone algunas actividades principales que se deben llevar a cabo durante el desarrollo del proyecto. De estas actividades se realizaron: el refinamiento del **Backlog**, la planificación del **Sprint**, la revisión del **Sprint** y la Retrospectiva del **Sprint**. No se llevó a cabo la actividad de Scrum diario debido a que no era posible tener reuniones diarias con las personas que apoyaron el desarrollo del proyecto.

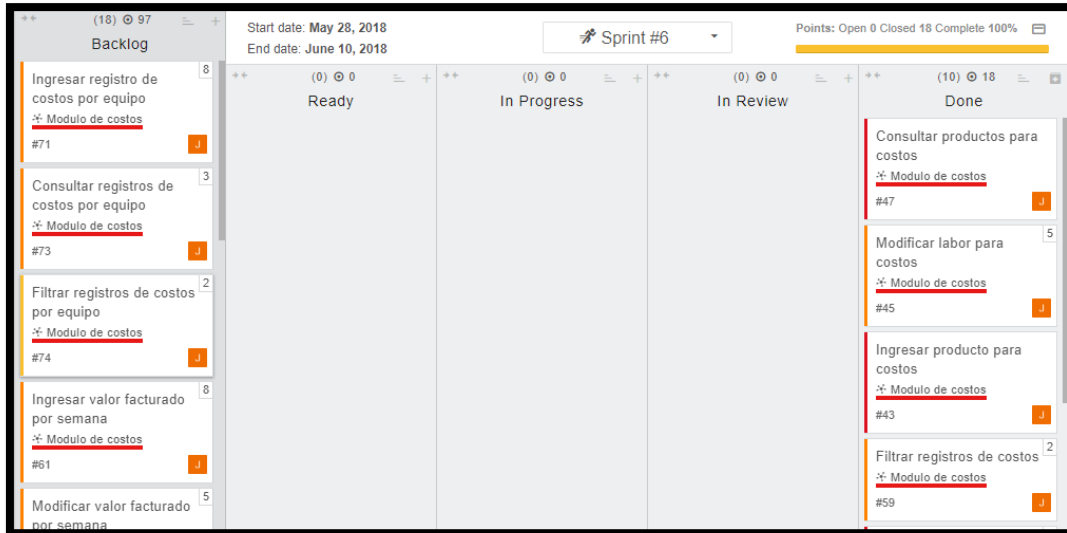


Ilustración 13 Sprint Backlog definido en Zube
Fuente: Elaboración propia

En conclusión, la documentación de análisis se alojó dentro de la herramienta **Zube** donde se definieron las historias de usuario, el **Product Backlog**, **Sprints**, y los **Sprint Backlog** que se iban a usar para el desarrollo de esta aplicación. En total se definieron 120 historias de usuario y 13 **Sprints** para ser desarrollados durante la duración del proyecto.

6.1.2 Herramientas de ambiente de desarrollo de software

Durante el desarrollo de este proyecto se seleccionaron algunas herramientas de ambiente de desarrollo de software, para ser utilizadas durante todo el proceso. Entre ellas tenemos **AWS Cloud9**, que es un **entorno de desarrollo integrado (IDE)** basado en la nube, que permite escribir, ejecutar y depurar código con solo un navegador web. Incluye un editor de código, así como un depurador y un terminal. Además este IDE incluye herramientas esenciales para los lenguajes de programación más conocidos, como **JavaScript**, **Python**, **PHP**, entre otros, por lo que no debe instalar archivos ni configurar su máquina de desarrollo para empezar nuevos proyectos. Ya que el IDE de **AWS Cloud9** se basa en la nube, puede trabajar en sus proyectos desde la oficina, desde su casa o desde cualquier sitio con conexión a Internet (Amazon Web Services, 2018). Esta herramienta fue utilizada para llevar a cabo el desarrollo de esta aplicación web. La Ilustración 14 muestra como es gráficamente el entorno de desarrollo integrado de **AWS Cloud9**.

También se usó el entorno de desarrollo integrado (IDE) **NetBeans**, que permite desarrollar rápida y fácilmente aplicaciones de escritorio, móviles y web de **Java** (Oracle Corporation, 2018b). Se utilizó esta herramienta para la implementación de un módulo de biometría programado en Lenguaje **Java** para llevar el control del ingreso de los trabajadores que tenían labores asignadas en las diferentes fincas. La Ilustración 15 muestra como es gráficamente el entorno de desarrollo integrado **NetBeans**.

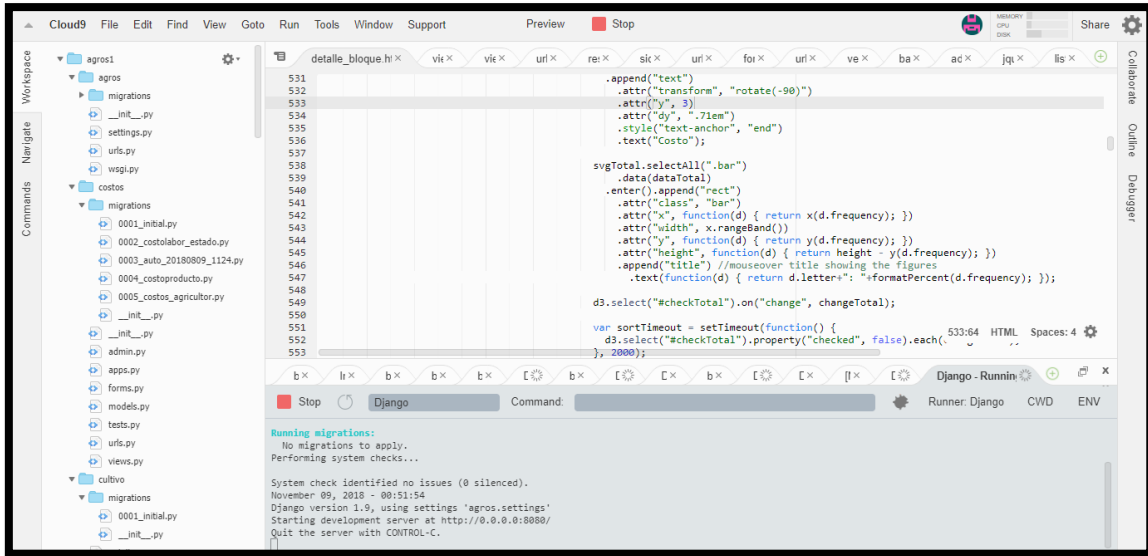


Ilustración 14 Entorno de desarrollo integrado AWS Cloud9
Fuente: Elaboración propia

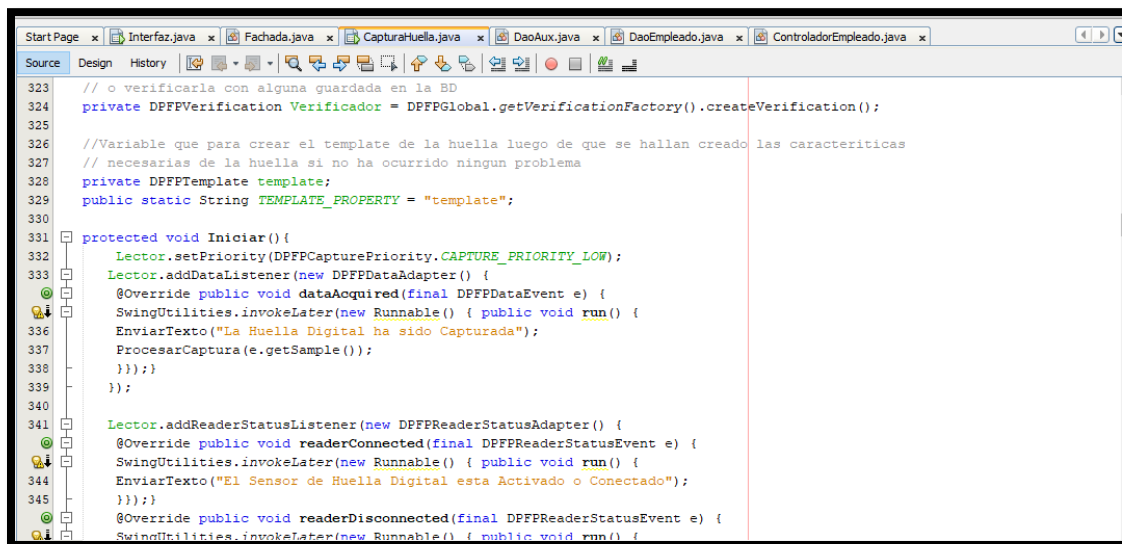


Ilustración 15 Entorno de desarrollo integrado NetBeans
Fuente: Elaboración propia

Para tener acceso a la base de datos de **Postgresql** se utilizó la herramienta **pgAdmin**, la cual es una plataforma de desarrollo y administración de código abierto más popular y rica en funciones para **Postgresql** (pgadmin.org, 2018). Esta herramienta fue utilizada para configurar y administrar la base de datos que se usó para el desarrollo de esta aplicación. La Ilustración 16 muestra la interfaz principal de la plataforma de desarrollo **pgAdmin**.

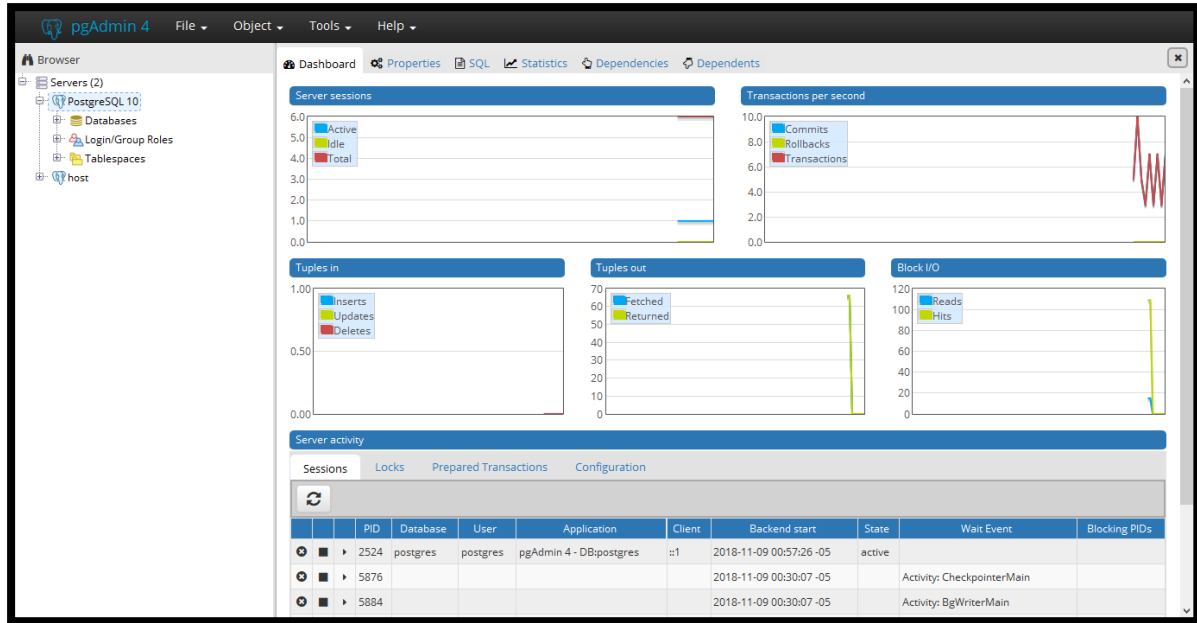


Ilustración 16 pgAdmin, plataforma de desarrollo para PostgreSQL
Fuente: Elaboración propia

Para la implementación de las pruebas funcionales se utilizó **Selenium IDE**, esta herramienta es un entorno de desarrollo integrado para scripts de **Selenium** que permite grabar y reproducir pruebas funcionales sin necesidad de usar un lenguaje de scripting para pruebas. **Selenium** incorpora un lenguaje específico de dominio para pruebas llamado **Selane** que se usa para escribir pruebas en un amplio número de lenguajes de programación (seleniumhq.org, 2018). En la Ilustración 17 Se puede observar un test suite definido para probar los casos de prueba para la funcionalidad de registrar un agricultor, en la parte izquierda podemos encontrar todos los casos de prueba grabados para esta funcionalidad y en la parte derecha los pasos que va a seguir la prueba para esperar un comportamiento deseado.

Durante el proceso de desarrollo de esta aplicación, se usó la herramienta **GitHub**, esta es una plataforma de desarrollo colaborativo para alojar proyectos de software utilizando el sistema de control de versiones **Git**. En la Ilustración 18 se puede observar el repositorio de este proyecto creado en **GitHub**.

El modelo de la base de datos fue realizado en la herramienta web **Vertabelo**, esta permite realizar el diseño de la base de datos de manera visual representando el modelo en forma relacional (Vertabelo, 2016). En la Ilustración 19 se muestra la interfaz gráfica de la aplicación web **Vertabelo** en donde se está haciendo el diseño del modelo de datos para el desarrollo de este proyecto.

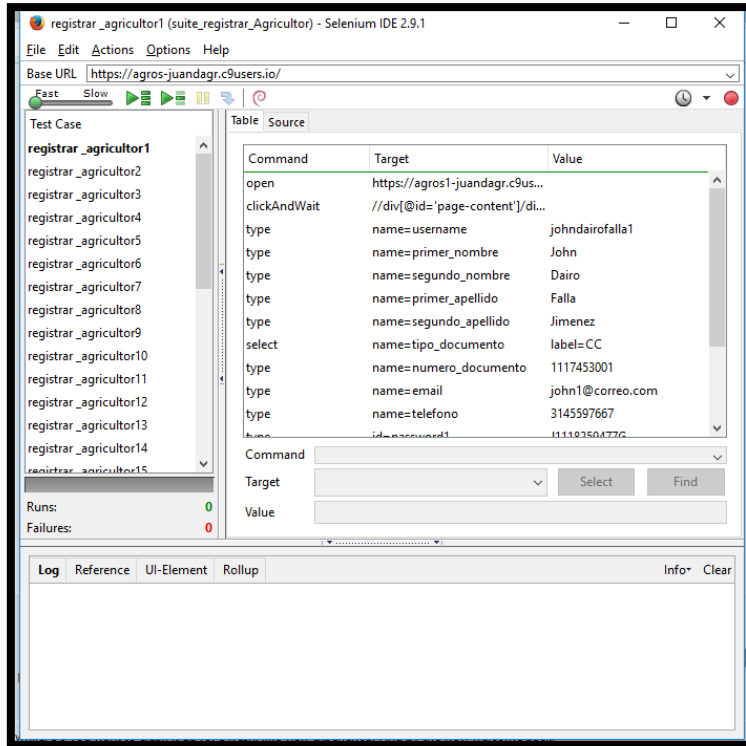


Ilustración 17 Selenium IDE
Fuente: Elaboración propia

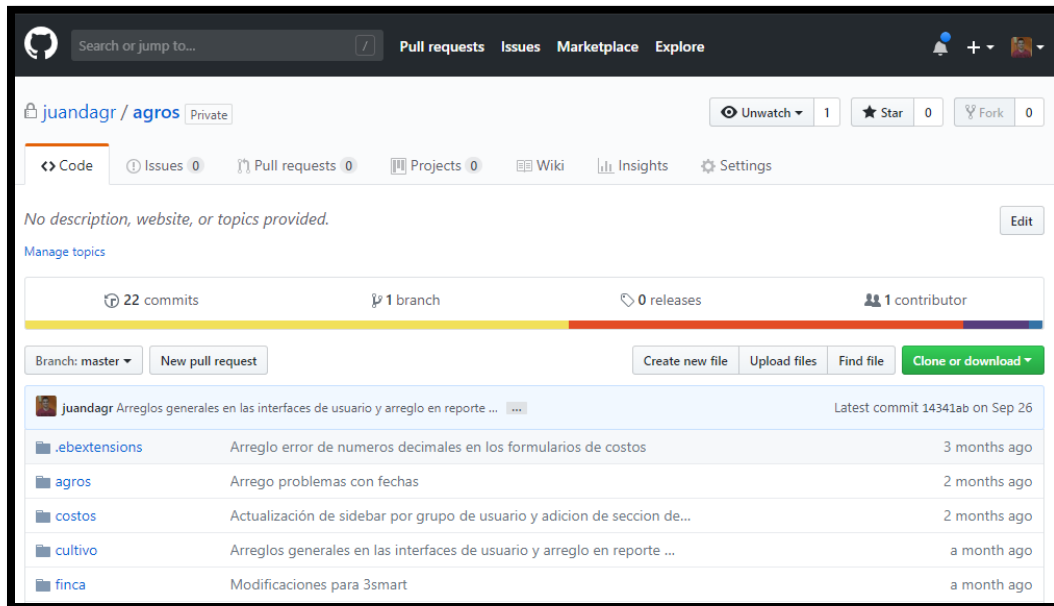


Ilustración 18 Repositorio agros en GitHub
Fuente: Elaboración propia

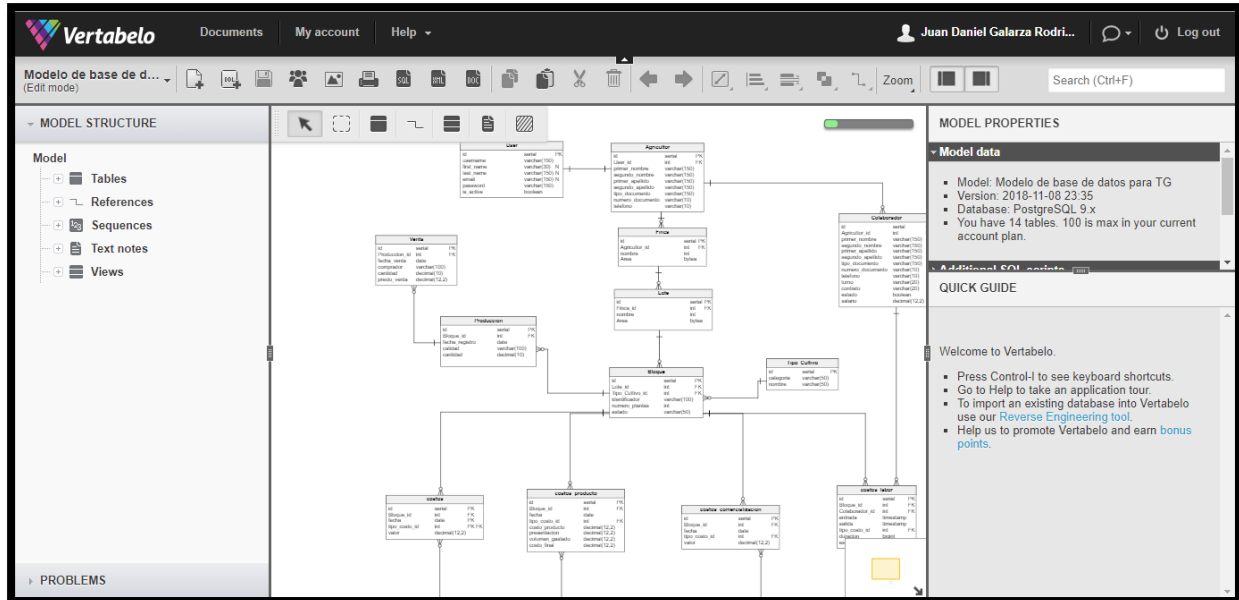


Ilustración 19 Vertabelo, herramienta para diseño de la base de datos
Fuente: Elaboración propia

6.2 Etapa de diseño

En la etapa de diseño de la aplicación se especificó la arquitectura del software, qué tecnologías se usarían para el funcionamiento de la aplicación y también se definió el modelo de datos (base de datos relacional).

6.2.1 Arquitectura del software y tecnologías utilizadas

Cuando se usa el **Framework Django** como en el caso de esta aplicación, hay que tener en cuenta que este es conocido por usar el patrón de diseño **Modelo-Plantilla-Vista (MTV, por sus siglas en Inglés)** (Adrian Holovaty, 2008). La Ilustración 20 muestra el funcionamiento del patrón de diseño **MVT**, donde la interacción parte desde una solicitud que se hace sobre el navegador web al servidor, esta solicitud es manejada por una vista definida en **Django**, a su vez, la vista interactúa con el modelo definido en **Django** para obtener los datos necesarios para atender la solicitud, el modelo consulta una base de datos persistente para hacer la consulta de los datos. Posteriormente, cuando la vista ha obtenido los datos, se los envía a una plantilla definida usando **Django** y por último la plantilla genera la interfaz gráfica con la respuesta de la solicitud enviada inicialmente y se la envía al navegador web.

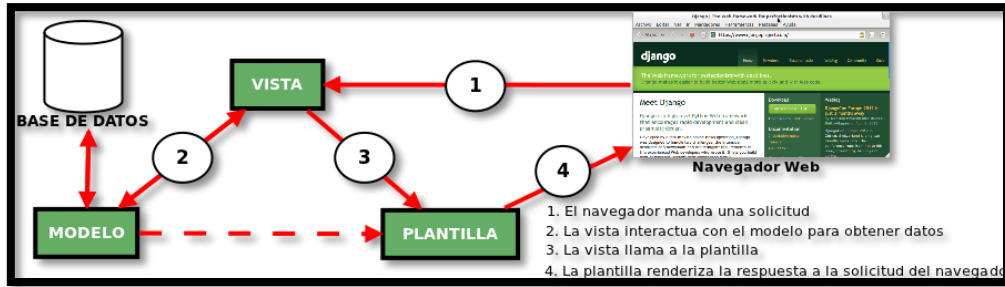


Ilustración 20 Arquitectura Modelo-Plantilla-Vista
Fuente: <http://www.maestrosdelweb.com>

En el patrón de diseño MTV, el **Modelo** hace referencia a la capa de acceso de datos, para este proyecto, esta capa se apoya en el sistema de gestión de bases de datos relacionales llamado *PostgreSQL* y su extensión *PostGIS* para la gestión de datos geográficos. La **Plantilla** es la capa de presentación, esta se relaciona con la interacción entre el usuario y la aplicación, así como que datos se van a mostrar y cómo hacerlo, en este proyecto esta capa está apoyada en las tecnologías de Front-End como *HTML* para el desarrollo de las interfaces gráficas, *CSS* para los estilos, además de *JavaScript* y *jQuery* para programar su funcionamiento. Adicionalmente se usó *Bootstrap* para el manejo de interfaces *responsive* para que estas fueran correctamente visualizadas en diferentes tamaños de pantallas. Otra de las tecnologías usadas fue *D3*, esta es una librería definida para la visualización de datos y generar reportes gráficos que puedan tener interacción con el usuario. Por último, La **Vista** es la capa de la lógica de negocios, esta capa contiene la lógica que accede al modelo y la delega a la plantilla apropiada, se puede pensar en esto como un puente entre los modelos y las plantillas. Para este proyecto esta capa se apoya sobre el *framework Django*, que esta soportado por el lenguaje de programación *Python*.

Adicionalmente se usó la tecnología *One Touch SDK*, que es una herramienta de desarrollo de software para integrar la biométrica de huellas dactilares en un amplio conjunto de aplicaciones, servicios y productos basados en Java. Esta se usó con el fin de realizar operaciones biométricas para controlar el ingreso de los trabajadores que tenían labores asignadas en las diferentes fincas. La Ilustración 21 muestra las tecnologías sobre las cuales se apoyó cada capa definida en el patrón de diseño seleccionado para el desarrollo de esta aplicación.

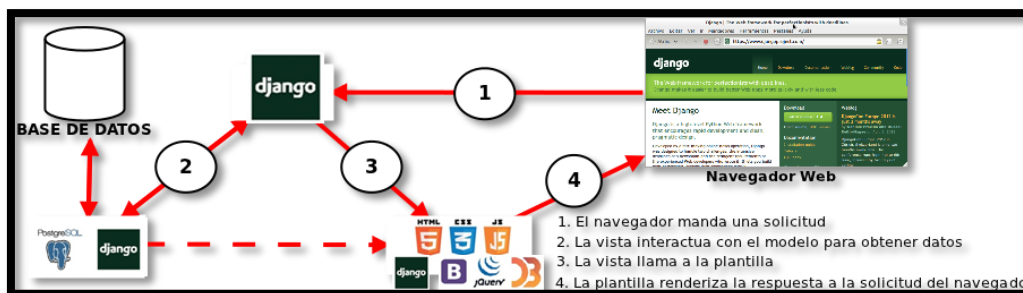


Ilustración 21 Tecnologías usadas por capa para el desarrollo de la aplicación Fuente: Elaboración propia

6.2.2 Modelo de datos

El modelo de datos determina la estructura lógica de la persistencia que en el caso de esta aplicación es una base de datos relacional. Esta estructura lógica determina el modo de almacenar, organizar y manipular los datos persistentes. Este modelo de datos se representa usando un modelo entidad-relación, ya que permite representar las entidades relevantes para la base de datos así como sus interacciones y propiedades.

En la Ilustración 22 se presenta una parte del diagrama con el modelo de la base de datos utilizado para el desarrollo de este proyecto. Para una completa visualización del modelo se puede observar en el archivo adjunto “Modelo de base de datos.png” En el diagrama podemos encontrar la entidad agricultor, que en la aplicación extiende de la entidad User de *Django*, un agricultor puede tener varios colaboradores y una o más fincas, estas fincas pueden tener uno o más lotes y los lotes pueden tener uno o más bloques. Los bloques tienen asignado un tipo de cultivo y pueden tener registros de producción, los cuales tienen asociados diferentes ventas debido a que toda la producción no se vende el mismo día, ni al mismo comprador. Además de los registros de producción, los bloques también pueden tener registros de diferentes tipos de costos, entre ellos tenemos los costos por producto, costos por labor, costos de comercialización y otros costos. Para los costos por labor es necesario asignar uno de los colaboradores que el agricultor tiene asociados, para determinar quién realizó dicha labor.

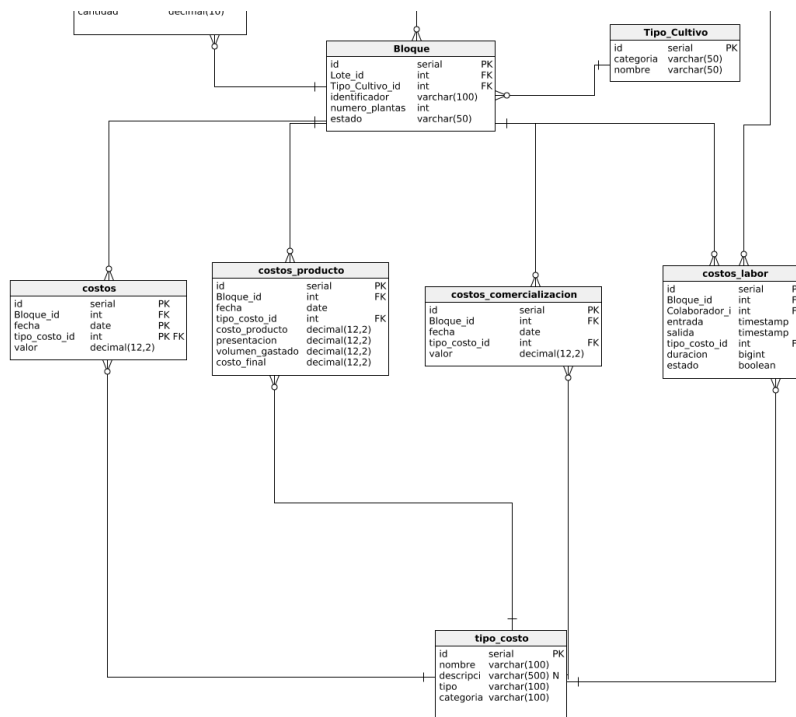


Ilustración 22 Modelo de la base de datos
Fuente: Elaboración propia

7 Desarrollo de un módulo para la gestión de usuarios y permisos

En esta sección del documento se presentan las etapas del proceso de desarrollo para el módulo de gestión de usuarios y permisos. Entre estas etapas encontramos el análisis, diseño, implementación y pruebas.

7.1 Análisis

Para el desarrollo del módulo de gestión de usuarios y permisos, lo primero que se realizó fue la caracterización de los usuarios que iban a tener interacción con la aplicación. La aplicación está diseñada para tener como usuario a cualquier persona que se desempeñe en el sector agrícola, preferiblemente con conocimientos básicos en computación, para hacer más fácil el reconocimiento del sistema. Es importante aclarar que para hacer uso de la aplicación los usuarios deben ser capacitados para que puedan adquirir los conocimientos y habilidades básicas que les permitan hacer un correcto uso de ella.

En esta aplicación se identificaron los siguientes tipos de usuario: El administrador, el agricultor y colaboradores (personas que realizan el trabajo de campo).

- **Administrador:** Son los usuarios responsables de configurar, mantener y asegurar el correcto funcionamiento de la aplicación. A estos usuarios se les permite llevar la gestión de otros usuarios, tanto administradores como agricultores, así como también hacer los registros de la información básica necesaria para que los agricultores registren sus costos.
- **Agricultor:** Son los usuarios que tendrán la posibilidad de llevar a cabo el seguimiento de planeación y proceso de producción agrícola por medio de la aplicación. A estos usuarios se les permite registrar sus fincas con sus respectivos lotes y bloques, también pueden hacer registros de costos y producción, y también pueden observar su información organizada y observar reportes que le permitan tener una visión crítica acerca del proceso que se lleva en sus terrenos.
- **Colaborador:** Son los usuarios que usan las funcionalidades de biometría para llevar un control del inicio y final de las labores que tienen asignadas en las diferentes fincas.

En la Tabla 3 se presentan las funcionalidades que cada tipo de usuario podrá realizar dentro del módulo de gestión de usuarios y permisos.

Tabla 3 Funcionalidades por tipo de usuario para el módulo de gestión de usuarios y permisos

Funcionalidad	Administrador	Agricultor	Colaborador
Ingresar al sistema	X	X	
Salir del sistema	X	X	
Crear administrador	X		
Modificar administrador	X		
Consultar administradores	X		
Crear agricultor administrador	X		
Modificar un agricultor administrador	X		
Consultar agricultores	X		
Cambiar estado agricultor	X		
Modificar perfil agricultor		X	
Crear agricultor usuario publico		X	
Filtrar agricultores	X		
Reestablecer contraseña	X	X	
Deshabilitar administrador	X		
Filtrar usuarios	X		

Una vez identificados los usuarios y cuál es su interacción con la aplicación, se procedió a definir las historias de usuario para este módulo, en total se definieron 15 historias de usuario, planificadas para llevarse a cabo a lo largo de 2 Sprints. En la Ilustración 23 se muestran las historias de usuario definidas en **Zube** con sus respectivos puntos funcionales y el sprint al que pertenecen.

7.2 Implementación

A continuación se presentan las capturas de pantalla de algunas de las funcionalidades implementadas en este módulo. La Ilustración 24 muestra la interfaz gráfica de la funcionalidad de crear un agricultor, en la ilustración podemos observar un formulario con los datos de entrada que se necesitan para hacer el registro, también están los botones para cancelar y confirmar el registro.

<input type="checkbox"/>	#12	Cambiar estado agricultor	2	Done	J	Modulo de usuarios	Sprint #1
<input type="checkbox"/>	#9	Modificar agricultor admin	5	Done	J	Modulo de usuarios	Sprint #1
<input type="checkbox"/>	#10	Modificar perfil agricultor	5	Done	J	Modulo de usuarios	Sprint #1
<input type="checkbox"/>	#8	Crear agricultor usuario publico	13	Done	J	Modulo de usuarios	Sprint #1
<input type="checkbox"/>	#11	Consultar agricultores	3	Done	J	Modulo de usuarios	Sprint #1
<input type="checkbox"/>	#7	Crear agricultor admin	13	Done	J	Modulo de usuarios	Sprint #1
<input type="checkbox"/>	#2	Salida del sistema	3	Done	J	Modulo de usuarios	Sprint #1
<input type="checkbox"/>	#1	Ingreso al sistema	13	Done	J	Modulo de usuarios	Sprint #1
<input type="checkbox"/>	#22	Filtrar agricultores	2	Done	J	Modulo de usuarios	Sprint #1
<input type="checkbox"/>	#27	Reestablecer contraseña	8	Done	J	Modulo de usuarios	Sprint #2
<input type="checkbox"/>	#6	Deshabilitar administrador	2	Done	J	Modulo de usuarios	Sprint #2
<input type="checkbox"/>	#21	Filtrar usuarios	2	Done	J	Modulo de usuarios	Sprint #2
<input type="checkbox"/>	#5	Consultar administradores	3	Done	J	Modulo de usuarios	Sprint #2
<input type="checkbox"/>	#4	Modificar administrador	5	Done	J	Modulo de usuarios	Sprint #2
<input type="checkbox"/>	#3	Crear administrador	8	Done	J	Modulo de usuarios	Sprint #2

Ilustración 23 Historias de usuario del módulo de gestión de usuarios y permisos
Fuente: Elaboración propia

La Ilustración 25 muestra la interfaz gráfica de la funcionalidad de modificar un agricultor, en la ilustración podemos observar un formulario con los datos que se pueden modificar en el registro previamente creado, también están los botones para cancelar y confirmar la modificación.

La Ilustración 26 muestra la interfaz gráfica de la funcionalidad de listar agricultores, en la ilustración podemos observar una tabla con los agricultores que se encuentran registrados en el sistema, estos registros se muestran con los campos de primer nombre, segundo nombre, primer apellido, segundo apellido, documento, teléfono y estado. Además de un botón de acciones por medio del cual se puede seleccionar un agricultor para ser deshabilitado o para modificar sus datos. También hay un botón que redirige al usuario para crear un nuevo agricultor.

FORMULARIO DE REGISTRO

Nombre de usuario	<input type="text"/>
Primer nombre	<input type="text"/>
Segundo nombre	<input type="text"/>
Primer apellido	<input type="text"/>
Segundo apellido	<input type="text"/>
Tipo de documento	----- ▾
Número de documento	<input type="text"/>
Correo electrónico	<input type="text"/>
Número de Telefono	<input type="text"/>
Contraseña	<input type="password"/>
Contraseña (confirmación)	<input type="password"/>

Ilustración 24 Funcionalidad de crear un agricultor
Fuente: Elaboración propia

La Ilustración 27 muestra la interfaz gráfica de la funcionalidad de modificar perfil de agricultor, en la ilustración podemos observar un formulario con los datos personales que puede modificar un agricultor, también se puede observar la pestaña que lleva a la funcionalidad de cambiar la contraseña.

FORMULARIO DE MODIFICACIÓN

Primer nombre	<input type="text" value="Jose"/>
Segundo nombre	<input type="text" value="David"/>
Primer apellido	<input type="text" value="Palma"/>
Segundo apellido	<input type="text"/>
Tipo de documento	CC ▾
Número de documento	<input type="text" value="1113628355"/>
Correo electrónico	<input type="text" value="jdpm07@gmail.com"/>
Número de Telefono	<input type="text" value="3187853551"/>

Ilustración 25 Funcionalidad de modificar agricultor
Fuente: Elaboración propia

10 registros por página

Primer nombre	Segundo nombre	Primer apellido	Segundo apellido	Documento	Telefono	Estado	Acciones
Elier	Fernando	Piamba		1118259325	3159638574	Activo	Acciones
Jose	David	Palma		1113628355	3187853551	Activo	Acciones
Juan	Felipe	Tellez	Acosta	11167873351	3132457765	Activo	Acciones
Juan	Daniel	Galarza	Rodriguez	1118259477	3136432165	Activo	Acciones
Marcela		Guerrero	Urbano	1118259990	3146154710	Activo	Acciones
Nathaly		Meneses	Ayala	1118398476	3156788876	Activo	Acciones
Ruben	Dario	Alavarado	Rivera	1118256503	3136864034	Activo	Acciones
Primer nombre	Segundo nombre	Primer apellido	Segundo apellido	Documento	Telefono	Estado	Acciones


Mostrando desde 1 hasta 7 de un total de 7 registros

< Previous **1** Next >

[REGISTRAR AGRICULTOR](#)

Ilustración 26 Funcionalidad de listar agricultores
Fuente: Elaboración propia

Perfil



Juan Daniel Galarza Rodriguez
Agricultor registrado

Información Personal
Configuración de cuenta

INFORMACIÓN PERSONAL

Primer nombre

Segundo nombre

Primer apellido

Segundo apellido

Tipo de documento

Número de documento

Correo electrónico

Número de Telefono

Ilustración 27 Funcionalidad de modificar perfil agricultor
Fuente: Elaboración propia

7.3 Pruebas

Durante el desarrollo del módulo de Gestión de usuarios y permisos se realizaron pruebas funcionales del sistema. En primer lugar seleccionaron cuales serían las funcionalidades que se iban a probar, pues por cuestiones de tiempo y costo no se pudieron probar todas las funcionalidades para este módulo, se determinó que se le realizarían pruebas a un total de 8 funcionalidades. Una vez identificadas las funcionalidades se llevó a cabo el diseño de las pruebas funcionales que se iban a realizar. Posteriormente, con el apoyo de las clases identificadas se definieron los casos de prueba para la funcionalidad. Después de tener el diseño de todas las funcionalidades, el siguiente paso fue codificarlas y ejecutarlas usando la herramienta **Selenium IDE**. Con las pruebas codificadas, se procedió a agruparlas y realizar la ejecución para analizar el comportamiento de la funcionalidad.

En la Tabla 4 se muestran dos ilustraciones con el diseño y la ejecución de las pruebas funcionales para la funcionalidad de crear agricultor de este módulo. En la Ilustración 28 se muestran algunos casos de prueba diseñados para esta funcionalidad y en la Ilustración 29 se puede observar la ejecución de todos los casos de prueba que fueron diseñados para esta funcionalidad. En la parte izquierda se pueden ver los casos que se ejecutaron, el color verde significa que no fallo la prueba y en caso de que alguna hubiera fallado se hubiera subrayado en color rojo. En la parte derecha se pueden ver los pasos que se deben seguir para ejecutar algún caso de prueba que se haya seleccionado en la parte izquierda. Por último, en la parte inferior se puede ver un resumen de la ejecución de los casos de prueba.

Tabla 4 Diseño y ejecución de pruebas funcionales para el módulo de gestión de usuarios

No de caso	Clase de equivalencia	Contraseña	Confirmar contraseña	Resultado esperado
1	1,4,7,9,12,14,17,20,23,26,29	J1118259477G	J1118259477G	Redirigir al listado de agricultores con el agricultor recién registrado y mostrar un mensaje en un recuadro verde con el siguiente mensaje: "Exito! Agricultor registrado exitosamente."
2	1,4,7,9,12,14,17,20,23,26,29	J1118259477G	J1118259477G	Redirigir al listado de agricultores con el agricultor recién registrado y mostrar un mensaje en un recuadro verde con el siguiente mensaje: "Exito! Agricultor registrado exitosamente."
3	1,4,7,9,12,14,17,20,23,26,29	J1118259477G	J1118259477G	Redirigir al listado de agricultores con el agricultor recién registrado y mostrar un mensaje en un recuadro verde con el siguiente mensaje: "Exito! Agricultor registrado exitosamente."
4	1,4,7,9,12,14,17,20,23,26,29	J1118259477G	J1118259477G	Redirigir al listado de agricultores con el agricultor recién registrado y mostrar un mensaje en un recuadro verde con el siguiente mensaje: "Exito! Agricultor registrado exitosamente."
5	1,4,7,9,12,14,17,20,23,26,29	J1118259477G	J1118259477G	Redirigir al listado de agricultores con el agricultor recién registrado y mostrar un mensaje en un recuadro verde con el siguiente mensaje: "Exito! Agricultor registrado exitosamente."
6	1,4,7,9,12,14,17,20,23,26,29	J1118259477G	J1118259477G	Redirigir al listado de agricultores con el agricultor recién registrado y mostrar un mensaje en un recuadro verde con el siguiente mensaje: "Exito! Agricultor registrado exitosamente."
7	1,4,7,9,12,14,17,20,23,26,29	J1118259477G	J1118259477G	Redirigir al listado de agricultores con el agricultor recién registrado y mostrar un mensaje en un recuadro verde con el siguiente mensaje: "Exito! Agricultor registrado exitosamente."
8	1,4,7,9,12,14,17,20,23,26,29	J1118259477G	J1118259477G	Redirigir al listado de agricultores con el agricultor recién registrado y mostrar un mensaje en un recuadro verde con el siguiente mensaje: "Exito! Agricultor registrado exitosamente."
	1,4,7,9,12,14,17,20,23,26,29	J1118259477G	J1118259477G	Redirigir al listado de agricultores con el agricultor recién registrado y mostrar un mensaje en un recuadro verde con el siguiente mensaje: "Exito! Agricultor registrado exitosamente."

Ilustración 28 Casos de prueba para la funcionalidad crear agricultor

The screenshot shows the Selenium IDE interface. The main window displays a list of test cases for 'registrar_agricultor31'. The 'Table Source' window is open, showing a table with columns for Command, Target, and Value. The log window at the bottom shows the execution steps, including 'Executing: [type] [id=passwdor2] [1,2,3,4]', 'Executing: [clickAndWait] [//button[@type='submit']]', 'Executing: [verifyText] [css=li.parsley-required] [Los dos campos de contraseña no coinciden.]', and 'Executing: [verifyText] [css=div.alert-content] [Error! Error al registrar el agricultor, por favor verifique que sus datos estén correctos e intente de nuevo.]'. The log concludes with 'Test case passed' and 'Test suite completed: 31 played, all passed!'.

Ilustración 29 Ejecución de casos de prueba para la funcionalidad crear agricultor

Con la ejecución de las pruebas finalizada, se creó una matriz de requerimientos de prueba para todas las funcionalidades que se probaron en este módulo, se puede observar en el archivo adjunto “MRP-Modulo de usuarios.pdf”. En la columna *Elementos a probar* de la matriz, se hace una referencia a los valores que se usaron para el diseño de cada caso de prueba, esta tiene el nombre de la funcionalidad seguido de un número, el cual corresponde al número de caso de prueba que se probó en la codificación, esta referencia se hace al documento adjunto “Diseño de pruebas funcionales-Modulo de usuarios.pdf”.

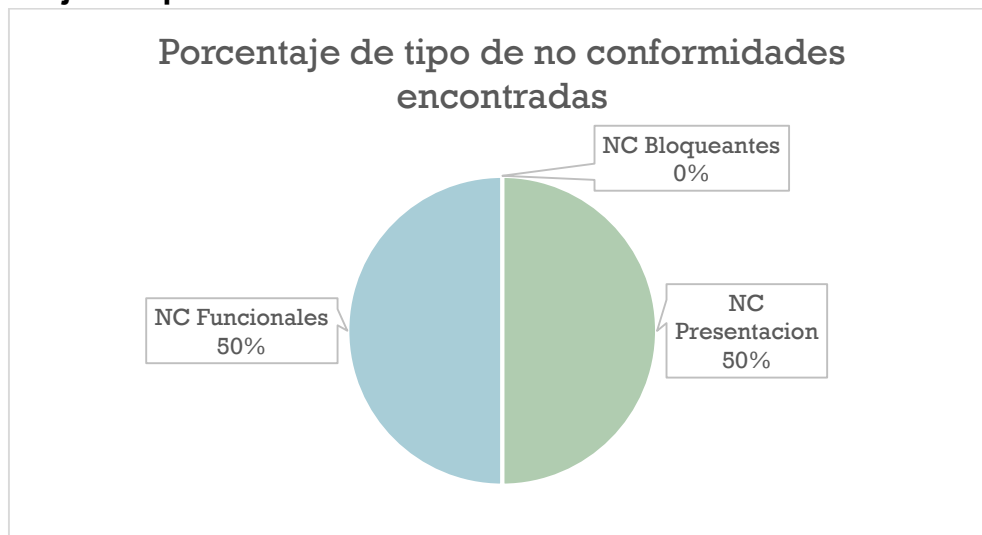
A Continuación en la Tabla 5 se muestra el resumen de ejecución de los casos pruebas que se diseñaron para este módulo, en donde se detalla por cada funcionalidad probada el número de pruebas en las cuales se identificaron no conformidades (separadas por tipo), número de diseños y por último el número de pruebas que fueron ejecutadas. En los encabezados de las columnas se reemplazaron las palabras “no conformidad” por la abreviatura “NC” y la palabra “requerimientos” por “reqs”.

Tabla 5 Resumen de ejecución de casos de prueba para el módulo de gestión de usuarios

Funcionalidad probada	Total NC bloqueantes	Total NC funcionales	Total NC presentación	Total NC	Total reqs de prueba diseñados	Reqs de prueba ejecutados
Ingresar al sistema	0	0	0	0	6	6
Crear agricultor administrador	0	0	0	0	31	31
Modificar un agricultor administrador	0	0	0	0	22	22
Consultar agricultores	0	0	1	1	6	6
Cambiar estado agricultor	0	0	0	0	2	2
Modificar perfil agricultor	0	0	0	0	22	22
Crear agricultor usuario publico	0	0	0	0	31	31
Reestablecer contraseña	0	1	0	1	6	6
Totales	0	1	1	2	126	126

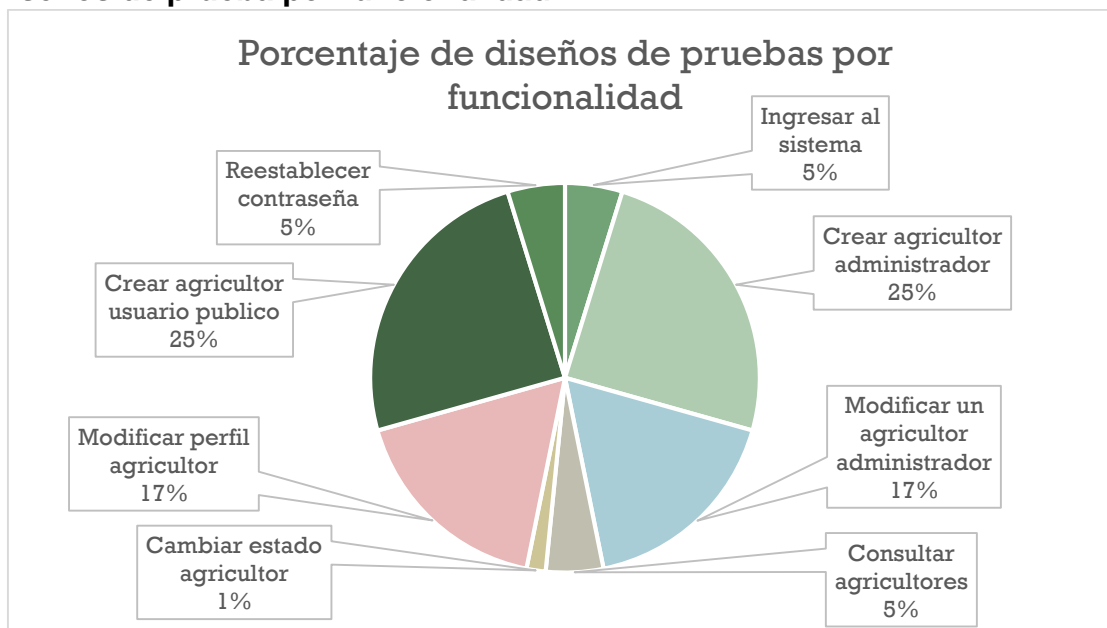
A continuación se presentan gráficos estadísticos con los datos obtenidos del resumen de ejecución.

- **Porcentaje de tipo de no conformidades encontradas**



En el gráfico anterior podemos observar el porcentaje de no conformidades encontradas según su tipo, podemos ver que no se encontró ninguna no conformidad bloqueante, lo que habla bien de los componentes que se evaluaron del sistema, pues no se detiene la ejecución de este. También podemos observar que el 50% de las no conformidades fueron funcionales y el otro 50% de presentación, esto quiere decir que se encontraron igual número de no conformidades de estos tipos.

- **Diseños de prueba por funcionalidad**

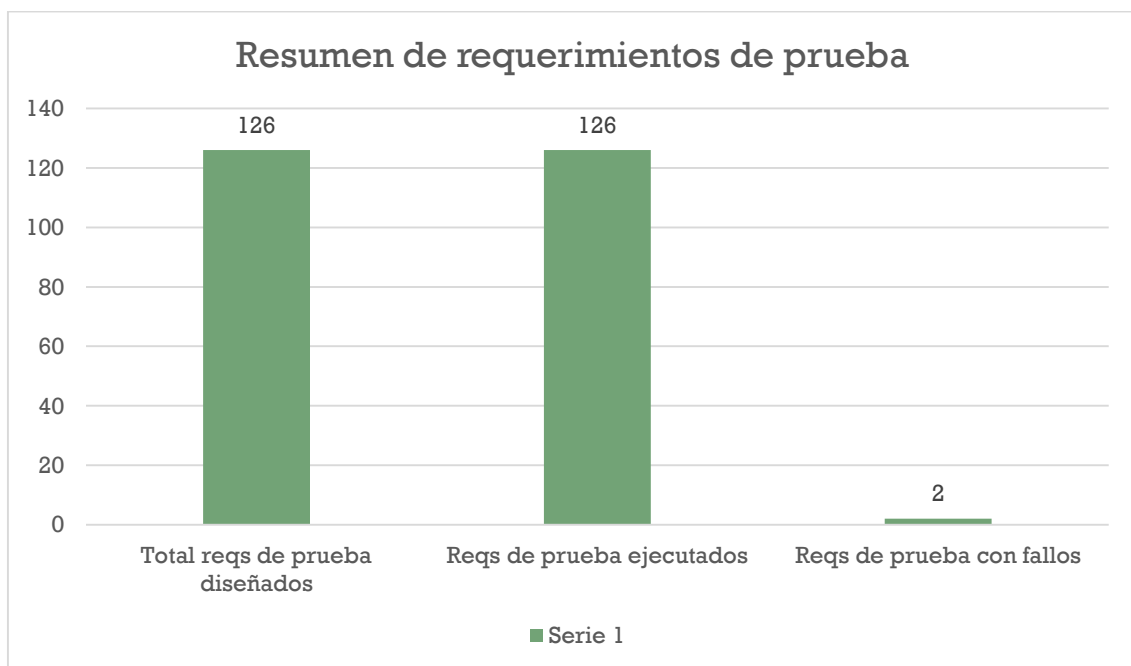


En el gráfico anterior podemos observar que las funcionalidades que tienen más diseños de pruebas son las de crear un agricultor, esto se debe a el número de campos que posee el formulario, pues es mayor que la de cualquier otro en este módulo.

- **Resumen de requerimientos de prueba**

En el siguiente gráfico podemos observar un resumen de requerimientos de pruebas, podemos ver cuál fue el total de requerimientos de prueba diseñados, ejecutados para este módulo y cuáles de estos presentaron fallos de algún tipo, del gráfico de barras podemos concluir que se ejecutaron la misma cantidad de requerimientos de prueba que diseños realizados, esto quiere decir que no quedaron diseños pendientes por ejecutar, también podemos observar que de 126 requerimientos de prueba ejecutado solo 2 tuvieron fallos de algún tipo, esto quiere decir que este módulo presento muy pocos errores en su funcionamiento al momento de realizar las pruebas.

-



8 Desarrollo de un módulo para gestionar los lotes que pertenecen a los agricultores.

8.1 Análisis

Para el desarrollo del módulo para gestionar los lotes que pertenecen a los agricultores, el primer paso fue realizar una identificación de la organización que los agricultores dan a sus terrenos. Con la ayuda de un ingeniero agrónomo que se desempeña en este entorno, se definió la organización básica que los agricultores tienen en sus tierras: en primer lugar los agricultores poseen fincas, que son la organización de tierra más grande que tienen. Una finca está compuesta por uno o más lotes, que son subdivisiones que están destinadas a la siembra de algún tipo de cultivo en específico. A su vez un lote está compuesto por uno o más bloques, es en estos donde se realiza la siembra del cultivo.

Además de la organización de sus tierras, los agricultores también tienen personas a su disposición para realizar las labores manuales en los bloques. Estas personas son llamadas Colaboradores-y es necesaria su identificación para llevar un registro de las labores realizadas en los bloques y quiénes son sus responsables.

En la Tabla 6 se presentan las funcionalidades que cada tipo de usuario podrá realizar dentro del módulo para gestionar los lotes que pertenecen a los agricultores.

Tabla 6 Funcionalidades por tipo de usuario del módulo para gestionar los lotes que pertenecen a los agricultores

Funcionalidad	Administrador	Agricultor	Colaborador
Crear finca		X	
Modificar finca		X	
Consultar fincas		X	
Filtrar fincas		X	
Asignar ubicación geográfica de una finca		X	
Modificar ubicación geográfica de una finca		X	
Consultar ubicación geográfica de una finca		X	
Registrar equipo de maquinaria	X	X	
Modificar equipo de maquinaria	X	X	
Filtrar equipos de maquinaria	X	X	
Consultar equipos de maquinaria	X	X	

Registrar colaborador		X	
Modificar colaborador		X	
Consultar colaboradores		X	
Cambiar estado colaborador		X	
Detalles de lote		X	
Detalles de finca		X	
Crear lote		X	
Modificar lote		X	
Consultar lotes		X	
Asignar ubicación geográfica de un lote		X	
Modificar ubicación geográfica lote		X	
Consular ubicación geográfica lote		X	
Registrar huella dactilar colaborador			X
Modificar huella dactilar colaborador			X

Una vez identificadas las funcionalidades, se procedió a definir las historias de usuario para este módulo, en total se definieron 25 historias de usuario, planificadas para llevarse a cabo a lo largo de 3 Sprints. En la Ilustración 30 se muestran algunas de las historias de usuario definidas en **Zube** con sus respectivos puntos funcionales y el sprint al que pertenecen.

8.2 Implementación

A continuación se presentan las capturas de pantalla de algunas de las funcionalidades implementadas en este módulo. La Ilustración 31 muestra la interfaz gráfica de la funcionalidad de crear una finca, en la ilustración podemos observar que como dato de entrada para hacer el registro se necesita el nombre de la finca, además de eso hay una sección para asignar la ubicación geográfica por medio de un dibujo del área. En esta sección se le presentan las instrucciones al agricultor para hacer la correcta asignación de la ubicación.

<input type="checkbox"/>	#17	Crear Finca	8	Done	J	Modulo de lotes	Sprint #3
<input type="checkbox"/>	#18	Modificar finca	5	Done	J	Modulo de lotes	Sprint #3
<input type="checkbox"/>	#19	Consultar fincas	3	Done	J	Modulo de lotes	Sprint #3
<input type="checkbox"/>	#20	Filtrar fincas	2	Done	J	Modulo de lotes	Sprint #3
<input type="checkbox"/>	#23	Asignar ubicación geográfica de una finca	20	Done	J	Modulo de lotes	Sprint #3
<input type="checkbox"/>	#24	Modificar ubicación geográfica de una finca	20	Done	J	Modulo de lotes	Sprint #3
<input type="checkbox"/>	#25	Consultar ubicación geográfica de una finca	13	Done	J	Modulo de lotes	Sprint #3
<input type="checkbox"/>	#66	Registrar equipo de maquinaria	13	Done	J	Modulo de lotes	Sprint #4
<input type="checkbox"/>	#67	Modificar equipo de maquinaria	8	Done	J	Modulo de lotes	Sprint #4
<input type="checkbox"/>	#68	Consultar equipos de maquinaria	3	Done	J	Modulo de lotes	Sprint #4
<input type="checkbox"/>	#69	Filtrar equipos de maquinaria	2	Done	J	Modulo de lotes	Sprint #4
<input type="checkbox"/>	#75	Registrar colaborador	8	Done	J	Modulo de lotes	Sprint #5
<input type="checkbox"/>	#76	Modificar colaborador	5	Done	J	Modulo de lotes	Sprint #5
<input type="checkbox"/>	#77	Consultar colaboradores	2	Done	J	Modulo de lotes	Sprint #5
<input type="checkbox"/>	#78	Cambiar estado colaborador	2	Done	J	Modulo de lotes	Sprint #5

Ilustración 30 Historias de usuario del módulo de gestión de usuarios y permisos
Fuente: Elaboración propia

En la Ilustración 32 se puede observar la interfaz de detalle de una finca. Primeramente se indican el identificador y el nombre de la finca, después se puede ver la ubicación geográfica con su área y dentro de ella, también podemos observar la ubicación de sus lotes y bloques con enlaces para re direccionar al agricultor a ver sus detalles. En la parte derecha de la interfaz se encuentra una tabla con los lotes que tiene la finca y además hay un botón para agregar un lote nuevo.

DATOS DEL LOTE ^

Nombre de la finca

DIBUJAR AREA ^

Esta opción consiste en seleccionar el área en el mapa que se presenta en pantalla. A continuación las instrucciones para interactuar con el mapa:

- Para desplazarse por el mapa se hace con clic sostenido.
- Para hacer un zoom o acercamiento sobre el mapa se debe seleccionar la mano en la parte superior derecha del mapa y hacer doble clic sobre el lugar al que se desea hacer zoom, o también se puede hacer usando el botón + que está ubicado en la parte superior izquierda del mapa.
- Para dibujar el área seleccione el polígono en la parte superior derecha del mapa, a continuación haga clic para marcar un punto e ir dibujando el área deseada, realice tantos puntos como sea necesario y cuando el área esté terminada hacer doble clic sobre el último punto.

Area de la finca

Ilustración 31 Funcionalidad de crear una finca
Fuente: Elaboración propia

FINCA

A continuación se presenta la información de la finca de id 1 y de nombre El Paraíso.

ID de la Finca: 1
Nombre de la finca: El Paraíso

AGREGAR LOTE

LOTES EN LA FINCA EL PARAÍSO:

10

por página

Identificador	Opciones
Lote El Paraíso-1	DETALLES
Identificador	Opciones

Mostrando desde 1 hasta 1 de un total de 1 registros

Ilustración 32 Funcionalidad de detalles de finca
Fuente: Elaboración propia

8.3 Pruebas

Durante el desarrollo del módulo para gestionar los lotes que pertenecen a los agricultores se realizaron pruebas funcionales del sistema. En primer lugar seleccionaron cuales serían las funcionalidades que se iban a probar, pues por cuestiones de tiempo y costo no se pudieron probar todas las funcionalidades para este módulo, se determinó que se le realizarían pruebas a un total de 16 funcionalidades. Una vez identificadas las funcionalidades se llevó a cabo el diseño de las pruebas funcionales que se iban a realizar. Posteriormente, con el apoyo de las clases identificadas se definieron los casos de prueba para la funcionalidad. Después de tener el diseño de todas las funcionalidades, el siguiente paso fue codificarlas y ejecutarlas usando la herramienta **Selenium IDE**. Con las pruebas codificadas, se procedió a agruparlas y realizar la ejecución para analizar el comportamiento de la funcionalidad.

En la se muestran dos ilustraciones con el diseño y la ejecución de las pruebas funcionales para la funcionalidad de crear agricultor de este módulo. En la Ilustración 33 se muestran algunos casos de prueba diseñados para esta funcionalidad y en la Ilustración 34 se puede observar la ejecución de todos los casos de prueba que fueron diseñados para esta funcionalidad. En la parte izquierda se pueden ver los casos que se ejecutaron, el color verde significa que no fallo la prueba y el color rojo indica que la prueba fallo. En la parte derecha se pueden ver los pasos que se deben seguir para ejecutar algún caso de prueba que se haya seleccionado en la parte izquierda. Por último, en la parte inferior se puede ver un resumen de la ejecución de los casos de prueba.

Tabla 7 Diseño y ejecución de pruebas funcionales para el módulo de gestión de lotes

Tecnica:	Tablas de decisión							
Funcionalidad:	Modificar finca							
	1	2	3	4	5	6	7	8
Digitar nombre finca valido	S	S	S	S	N	N	N	N
Dibujar area	S	N	S	N	S	S	N	N
subir archivo kml	S	S	N	N	S	N	S	N
Registro de finca en base de datos, redirecciona la lista de fincas y mensaje: "Finca modificada exitosamente"	X	X	X	-	-	-	-	-
Registro fallido y mensaje debajo del campo Nombre: "El campo es requerido."	-	-	-	-	X	X	X	X
Registro fallido y mensaje al principio del formulario: "Error al registrar la finca, por favor guarde la ubicación e inténtelo de nuevo."	-	-	-	X	X	X	X	X
Paso	S	S	S	S	N	N	N	S

Ilustración 33 Casos de prueba para la funcionalidad modificar finca

Ilustración 34 Ejecución de casos de prueba para la funcionalidad modificar finca

Con la ejecución de las pruebas finalizada, se consignaron los resultados en el archivo de diseño de pruebas, se puede observar en el archivo adjunto “Diseños de pruebas funcionales-Modulo de lotes.pdf”. En este documento encontramos los diseños de las pruebas y también el resultado de su ejecución.

A Continuación en la Tabla 8 se muestra el resumen de ejecución de los casos pruebas que se diseñaron para este módulo, en donde se detalla por cada funcionalidad probada el número de pruebas en las cuales se identificaron no conformidades (separadas por tipo), número de diseños y por último el número de pruebas que fueron ejecutadas. En los encabezados de las columnas se reemplazaron las palabras “no conformidad” por la abreviatura “NC” y la palabra “requerimientos” por “reqs”.

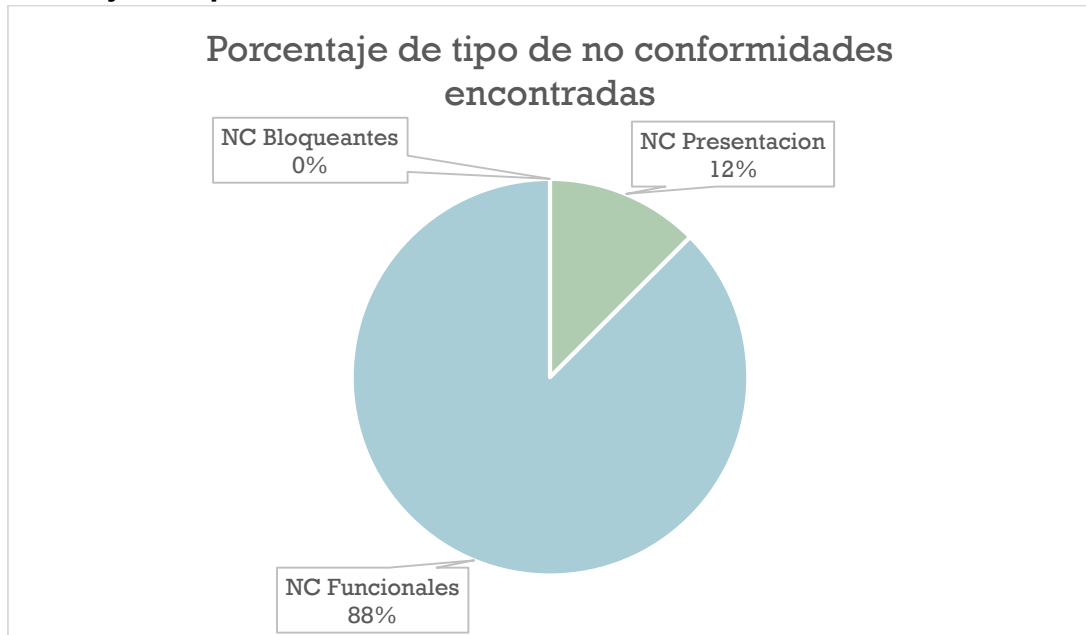
Tabla 8 Resumen de ejecución de casos de prueba para el módulo de gestión de usuarios

Funcionalidad probada	Total NC bloqueantes	Total NC funcionales	Total NC presentación	Total NC	Total reqs de prueba diseñados	Reqs de prueba ejecutados
Crear finca	0	3	0	3	8	8
Modificar finca	0	3	0	3	8	8
Consultar fincas	0	0	1	1	6	6
Asignar ubicación geográfica de una finca	0	0	0	0	4	4
Modificar ubicación geográfica de una finca	0	0	0	0	4	4
Consultar ubicación geográfica de una finca	0	0	0	0	3	3
Registrar colaborador	0	1	0	1	31	31
Modificar colaborador	0	1	0	1	31	31
Consultar colaboradores	0	0	0	0	6	6
Cambiar estado colaborador	0	0	0	0	2	2
Crear lote	0	3	0	3	8	8
Modificar lote	0	3	0	3	8	8
Consultar lotes	0	0	1	1	6	6
Asignar ubicación geográfica de un lote	0	0	0	0	4	4

Modificar ubicación geográfica lote	0	0	0	0	4	4
Consular ubicación geográfica lote	0	0	0	0	3	3
Totales	0	14	2	16	136	136

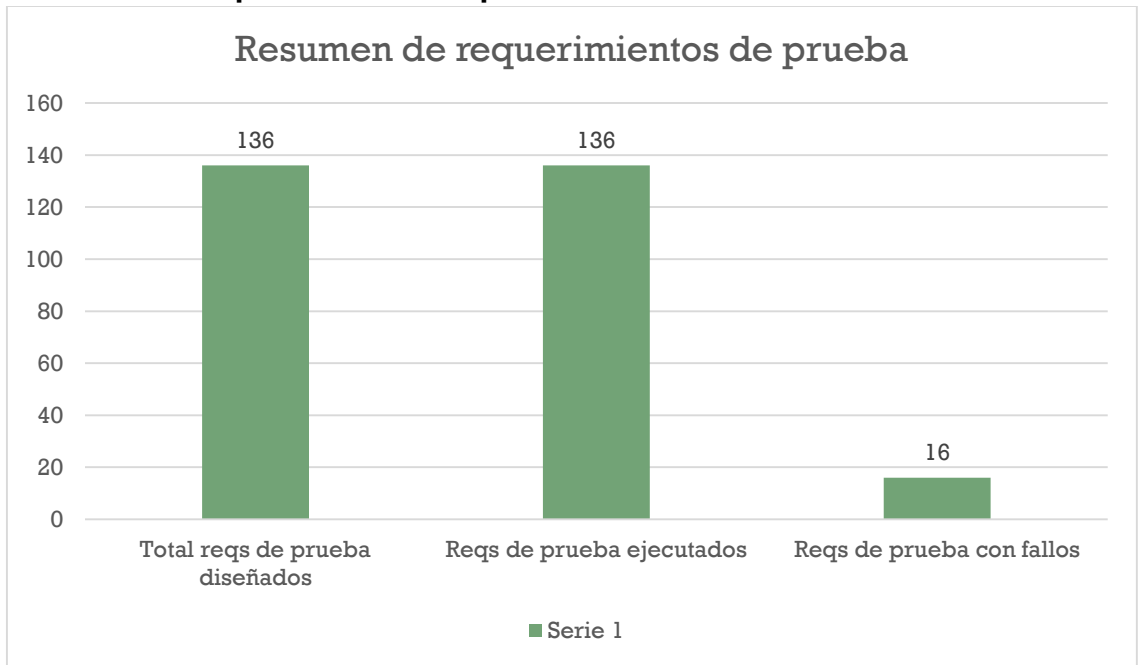
A continuación se presentan gráficos estadísticos con los datos obtenidos del resumen de ejecución.

- **Porcentaje de tipo de no conformidades encontradas**



En el gráfico anterior podemos observar el porcentaje de no conformidades encontradas según su tipo, podemos ver que no se encontró ninguna no conformidad bloqueante, lo que habla bien de los componentes que se evaluaron del sistema, pues no se detiene la ejecución de este. También podemos observar que el 88% de las no conformidades fueron funcionales y el otro 12% de presentación, esto quiere decir que la mayoría de no conformidades encontradas fueron funcionales y en este caso se debieron a campos que no realizaban las validaciones adecuadas en sus respectivos formularios.

- **Resumen de requerimientos de prueba**



En el gráfico anterior podemos observar un resumen de requerimientos de pruebas, podemos ver cuál fue el total de requerimientos de prueba diseñados y ejecutados para este módulo, además cuáles de estos presentaron fallos de algún tipo, del gráfico de barras podemos concluir que se ejecutaron la misma cantidad de requerimientos de prueba que diseños realizados, esto quiere decir que no quedaron diseños pendientes por ejecutar, también podemos observar que de 136 requerimientos de prueba ejecutado 16 tuvieron fallos de algún tipo, esto quiere decir que este módulo presentó muy pocos errores en su funcionamiento al momento de realizar las pruebas, aunque el número de errores aumentó considerablemente respecto al módulo anterior.

9 Desarrollo de un módulo para la gestión de datos de los diferentes cultivos sembrados en los lotes.

9.1 Análisis

Para el desarrollo del módulo para gestionar los datos de los diferentes cultivos sembrados en los lotes, el primer paso fue realizar una caracterización de los cultivos identificando los datos necesarios para llevar a cabo una gestión correcta. En esta sección debemos tener en cuenta que en el sector agrícola los cultivos sembrados son conocidos como “bloques”. Con la ayuda de un ingeniero agrónomo que se desempeña en este entorno, se identificaron los datos más importantes para un agricultor a la hora de sembrar un bloque. Es necesario identificar el área de siembra del bloque, además se debe tener un registro de cuál es el número de plantas y el tipo de cultivo que se sembrará en él. Es importante también conocer si el bloque se encuentra en estado de inversión o producción, pues en el primero solo se generan gastos y en el otro se comienzan a generar ganancias por ventas. También es necesario conocer si un bloque se encuentra actualmente en uso, pues en la misma área dentro de un lote, se pueden haber sembrado bloques diferentes a lo largo de un periodo de tiempo extenso. Por último se deben tener en cuenta las fechas de siembra y cierre del bloque. Es importante también tener en cuenta que los diferentes tipos de cultivos que van a ser sembrados, pueden clasificarse entre frutas, hortalizas y cultivos industriales como la caña de azúcar y el algodón

En la Tabla 9 se presentan las funcionalidades que cada tipo de usuario podrá realizar dentro del módulo para gestionar los diferentes cultivos sembrados en los lotes.

Tabla 9 Funcionalidades por tipo de usuario del módulo para gestionar los diferentes cultivos sembrados en los lotes

Funcionalidad	Administrador	Agricultor	Colaborador
Crear bloque		X	
Modificar bloque		X	
Consultar bloques		X	
Filtrar bloques		X	
Crear tipo de cultivo	X	X	
Modificar tipo de cultivo	X	X	
Consultar tipos de cultivo	X	X	
Filtrar tipos de cultivo		X	

Cambiar estado del bloque		X	
Asignar ubicación geográfica de un bloque		X	
Modificar ubicación geográfica de un bloque		X	
Consultar ubicación geográfica de un bloque		X	

Una vez identificadas las funcionalidades, se procedió a definir las historias de usuario para este módulo, en total se definieron 12 historias de usuario, planificadas para llevarse a cabo a lo largo de 2 Sprints. En la se muestran algunas de las historias de usuario definidas en Zube con sus respectivos puntos funcionales y el sprint al que pertenecen.

<input type="checkbox"/>	#13	Crear Bloque	2	Done	Modulo de cultivos	Sprint #4
<input type="checkbox"/>	#14	Modificar bloque	2	Done	Modulo de cultivos	Sprint #4
<input type="checkbox"/>	#15	Consultar bloques	2	Done	Modulo de cultivos	Sprint #4
<input type="checkbox"/>	#16	Filtrar bloques	2	Done	Modulo de cultivos	Sprint #4
<input type="checkbox"/>	#28	Crear tipo de cultivo	2	Done	Modulo de cultivos	Sprint #4
<input type="checkbox"/>	#29	Modificar tipo de cultivo	2	Done	Modulo de cultivos	Sprint #4
<input type="checkbox"/>	#30	Consultar tipos de cultivo	2	Done	Modulo de cultivos	Sprint #4
<input type="checkbox"/>	#65	Cambiar estado del bloque	2	Done	Modulo de cultivos	Sprint #4
<input type="checkbox"/>	#89	Asignar ubicacion geografica a un bloque	13	Ready	Modulo de cultivos	Sprint #5
<input type="checkbox"/>	#90	Modificar ubicación geográfica de un bloque	8	Ready	Modulo de cultivos	Sprint #5
<input type="checkbox"/>	#91	Consultar ubicacion geografica de un bloque	5	Ready	Modulo de cultivos	Sprint #5
<input type="checkbox"/>	#92	Filtrar tipos de cultivo	2	Ready	Modulo de cultivos	Sprint #5

Ilustración 35 Historias de usuario del módulo para la gestión de datos de los diferentes cultivos sembrados
Fuente: Elaboración propia

9.2 Implementación

A continuación se presentan las capturas de pantalla de algunas de las funcionalidades implementadas en este módulo. La Ilustración 36 muestra la interfaz gráfica de la funcionalidad de registrar un tipo de cultivo, en esta podemos observar que como dato de entrada para hacer el registro se necesita ingresar el nombre del cultivo y seleccionar la categoría a la cual pertenece, ya sea fruta, hortaliza, etc. También están los botones para cancelar y confirmar el registro.

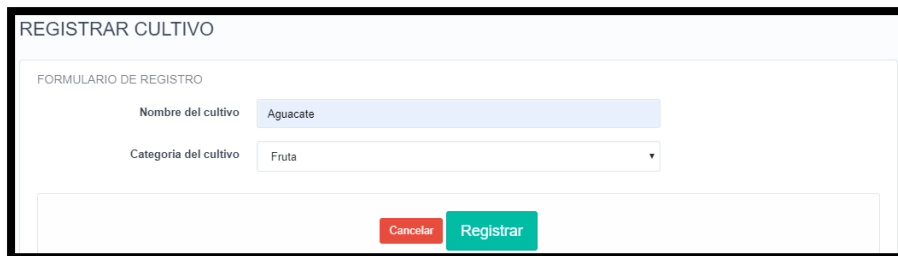


Ilustración 36 Funcionalidad de registrar un tipo de cultivo
Fuente: Elaboración propia

La Ilustración 37 muestra la interfaz gráfica de la funcionalidad de registrar un bloque, en esta podemos observar un botón para ver el lote al cual se le asignará el bloque, también está el formulario consta de varias secciones, en la primera se ingresan los datos del bloque, como el número de plantas, el tipo de cultivo y la fecha de siembra. En la otra sección se puede asignar la ubicación geográfica por medio de un dibujo del área. En esta sección se le presentan las instrucciones al agricultor para hacer la correcta asignación de la ubicación.

En la Ilustración 38 se muestra la interfaz gráfica de la funcionalidad de listar los bloques registrados por un agricultor, en la ilustración podemos observar una tabla con los bloques, estos se muestran con los campos del identificador del bloque, el identificador del lote, el tipo de cultivo sembrado en el bloque, el estado, la etapa en que se encuentra el bloque, por defecto se crean en etapa de inversión, la fecha de siembra y la fecha de cierre en caso de que ya no se encuentre activo el bloque. Además de un botón de acciones por medio del cual se puede seleccionar un bloque, ya sea para ser modificado, para ver sus detalles, para ingresar registros de costos asociados a él, o para registrar costos de producción y ventas.



Ilustración 37 Funcionalidad de registrar un bloque
Fuente: Elaboración propia

LISTA DE BLOQUES
A continuación se presenta una lista con los bloques que se encuentran registrados en el sistema.

BLOQUES.

10 registros por página

Identificador	Lote ID	Tipo de cultivo	Estado	Etapa	Fecha de siembra	Fecha de cierre	Acciones
Casa grande-1-1	Casa grande-1	Papaya	Activo	Inversión	9 de Mayo de 2018	En funcionamiento	Acciones
El Paraíso-1-1	El Paraíso-1	Papaya	Activo	Inversión	8 de Marzo de 2018	En funcionamiento	Acciones
El Paraíso-1-2	El Paraíso-1	Papaya	Activo	Inversión	8 de Enero de 2018	En funcionamiento	Acciones
Identificador	Lote ID	Tipo de cultivo	Estado	Etapa	Fecha de siembra	Fecha de cierre	Acciones

Mostrando desde 1 hasta 3 de un total de 3 registros

< Previous 1 Next >

Ilustración 38 Funcionalidad de listar bloques
Fuente: Elaboración propia

9.3 Pruebas

Durante el desarrollo del módulo para gestionar los datos de los cultivos sembrados en los lotes se realizaron pruebas funcionales del sistema. En primer lugar seleccionaron cuales serían las funcionalidades que se iban a probar, pues por cuestiones de tiempo y costo no se pudieron probar todas las funcionalidades para este módulo, se determinó que se le realizarían pruebas a un total de 8 funcionalidades. Una vez identificadas las funcionalidades se llevó a cabo el diseño de las pruebas funcionales que se iban a realizar. Posteriormente, con el apoyo de las clases identificadas se definieron los casos de prueba para la funcionalidad. Después de tener el diseño de todas las funcionalidades, el siguiente paso fue codificarlas y ejecutarlas usando la herramienta **Selenium IDE**. Con las pruebas codificadas, se procedió a agruparlas y realizar la ejecución para analizar el comportamiento de la funcionalidad.

En la se muestran dos ilustraciones con el diseño y la ejecución de las pruebas funcionales para algunas funcionalidades de este módulo. En la Ilustración 39 se muestran algunos casos de prueba diseñados para la funcionalidad de asignar una ubicación geográfica a un bloque y en la Ilustración 40 se puede observar la ejecución de todos los casos de prueba que fueron diseñados para la funcionalidad de modificar un tipo de cultivo. En la parte izquierda se pueden ver los casos que se ejecutaron, el color verde significa que no fallo la prueba y en caso de que alguna hubiera fallado se hubiera subrayado en color rojo. En la parte derecha se pueden ver los pasos que se deben seguir para ejecutar algún caso de prueba que se haya seleccionado en la parte izquierda.

Tabla 10 Diseño y ejecución de pruebas funcionales para el módulo de gestión de cultivos

Técnica:	Tablas de decisión			
Funcionalidad:	Asignar ubicación geográfica de un bloque			
	1	2	3	4
Dibujar area	S	N	S	N
subir archivo kml	S	S	N	N
Registro de finca en base de datos, redirecciona la lista de fincas y mensaje: "Finca registrada exitosamente"	X	X	X	-
Registro fallido y mensaje al principio del formulario: "Error al registrar la finca, por favor guarde la ubicación e inténtelo de nuevo."	-	-	-	X
Paso	S	S	S	S

Ilustración 39 Casos de prueba para la funcionalidad de asignar una ubicación geográfica a un bloque

Ilustración 40 Ejecución de casos de prueba para la funcionalidad de modificar tipo de cultivo

Con la ejecución de las pruebas finalizada, se consignaron los resultados en el archivo de diseño de pruebas, se puede observar en el archivo adjunto “Diseños de pruebas funcionales-Modulo de cultivos.pdf”. En este documento encontramos los diseños de las pruebas y también el resultado de su ejecución.

A Continuación en la Tabla 8 se muestra el resumen de ejecución de los casos pruebas que se diseñaron para este módulo, en donde se detalla por cada funcionalidad probada el número de pruebas en las cuales se identificaron no conformidades (separadas por tipo), número de diseños y por último el número de pruebas que fueron ejecutadas. En los encabezados de las columnas se reemplazaron las palabras “no conformidad” por la abreviatura “NC” y la palabra “requerimientos” por “reqs”.

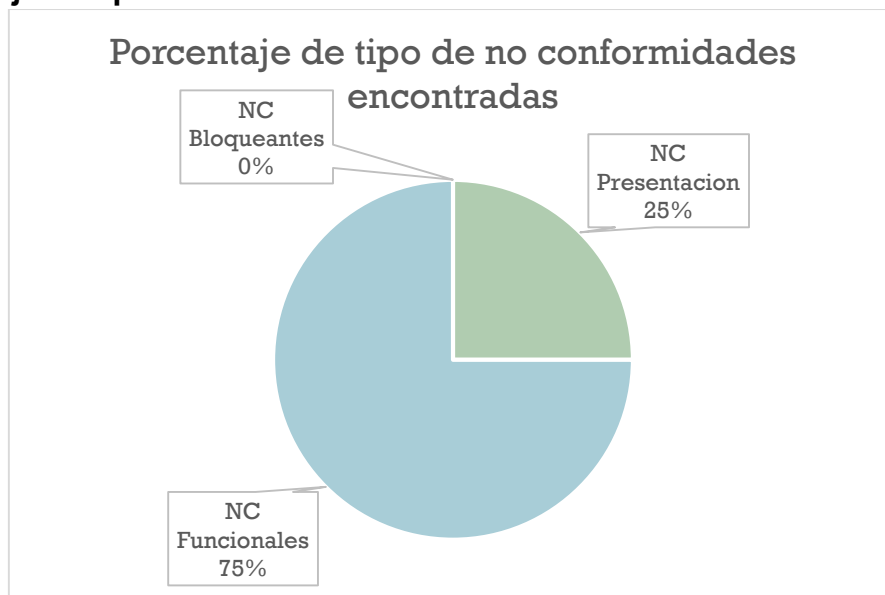
Tabla 11 Resumen de ejecución de casos de prueba para el módulo de gestión de usuarios

Funcionalidad probada	Total NC bloqueantes	Total NC funcionales	Total NC presentación	Total NC	Total reqs de prueba diseñados	Reqs de prueba ejecutados
Crear bloque	0	3	0	3	11	11
Modificar bloque	0	3	0	3	11	11
Consultar bloques	0	0	1	1	6	6
Crear tipo de cultivo	0	0	0	0	6	6
Modificar tipo de cultivo	0	0	0	0	6	6
Consultar tipos de cultivo	0	0	1	1	6	6

Cambiar estado del bloque	0	0	0	0	2	2
Asignar ubicación geográfica de un bloque	0	0	0	0	4	4
Modificar ubicación geográfica de un bloque	0	0	0	0	4	4
Totales	0	6	2	8	56	56

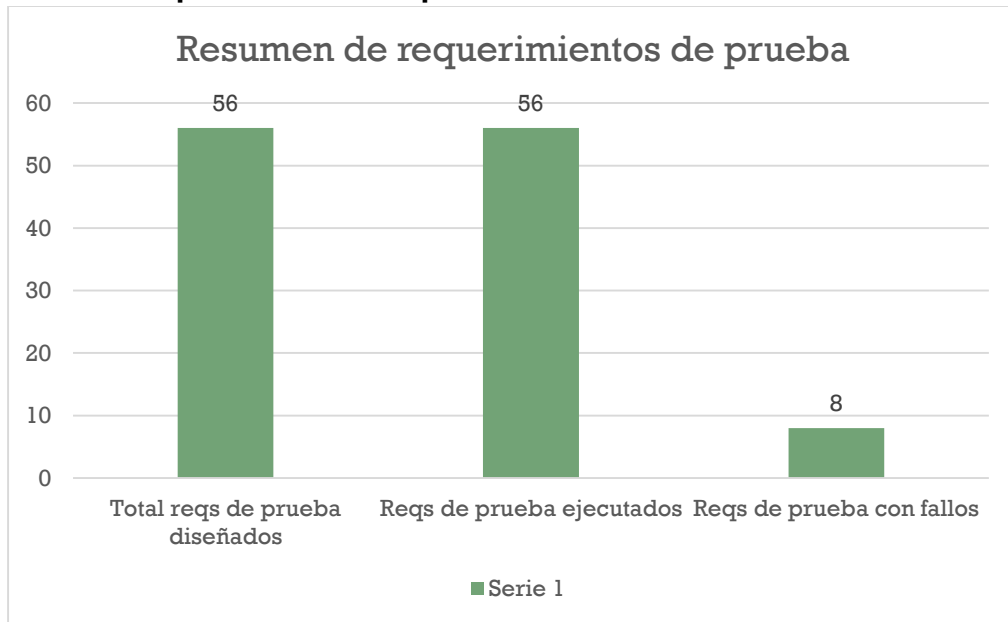
A continuación se presentan gráficos estadísticos con los datos obtenidos del resumen de ejecución.

- **Porcentaje de tipo de no conformidades encontradas**



En el gráfico anterior podemos observar el porcentaje de no conformidades encontradas según su tipo, podemos ver que no se encontró ninguna no conformidad bloqueante, lo que habla bien de los componentes que se evaluaron del sistema, pues no se detiene la ejecución de este. También podemos observar que el 75% de las no conformidades fueron funcionales y el otro 25% de presentación, esto quiere decir que la mayoría de no conformidades encontradas fueron funcionales y en este caso se debieron a campos que no realizaban las validaciones adecuadas en sus respectivos formularios.

- **Resumen de requerimientos de prueba**



En el gráfico anterior podemos observar un resumen de requerimientos de pruebas, podemos ver cuál fue el total de requerimientos de prueba diseñados y ejecutados para este módulo, además cuáles de estos presentaron fallos de algún tipo, del gráfico de barras podemos concluir que se ejecutaron la misma cantidad de requerimientos de prueba que diseños realizados, esto quiere decir que no quedaron diseños pendientes por ejecutar, también podemos observar que de 56 requerimientos de prueba ejecutado 8 tuvieron fallos de algún tipo, esto quiere decir que este módulo presentó muy pocos errores en su funcionamiento al momento de realizar las pruebas.

10 Desarrollo de un módulo de gestión de datos de costos y gastos de producción agrícola correspondientes a los lotes.

10.1 Análisis

Para el desarrollo del módulo de gestión de datos de costos y gastos de producción agrícola correspondientes a las fincas, el primer paso fue realizar una identificación de los diferentes tipos de costos que el agricultor puede registrar en la aplicación. Se identificaron los costos del proceso de producción y se dividieron en tres tipos de costos: los costos generados por los productos utilizados en los cultivos, los costos generados por las labores empleadas en los cultivos y los otros costos que no entran en las dos categorías anteriores como la adecuación del terreno o los estudios previos que se realizan antes de realizar la siembra.

Además de la organización de sus tierras, los agricultores también tienen personas a su disposición para realizar las labores manuales en los bloques. Estas personas son llamadas colaboradores-y es necesaria su identificación para llevar un registro de las labores realizadas en los bloques y quiénes son sus responsables.

En la Tabla 12 se presentan las funcionalidades que cada tipo de usuario podrá realizar dentro del módulo para gestionar los datos de costos y gastos de producción agrícola.

Tabla 12 Funcionalidades por tipo de usuario del módulo para gestionar los datos de costos y gastos de producción agrícola.

Funcionalidad	Administrador	Agricultor	Colaborador
Ingresar producto para costos	X	X	
Modificar producto para costos	X	X	
Ingresar labor para costos	X	X	
Modificar labor para costos	X	X	
Ingresar otro tipo de costos	X	X	
Modificar otro tipo de costos	X	X	
Consultar tipos de costos	X	X	
Filtrar tipos de costos	X	X	
Ingresar registro de costo por labor		X	

Modificar registro de costos por labor			
Filtrar registros de costos por labor		X	
Consultar registros de costos por labor		X	
Ingresar registro de costo por producto		X	
Modificar registro de costos por producto		X	
Filtrar registros de costos por producto		X	
Consultar registros de costos por producto		X	
Ingresar registro de costo de comercialización		X	
Modificar registro de costos de comercialización		X	
Filtrar registros de costos de comercialización		X	
Consultar registros de costos de comercialización		X	
Ingresar registro de costo		X	
Modificar registro de costos		X	
Filtrar registros de costos		X	
Consultar registros de costos		X	
Ingresar valor facturado por semana		X	
Modificar valor facturado por semana		X	
Consultar valor facturado por semana		X	
Generar valor de amortización		X	
Registrarse para iniciar labor			X
Registrarse para terminar labor			X
Consultar colaboradores con sus respectivas huellas dactilares software auxiliar		X	
Cargar colaboradores software auxiliar		X	
Actualizar registros de costos por labor periódicamente software auxiliar			

Una vez identificadas las funcionalidades, se procedió a definir las historias de usuario para este módulo, en total se definieron 34 historias de usuario, planificadas para llevarse a cabo a lo largo de 6 Sprints. En la **¡Error! No se encuentra el origen de la referencia.**

e muestran algunas de las historias de usuario definidas en Zube con sus respectivos puntos funcionales y el sprint al que pertenecen.

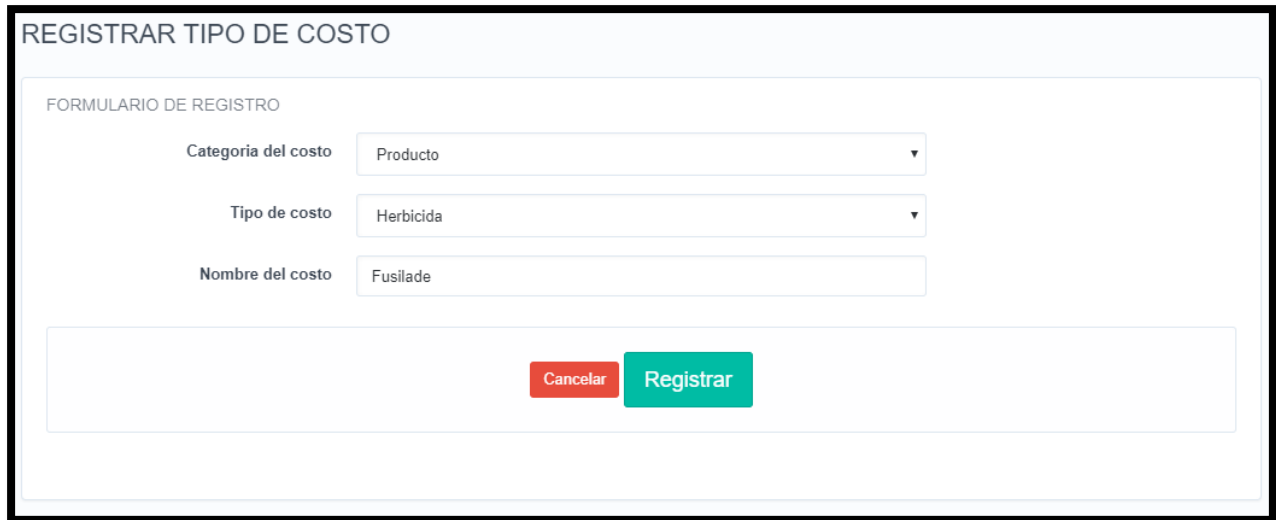
<input type="checkbox"/>	#63	Consultar valor facturado por semana	⦿ 2	Ready	J	✚ Modulo de costos 🏹 Sprint #8
<input type="checkbox"/>	#74	Registrar huella dactilar colaborador	⦿ 2	Ready	J	✚ Modulo de costos 🏹 Sprint #9
<input type="checkbox"/>	#42	Filtrar registros de costos por labor	⦿ 2	Ready	J	✚ Modulo de costos 🏹 Sprint #8
<input type="checkbox"/>	#41	Filtrar registros de costos por producto	⦿ 2	In Progress	J	✚ Modulo de costos 🏹 Sprint #7
<input type="checkbox"/>	#52	Filtrar costos de comercializacion	⦿ 2	Ready	J	✚ Modulo de costos 🏹 Sprint #7
<input type="checkbox"/>	#59	<input checked="" type="checkbox"/> Filtrar registros de costos	⦿ 2	Done	J	✚ Modulo de costos 🏹 Sprint #6
<input type="checkbox"/>	#73	Consultar colaboradores con sus respectivas huellas dactilares	⦿ 3	Ready	J	✚ Modulo de costos 🏹 Sprint #9
<input type="checkbox"/>	#48	<input checked="" type="checkbox"/> Consultar labores para costos	⦿ 3	Done	J	✚ Modulo de costos 🏹 Sprint #6
<input type="checkbox"/>	#62	Modificar valor facturado por semana	⦿ 5	Ready	J	✚ Modulo de costos 🏹 Sprint #8
<input type="checkbox"/>	#72	Registrarse para terminar labor	⦿ 5	Ready	J	✚ Modulo de costos 🏹 Sprint #9
<input type="checkbox"/>	#45	<input checked="" type="checkbox"/> Modificar labor para costos	⦿ 5	Done	J	✚ Modulo de costos 🏹 Sprint #6
<input type="checkbox"/>	#61	Ingresar valor facturado por semana	⦿ 8	Ready	J	✚ Modulo de costos 🏹 Sprint #8
<input type="checkbox"/>	#71	Registrarse para iniciar labor	⦿ 8	Ready	J	✚ Modulo de costos 🏹 Sprint #9
<input type="checkbox"/>	#44	<input checked="" type="checkbox"/> Crear labor para costos	⦿ 8	Done	J	✚ Modulo de costos 🏹 Sprint #6
<input type="checkbox"/>	#70	Generar valor de amortización	⦿ 20	Ready	J	✚ Modulo de costos 🏹 Sprint #9

Ilustración 41 Historias de usuario del módulo para la gestión de datos de costos y gastos de producción agrícola

Fuente: Elaboración propia

10.2 Implementación

A continuación se presentan las capturas de pantalla de algunas de las funcionalidades implementadas en este módulo. La Ilustración 42 muestra la interfaz gráfica de la funcionalidad de registrar un tipo de costo, en esta podemos observar que como dato de entrada para hacer el registro se necesita seleccionar la categoría a la cual pertenece el costo, después en este caso como es un producto, se debe seleccionar el tipo de producto que se desea registrar, además del nombre del producto. También están los botones para cancelar y confirmar el registro.



REGISTRAR TIPO DE COSTO

FORMULARIO DE REGISTRO

Categoria del costo

Tipo de costo

Nombre del costo

Ilustración 42 Funcionalidad de registrar tipo de costo
Fuente: Elaboración propia

En la Ilustración 43 se muestra la interfaz gráfica de la funcionalidad de listar los registros de costos por producto, en la ilustración podemos observar un título con el nombre del bloque al que se asocian los registros de costos que en este caso es El paraíso-1-1, además un panel con tres pestañas, una por cada tipo de costo que se le puede registrar al bloque: costos por producto, costos de comercialización y otros costos. Además de eso encontramos una tabla con los costos por producto que se han registrado en el sistema para el bloque El paraíso-1-1, estos registros se muestran con los campos de fecha de registro, nombre del producto, costo del producto, presentación del producto en ml, el volumen gastado en ml y el costo generado por la utilización de ese volumen de producto. Además de un botón de acciones por medio del cual se puede seleccionar un registro para ser modificado. También hay un botón que redirige al usuario para crear un nuevo agricultor.

La Ilustración 44 muestra la interfaz gráfica de la funcionalidad de ingresar un registro de costo por producto, en la ilustración podemos observar un formulario con los datos de entrada que se necesitan para hacer el registro, entre ellos tenemos la fecha, el tipo de costo que en este caso es el producto utilizado, el costo y la presentación del mismo y por último el volumen utilizado en la aplicación. También están los botones para cancelar y confirmar el registro.

REGISTROS DE COSTOS DEL BLOQUE EL PARAÍSO-1-1

A continuación se presentan tres pestañas con las listas de registros de costos separados por costo por producto, costos de comercialización y otros costos.

Costos por producto		Costos de comercialización			Otros costos	
Fecha de registro	Producto	Costo del producto	Presentación	Volumen gastado	Costo generado	Acciones
10 de Agosto de 2018	Agrodyne	90000,00	1000,00	60,00	5400,00	Acciones ▾
10 de Enero de 2019	Engeo	150000,00	1000,00	320,00	48000,00	Acciones ▾
11 de Julio de 2018	Finale	150000,00	500,00	80,00	24000,00	Acciones ▾
11 de Julio de 2018	Formax	350000,00	1000,00	230,00	80500,00	Acciones ▾
15 de Enero de 2019	Daconil	32000,00	1000,00	400,00	12800,00	Acciones ▾
18 de Septiembre de 2018	Agrodyne	85000,00	500,00	36,00	6120,00	Acciones ▾

Ilustración 43 Funcionalidad de listar registros de costos por producto
Fuente: Elaboración propia

REGISTRAR COSTO POR PRODUCTO.

FORMULARIO DE REGISTRO

Fecha de registro	<input type="text" value="07/12/2018"/>
Tipo de costo	<input type="text" value="Exalt"/>
Costo del producto	<input type="text" value="59000"/>
Presentación del producto (ml)	<input type="text" value="1000"/>
Volumen de producto utilizado (ml)	<input type="text" value="234"/>

Ilustración 44 Funcionalidad de ingresar registro de costo por producto
Fuente: Elaboración propia

10.3 Pruebas

Durante el desarrollo del módulo de gestión de datos de costos y gastos de producción agrícola correspondientes a las fincas se realizaron pruebas funcionales del sistema. En primer lugar seleccionaron cuales serían las funcionalidades que se iban a probar, pues por cuestiones de tiempo y costo no se pudieron probar todas las funcionalidades para este módulo, se determinó que se le realizarían pruebas a un total de 16 funcionalidades. Una vez identificadas las funcionalidades se llevó a cabo el diseño de las pruebas funcionales que se iban a realizar. Posteriormente, con el apoyo de las clases identificadas se definieron los casos de prueba para la funcionalidad. Después de tener el diseño de todas las funcionalidades, el siguiente paso fue codificarlas y ejecutarlas usando la herramienta **Selenium IDE**. Con las pruebas codificadas, se procedió a agruparlas y realizar la ejecución para analizar el comportamiento de la funcionalidad.

En la Tabla 13 se muestran dos ilustraciones con el diseño y la ejecución de las pruebas funcionales para la funcionalidad de ingresar un registro de costo por producto perteneciente a este módulo. En la Ilustración 45 se muestran algunos casos de prueba diseñados para esta funcionalidad y en la Ilustración 46 se puede observar la ejecución de todos los casos de prueba que fueron diseñados para esta funcionalidad. En la parte izquierda se pueden ver los casos que se ejecutaron, el color verde significa que no fallo la prueba y el color rojo significa que la prueba fallo. En la parte derecha se pueden ver los pasos que se deben seguir para ejecutar algún caso de prueba que se haya seleccionado en la parte izquierda.

Tabla 13 Diseño y ejecución de pruebas funcionales para el módulo de gestión de costos y gastos

Presentación	Volumen utilizado	Resultado esperado
)	1000	180
)	1000	180
)	1000	180
)	1000	180
)	1000	180

Redireccionar a la lista de registros costos y mostrar un mensaje en un recuadro verde con el siguiente mensaje: "Éxito! El registro de costo por producto fue modificado exitosamente."

Redireccionar a la lista de registros costos y mostrar un mensaje en un recuadro verde con el siguiente mensaje: "Éxito! El registro de costo por producto fue modificado exitosamente."

Redireccionar a la lista de registros costos y mostrar un mensaje en un recuadro verde con el siguiente mensaje: "Éxito! El registro de costo por producto fue modificado exitosamente."

Redireccionar a la lista de registros costos y mostrar un mensaje en un recuadro verde con el siguiente mensaje: "Éxito! El registro de costo por producto fue modificado exitosamente."

Redireccionar a la lista de registros costos y mostrar un mensaje en un recuadro verde con el siguiente mensaje: "Éxito! El registro de costo por producto fue modificado exitosamente."

Ilustración 45 Casos de prueba para la funcionalidad de modificar un registro de costo por producto

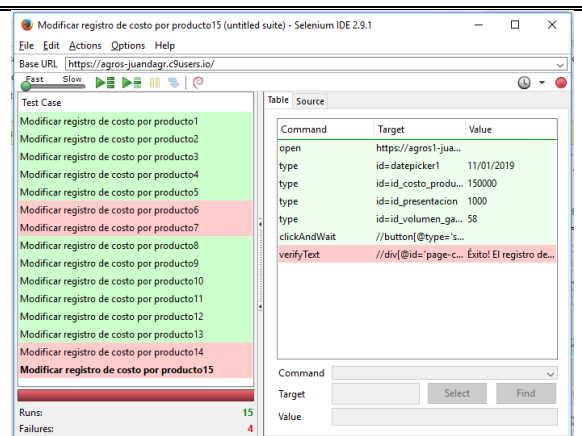


Ilustración 46 Ejecución de casos de prueba para la funcionalidad de modificar un registro de costo por producto

Con la ejecución de las pruebas finalizada, se consignaron los resultados en el archivo de diseño de pruebas, se puede observar en el archivo adjunto "Diseños de pruebas funcionales-Modulo de costos.pdf". En este documento encontramos los diseños de las pruebas y también el resultado de su ejecución.

A Continuación en la Tabla 14 se muestra el resumen de ejecución de los casos pruebas que se diseñaron para este módulo, en donde se detalla por cada funcionalidad probada el número de pruebas en las cuales se identificaron no conformidades (separadas por tipo), número de diseños y por último el número de pruebas que fueron ejecutadas. En los encabezados de las columnas se reemplazaron las palabras “no conformidad” por la abreviatura “NC” y la palabra “requerimientos” por “reqs”.

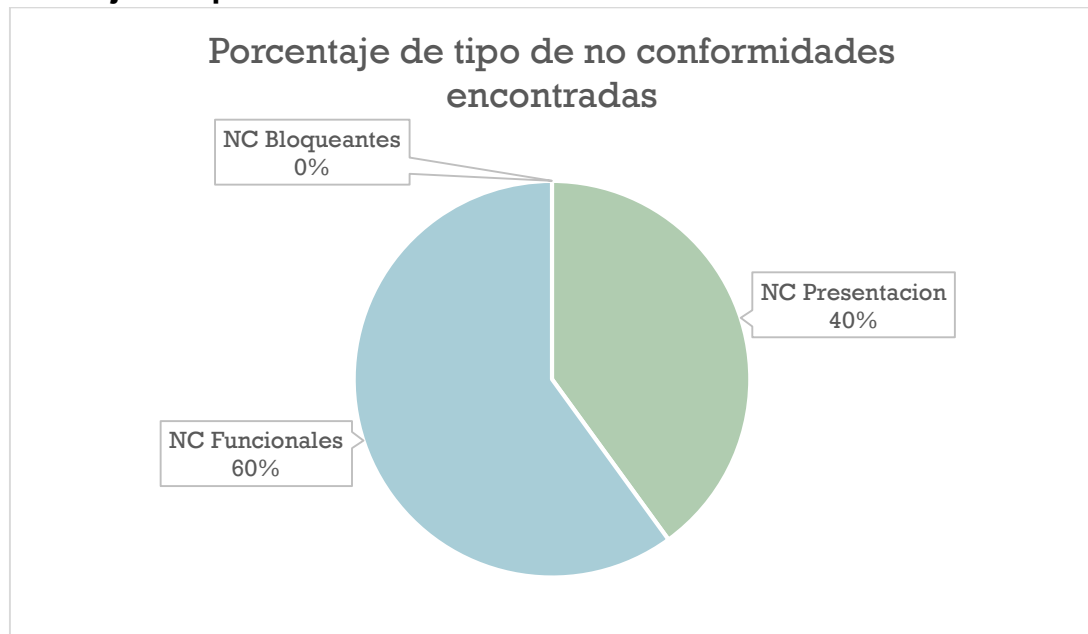
Tabla 14 Resumen de ejecución de casos de prueba para el módulo de gestión de usuarios

Funcionalidad probada	Total NC bloqueantes	Total NC funcionales	Total NC presentación	Total NC	Total reqs de prueba diseñados	Reqs de prueba ejecutados
Ingresar producto para costos	0	0	0	0	6	6
Modificar producto para costos	0	0	0	0	6	6
Ingresar labor para costos	0	0	0	0	3	3
Modificar labor para costos	0	0	0	0	3	3
Ingresar otro tipo de costos	0	0	0	0	3	3
Modificar otro tipo de costos	0	0	0	0	3	3
Consultar tipos de costos	0	0	1	1	6	6
Ingresar registro de costo por producto	0	4	0	4	15	15
Modificar registro de costos por producto	0	4	0	4	15	15
Consultar registros de costos por producto	0	0	1	1	6	6
Ingresar registro de costo de comercialización	0	0	0	0	9	9
Modificar registro de costos de comercialización	0	0	0	0	9	9
Consultar registros de costos de comercialización	0	0	1	1	6	6

Ingresar registro de costo	0	0	0	0	9	9
Modificar registro de costos	0	0	0	0	9	9
Consultar registros de costos	0	0	1	1	6	6
Totales	0	8	4	12	114	114

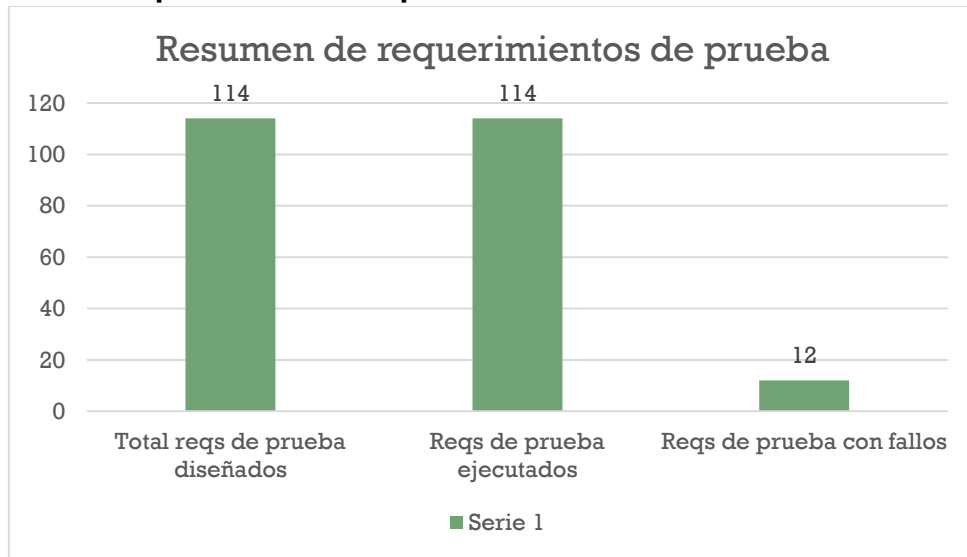
A continuación se presentan gráficos estadísticos con los datos obtenidos del resumen de ejecución.

- **Porcentaje de tipo de no conformidades encontradas**



En el gráfico anterior podemos observar el porcentaje de no conformidades encontradas según su tipo, podemos ver que no se encontró ninguna no conformidad bloqueante, lo que habla bien de los componentes que se evaluaron del sistema, pues no se detiene la ejecución de este. También podemos observar que el 60% de las no conformidades fueron funcionales y el otro 40% de presentación, esto quiere decir que la mayoría de no conformidades encontradas fueron funcionales y en este caso se debieron a campos que no realizaban las validaciones adecuadas en sus respectivos formularios.

- **Resumen de requerimientos de prueba**



En el gráfico anterior podemos observar un resumen de requerimientos de pruebas, podemos ver cuál fue el total de requerimientos de prueba diseñados y ejecutados para este módulo, además cuáles de estos presentaron fallos de algún tipo, del gráfico de barras podemos concluir que se ejecutaron la misma cantidad de requerimientos de prueba que diseños realizados, esto quiere decir que no quedaron diseños pendientes por ejecutar, también podemos observar que de 114 requerimientos de prueba ejecutado 12 tuvieron fallos de algún tipo, esto quiere decir que este módulo presentó muy pocos errores en su funcionamiento y presentación al momento de realizar las pruebas.

11 Desarrollo de un módulo para la gestión de datos de producción que se presenta en los diferentes cultivos.

11.1 Análisis

Para el desarrollo del módulo de gestión de datos de producción que se presenta en los diferentes cultivos, el primer paso fue realizar una caracterización del proceso de producción y venta que se presentan en un bloque. Con la ayuda de un ingeniero agrónomo que se desempeña en este entorno, se identificó que una vez el bloque se encuentre en estado de producción, este comienza a tener producciones periódicas, estas producciones se pueden clasificar en tres tipos de calidades: Selecta, que es la mayor calidad que se puede producir; Corriente, que es una calidad intermedia e Industrial que es la calidad más baja que puede producir un bloque. Además de una calidad, se produce una cantidad limitada de fruta o verdura y también se debe tener un registro de la fecha en que se recolecto dicha producción. Después de tener la fruta o verdura producidas, el siguiente paso en el proceso de producción agrícola es la venta, una producción puede tener una o más ventas asociadas, pues no toda la producción se le vende al mismo comprador, además del comprador, también es necesario conocer que día se realizó la venta. En una venta se especifica cual fue la cantidad de kilogramos que se vendieron y cuál fue el precio de venta por cada kilogramo.

En la Tabla 12 se presentan las funcionalidades que cada tipo de usuario podrá realizar dentro del módulo para gestionar los datos de costos y gastos de producción agrícola.

Tabla 15 Funcionalidades por tipo de usuario del módulo para gestionar los datos de producción que se presenta en los diferentes cultivos.

Funcionalidad	Administrador	Agricultor	Colaborador
Crear registro de producción		X	
Modificar registro de producción		X	
Consultar registros de producción		X	
Filtrar registros de producción		X	
Crear registro de ventas		X	
Modificar registro de ventas		X	
Consultar registros de ventas		X	
Filtrar registros de ventas		X	

Registrar múltiples ventas asociadas a un producción		X	
Eliminar venta		X	

Una vez identificadas las funcionalidades, se procedió a definir las historias de usuario para este módulo, en total se definieron 10 historias de usuario, planificadas para llevarse a cabo a lo largo de 2 Sprints. En la Ilustración 47 se muestran algunas de las historias de usuario definidas en Zube con sus respectivos puntos funcionales y el sprint al que pertenecen.

<input type="checkbox"/>	#34	Filtrar registros de producción	2	Done	High	Módulo de producción	Sprint #10
<input type="checkbox"/>	#98	Filtrar registros de ventas	2	Done	High	Módulo de producción	Sprint #10
<input type="checkbox"/>	#33	Consultar registros de producción	3	Done	High	Módulo de producción	Sprint #10
<input type="checkbox"/>	#95	Consultar ventas	3	Done	High	Módulo de producción	Sprint #10
<input type="checkbox"/>	#97	Eliminar venta	3	Done	High	Módulo de producción	Sprint #11
<input type="checkbox"/>	#94	Modificar registro de venta	5	Done	High	Módulo de producción	Sprint #10
<input type="checkbox"/>	#32	Modificar registro de producción	5	Done	High	Módulo de producción	Sprint #10
<input type="checkbox"/>	#31	Crear registro de producción	8	Done	High	Módulo de producción	Sprint #10
<input type="checkbox"/>	#93	Crear registro de venta	8	Done	High	Módulo de producción	Sprint #10
<input type="checkbox"/>	#96	Registrar múltiples ventas asociadas a una producción	13	Done	High	Módulo de producción	Sprint #11

Ilustración 47 Historias de usuario del módulo para la gestión de datos de producción
Fuente: Elaboración propia

11.2 Implementación

A continuación se presentan las capturas de pantalla de algunas de las funcionalidades implementadas en este módulo. La Ilustración 48 muestra la interfaz gráfica de la funcionalidad de listar los registros de producción asociados a un bloque, en la ilustración podemos observar una tabla con los registros que se encuentran en el sistema y están asociados al bloque El Paraíso1-1, estos registros se muestran con los campos de calidad de la producción, la cantidad de kilos producidos y la fecha de producción. Además de un botón de acciones por medio del cual se puede seleccionar un registro para ser modificado o para registrar las ventas asociadas a esa producción. También hay un botón que redirige al usuario para ingresar un nuevo registro de producción.

En la Ilustración 49 se muestra la interfaz gráfica de la funcionalidad de gestión de registros de ventas asociadas a un registro de producción de un bloque. En la ilustración se puede observar la información de la producción a la cual se le van a gestionar los registros de ventas, esta tiene los datos de la fecha y calidad de la producción, los kilos producidos y los kilos no vendidos basados en la diferencia entre el total de kilos y la cantidad de kilos que se han indicado como vendidos en los registros de ventas. Después

esta la información de las ventas, para adicionar un registro de venta se debe hacer clic en el botón “Agregar venta”, una vez agregada se habilita un formulario para ingresar los datos de la venta, entre ellos tenemos el comprador, la fecha de venta, la cantidad de kilos vendidos y el precio de venta por kilo, adicionalmente el sistema calcula cual fue el precio total de la venta dependiendo de los datos anteriores. Se pueden agregar las ventas que el agricultor desee y si desea quitar una debe hacer clic en el botón “Eliminar Venta” que está debajo del registro. Por ultimo también están los botones para cancelar y guardar las ventas.

LISTA DE REGISTROS DE PRODUCCIÓN
 A continuación se presenta una lista con los registros de producción del bloque El Paraiso-1-1.

10 registros por página

Calidad	Cantidad	Fecha de registro	Acciones
Selecta	55,00 kg	9 de Octubre de 2018	Acciones ▾
Industrial	60,00 kg	9 de Octubre de 2018	Acciones ▾
Industrial	100,00 kg	10 de Octubre de 2018	Acciones ▾
Corriente	200,00 kg	28 de Septiembre de 2018	Acciones ▾
Calidad	Cantidad	Fecha de registro	Acciones

Mostrando desde 1 hasta 1 de un total de 1 registros < Previous 1 Next >

REGISTRAR PRODUCCIÓN

Ilustración 48 listar registros de producción asociados a un bloque Fuente: Elaboración propia

INFORMACIÓN DE PRODUCCIÓN

Fecha de producción: 9 de Octubre de 2018 Calidad de producción: Selecta Kilos producidos: 55,00 Kilos no vendidos: 14

INFORMACIÓN DE VENTAS

Comprador: Fecha de venta: Cantidad vendida en kg: Precio de venta por kg:

Total precio de venta:

Eliminar venta

Comprador: Fecha de venta: Cantidad vendida en kg: Precio de venta por kg:

Total precio de venta:

Eliminar venta

Agregar venta

Ilustración 49 Gestión de registros de ventas de un bloque Fuente: Elaboración propia

11.3 Pruebas

Durante el desarrollo del módulo de gestión de datos de producción se realizaron pruebas funcionales del sistema. En primer lugar seleccionaron cuales serían las funcionalidades que se iban a probar, pues por cuestiones de tiempo y costo no se pudieron probar todas las funcionalidades para este módulo, se determinó que se le realizarían pruebas a un total de 8 funcionalidades. Una vez identificadas las funcionalidades se llevó a cabo el diseño de las pruebas funcionales que se iban a realizar. Posteriormente, con el apoyo de las clases identificadas se definieron los casos de prueba para la funcionalidad. Después de tener el diseño de todas las funcionalidades, el siguiente paso fue codificarlas y ejecutarlas usando la herramienta **Selenium IDE**. Con las pruebas codificadas, se procedió a agruparlas y realizar la ejecución para analizar el comportamiento de la funcionalidad.

En la Tabla 16 se muestran dos ilustraciones con el diseño y la ejecución de las pruebas funcionales para la funcionalidad de modificar un registro de producción que se encuentra definida en este módulo. En la Ilustración 50 se muestran algunos casos de prueba diseñados para esta funcionalidad y en la Ilustración 51 se puede observar la ejecución de todos los casos de prueba que fueron diseñados para esta funcionalidad. En la parte izquierda se pueden ver los casos que se ejecutaron, el color verde significa que no fallo la prueba y el color rojo significa que fallo la prueba. En la parte derecha se pueden ver los pasos que se deben seguir para ejecutar algún caso de prueba que se haya seleccionado en la parte izquierda.

Tabla 16 Diseño y ejecución de pruebas funcionales para el módulo de gestión de producción

Cantidad	Resultado esperado	Resultado de caso de prueba
50	Redireccionar a la lista de registros de producción y mostrar un mensaje en un recuadro verde con el siguiente mensaje: "Éxito! El registro de producción fue modificado exitosamente."	Fallo
50	Redireccionar a la lista de registros de producción y mostrar un mensaje en un recuadro verde con el siguiente mensaje: "Éxito! El registro de producción fue modificado exitosamente."	Fallo
50	Redireccionar a la lista de registros de producción y mostrar un mensaje en un recuadro verde con el siguiente mensaje: "Éxito! El registro de producción fue modificado exitosamente."	Fallo
50	Redireccionar a la misma pagina y mostrar un mensaje en letras rojas debajo del campo fecha, con el siguiente mensaje: "Introduzca una fecha válida."	Paso
50	Redireccionar a la misma pagina y mostrar un mensaje en letras rojas debajo del campo Fecha de producción, con el siguiente mensaje: "El campo es requerido."	Paso
50	Redireccionar a la misma pagina y mostrar un mensaje en letras rojas debajo del campo calidad, con el siguiente mensaje: "El campo es requerido."	Paso
50	Redireccionar a la misma pagina y mostrar un mensaje en letras rojas debajo del campo calidad, con el siguiente mensaje: "El campo es requerido."	Paso
50	Redireccionar a la misma pagina y mostrar un mensaje en	Paso

Ilustración 50 Casos de prueba para la funcionalidad de modificar un registro de producción

Ilustración 51 Ejecución de casos de prueba para la funcionalidad de modificar un registro de producción

Con la ejecución de las pruebas finalizada, se consignaron los resultados en el archivo de diseño de pruebas, se puede observar en el archivo adjunto "Diseños de pruebas

funcionales-Modulo de cultivos.pdf". En este documento encontramos los diseños de las pruebas y también el resultado de su ejecución.

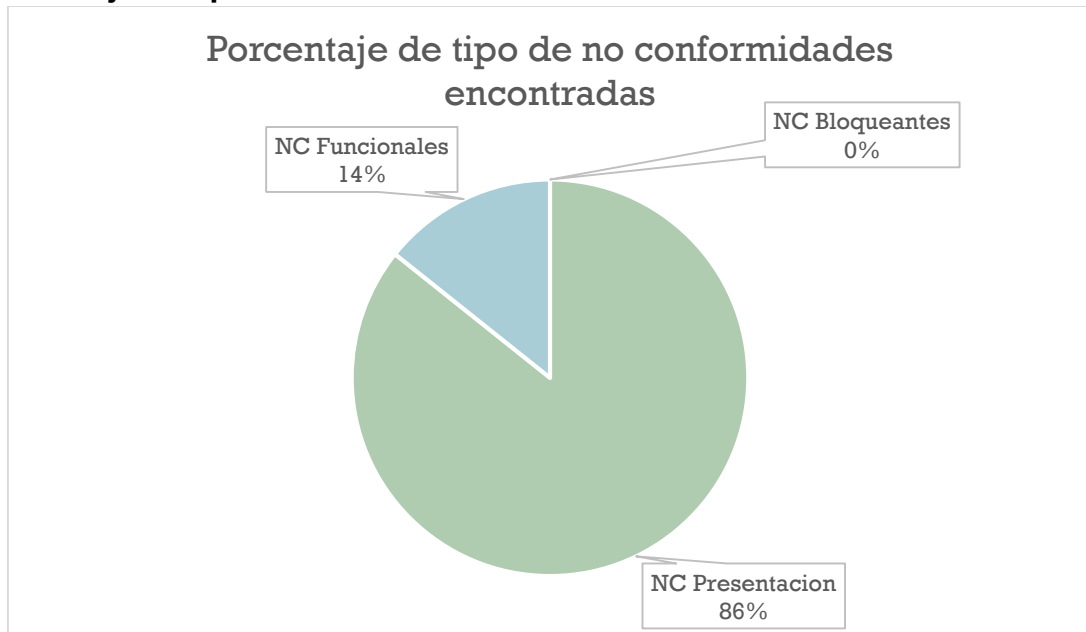
A Continuación en la Tabla 17 se muestra el resumen de ejecución de los casos pruebas que se diseñaron para este módulo, en donde se detalla por cada funcionalidad probada el número de pruebas en las cuales se identificaron no conformidades (separadas por tipo), número de diseños y por último el número de pruebas que fueron ejecutadas. En los encabezados de las columnas se reemplazaron las palabras "no conformidad" por la abreviatura "NC" y la palabra "requerimientos" por "reqs".

Tabla 17 Resumen de ejecución de casos de prueba para el módulo de gestión de producción

Funcionalidad probada	Total NC bloqueantes	Total NC funcionales	Total NC presentación	Total NC	Total reqs de prueba diseñados	Reqs de prueba ejecutados
Crear registro de producción	0	0	3	3	9	9
Modificar registro de producción	0	0	0	0	9	9
Consultar registros de producción	0	0	1	1	6	6
Crear registro de ventas	0	1	0	1	6	6
Modificar registro de ventas	0	0	1	1	6	6
Consultar registros de ventas	0	0	1	1	6	6
Eliminar venta	0	0	0	0	2	2
Totales	0	1	6	7	44	44

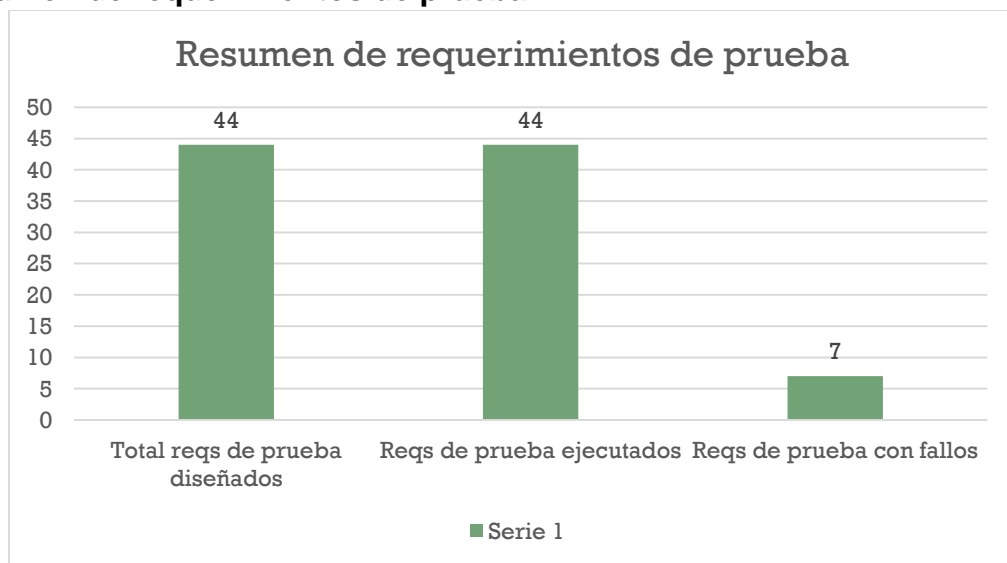
A continuación se presentan gráficos estadísticos con los datos obtenidos del resumen de ejecución.

- **Porcentaje de tipo de no conformidades encontradas**



En el gráfico anterior podemos observar el porcentaje de no conformidades encontradas según su tipo, podemos ver que no se encontró ninguna no conformidad bloqueante, lo que habla bien de los componentes que se evaluaron del sistema, pues no se detiene la ejecución de este. También podemos observar que el 14% de las no conformidades fueron funcionales y el otro 86% de presentación, esto quiere decir que la mayoría de no conformidades encontradas fueron de presentación y en este caso se debieron a mensajes mostrados de manera incorrecta o fallos en las redirecciones de las páginas.

- **Resumen de requerimientos de prueba**



En el gráfico anterior podemos observar un resumen de requerimientos de pruebas, podemos ver cuál fue el total de requerimientos de prueba diseñados y ejecutados para este módulo, además cuáles de estos presentaron fallos de algún tipo, del gráfico de barras podemos concluir que se ejecutaron la misma cantidad de requerimientos de prueba que diseños realizados, esto quiere decir que no quedaron diseños pendientes por ejecutar, también podemos observar que de 44 requerimientos de prueba ejecutado 7 tuvieron fallos de algún tipo, esto quiere decir que este módulo presento muy pocos errores en su funcionamiento y presentación al momento de realizar las pruebas.

12 Desarrollo de un módulo de reportes que permita procesar y analizar los datos de producción agrícola para el agricultor y su entorno de red de agricultores.

12.1 Análisis

Para el desarrollo del módulo de reportes que permita procesar y analizar los datos de producción agrícola para el agricultor y su entorno de red de agricultores, el primer paso fue identificar la información de importancia para el agricultor que se podía obtener con los datos registrados. Es importante que los reportes que se les muestren a los agricultores les permitan tomar decisiones críticas con respecto a diferentes aspectos de sus bloques, lotes, fincas y en general. Decisiones como la de cambiar un producto basado en el costo que le está generando en sus fincas o decidir si continuar o no trabajando con un colaborador basado en el tiempo tarda realizando diferentes labores.

Basado en lo anterior, se comenzó por realizar un resumen detallado desde la extensión de tierra más pequeña, hasta la más grande, empezando por los bloques y terminando en las fincas. En esta información detallada fue agrupada según el tipo de costo que la generaba, información general y además la información de producción y ventas.

También se determinó que los datos recolectados de los colaboradores se podían convertir en información muy útil como el hacer el seguimiento de las horas laboradas por cada labor y por cada finca, así mismo con los lotes y bloques. Esta información se muestra gráficamente y también de forma tabulada en una tabla que puede ser exportada en formato xls.

Además de lo anterior, se identificó la necesidad de informar al agricultor de las labores que están siendo empleadas en tiempo real en sus fincas, pues de esta manera se le permite monitorear la actividad en ellas.

En la Tabla 18 se presentan las funcionalidades que cada tipo de usuario podrá realizar dentro del módulo para gestionar los datos de costos y gastos de producción agrícola.

Tabla 18 Funcionalidades por tipo de usuario del módulo de reportes.

Funcionalidad	Administrador	Agricultor	Colaborador
Resumen costo general de finca		X	
Resumen de producción y ventas de bloque		X	
Resumen costo general de lote		X	

Resumen de producción y ventas de lote		X	
Resumen de producción y ventas de finca		X	
Resumen de costos por producto lote		X	
Resumen costo de comercialización de bloque		X	
Resumen de costos por producto de bloque		X	
Resumen de costos por producto de finca		X	
Resumen costo de comercialización de lote		X	
Resumen costo general de bloque		X	
Resumen costo de comercialización de finca		X	
Implementar búsqueda de rango de fechas en reportes de labores		X	
Generar reporte de tabla de labores		X	
Generar reporte de colaboradores hoy		X	
Exportar reporte de tabla de labores		X	
Detalles generales del lote		X	
Detalles generales de la finca		X	
Dashboard		X	
Generar gráficos con información de horas laboradas por finca, lote y bloque		X	
Generar reporte de horas empleadas por colaboradores en labores		X	
Generar reporte básico		X	
Detalles generales del bloque		X	

Una vez identificadas las funcionalidades, se procedió a definir las historias de usuario para este módulo, en total se definieron 23 historias de usuario, planificadas para llevarse a cabo a lo largo de 3 Sprints. En la se muestran algunas de las historias de usuario definidas en Zube con sus respectivos puntos funcionales y el sprint al que pertenecen.

<input type="checkbox"/>	#117	Resumen costo general de finca	2	Done	J	Modulo de reportes	Sprint #14
<input type="checkbox"/>	#118	Resumen de producción y ventas de bloque	2	Done	J	Modulo de reportes	Sprint #12
<input type="checkbox"/>	#116	Resumen costo general de lote	2	Done	J	Modulo de reportes	Sprint #13
<input type="checkbox"/>	#119	Resumen de producción y ventas de lote	2	Done	J	Modulo de reportes	Sprint #13
<input type="checkbox"/>	#120	Resumen de producción y ventas de bloque	2	Done	J	Modulo de reportes	Sprint #14
<input type="checkbox"/>	#111	Resumen de costos por producto lote	3	Done	J	Modulo de reportes	Sprint #13
<input type="checkbox"/>	#112	Resumen costo de comercialización de bloque	3	Done	J	Modulo de reportes	Sprint #12
<input type="checkbox"/>	#109	Resumen de costos por producto de bloque	3	Done	J	Modulo de reportes	Sprint #12
<input type="checkbox"/>	#110	Resumen de costos por producto de finca	3	Done	J	Modulo de reportes	Sprint #14
<input type="checkbox"/>	#113	Resumen costo de comercialización de lote	3	Done	J	Modulo de reportes	Sprint #13
<input type="checkbox"/>	#115	Resumen costo general de bloque	3	Done	J	Modulo de reportes	Sprint #12
<input type="checkbox"/>	#114	Resumen costo de comercialización de finca	3	Done	J	Modulo de reportes	Sprint #14
<input type="checkbox"/>	#107	Implementar búsqueda de rango de fechas en reportes de labores:	8	Done	J	Modulo de reportes	Sprint #13
<input type="checkbox"/>	#105	Generar reporte de tabla de labores	8	Done	J	Modulo de reportes	Sprint #13
<input type="checkbox"/>	#108	Generar reporte de colaboradores hoy	13	Done	J	Modulo de reportes	Sprint #13

Ilustración 52 Historias de usuario del módulo de reportes
Fuente: Elaboración propia Implementación

12.2 Implementación

A continuación se presentan las capturas de pantalla de algunas de las funcionalidades implementadas en este módulo. La Ilustración 53 muestra la interfaz gráfica de la funcionalidad de generar el dashboard, en la ilustración podemos observar que en la pantalla principal al agricultor se le va a mostrar una gráfica de líneas donde dado un año seleccionado, se muestren los costos generados por cada finca durante todos los meses del respectivo año. Después podemos ver datos de interés generales para el agricultor, como cuál es el producto que más costos ha generado en todas sus fincas o cual es el total de costos que han generado todas sus fincas. Además también se muestran dos bloques, uno hace referencia al agricultor y el otro a la aplicación, es decir todos los agricultores registrados, en esta sección un agricultor puede comparar por ejemplo cuales son los 5 productos que más costos han generado en sus fincas con los 5 productos que más costos han generado en todas las fincas registradas en el sistema.



Ilustración 53 Funcionalidad de generar dashboard
Fuente: Elaboración propia

La Ilustración 54 muestra la interfaz gráfica de la funcionalidad de generar los detalles generales de un bloque, en la ilustración podemos observar se le muestra al agricultor un consolidado con la suma de todos los gastos y costos generados, agrupados según su tipo: totales, por producto, de comercialización y otros. Además se muestra esta información usando dos tipos de grafico diferentes para una mejor visualización de la información. En el primer grafico de torta se comparan todos los costos según su tipo y se indica cual es el porcentaje correspondiente del total de costos. El otro grafico muestra los mismos datos pero en forma de barras, donde es fácil observar cual ha sido el tipo de costo que más dinero ha generado, adicionalmente este grafico se puede ordenar de manera descendente para una mejor comparación.

En la Ilustración 55 se muestra la interfaz gráfica de la funcionalidad de generar los detalles de producción y ventas de un bloque, en la ilustración se pueden observar dos gráficos de torta, en uno se muestra la información de producción agrupada según su calidad, que puede ser corriente, selecta o industrial. En el otro grafico se muestra la información de ventas agrupada según su calidad. Posteriormente encontramos un calendario para producción y otro para ventas, donde se pueden ver los registros realizados en sus respectivas fechas.



Ilustración 54 Funcionalidad de generar detalles generales de un bloque
Fuente: Elaboración propia

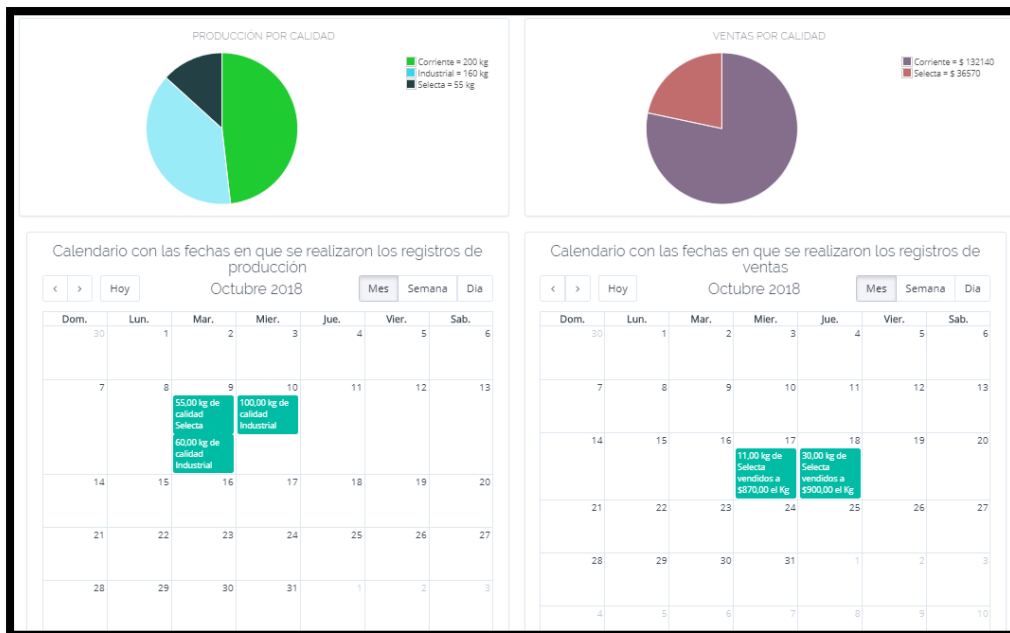


Ilustración 55 Funcionalidad generar detalles de producción y ventas de un bloque
Fuente: Elaboración propia

12.3 Pruebas

Durante el desarrollo del módulo de gestión de datos de producción se realizaron pruebas funcionales del sistema. En primer lugar seleccionaron cuales serían las funcionalidades que se iban a probar, pues por cuestiones de tiempo y costo no se pudieron probar todas las funcionalidades para este módulo, se determinó que se le realizarían pruebas a un total de 13 funcionalidades. Una vez identificadas las funcionalidades se llevó a cabo el

diseño de las pruebas funcionales que se iban a realizar. Posteriormente, con el apoyo de las clases identificadas se definieron los casos de prueba para la funcionalidad. Después de tener el diseño de todas las funcionalidades, el siguiente paso fue codificarlas y ejecutarlas usando la herramienta *Selenium IDE*. Con las pruebas codificadas, se procedió a agruparlas y realizar la ejecución para analizar el comportamiento de la funcionalidad.

En la Tabla 16 se muestran dos ilustraciones con el diseño y la ejecución de las pruebas funcionales para la funcionalidad de detalles generales del bloque que se encuentra definida en este módulo. En la Ilustración 56 se muestran algunos casos de prueba diseñados para esta funcionalidad y en la Ilustración 57 se puede observar la ejecución de todos los casos de prueba que fueron diseñados para esta funcionalidad. En la parte izquierda se pueden ver los casos que se ejecutaron, el color verde significa que no fallo la prueba y el color rojo significa que fallo la prueba. En la parte derecha se pueden ver los pasos que se deben seguir para ejecutar algún caso de prueba que se haya seleccionado en la parte izquierda.

Con la ejecución de las pruebas finalizada, se consignaron los resultados en el archivo de diseño de pruebas, se puede observar en el archivo adjunto “Diseños de pruebas funcionales-Modulo de reportes.pdf”. En este documento encontramos los diseños de las pruebas y también el resultado de su ejecución.

A Continuación en la Tabla 17 se muestra el resumen de ejecución de los casos pruebas que se diseñaron para este módulo, en donde se detalla por cada funcionalidad probada el número de pruebas en las cuales se identificaron no conformidades (separadas por tipo), número de diseños y por último el número de pruebas que fueron ejecutadas. En los encabezados de las columnas se reemplazaron las palabras “no conformidad” por la abreviatura “NC” y la palabra “requerimientos” por “reqs”.

Funcionalidad probada	Total NC bloqueantes	Total NC funcionales	Total NC presentación	Total NC	Total reqs de prueba diseñados	Reqs de prueba ejecutados
Resumen costo general de finca	0	0	0	0	8	8
Resumen de producción y ventas de bloque	0	1	1	2	8	8
Resumen de producción y ventas de finca	0	1	1	2	8	8
Resumen costo de comercialización de bloque	0	0	2	2	8	8
Resumen de costos por producto de bloque	0	1	4	5	15	15

Resumen de costos por producto de finca	0	1	4	5	15	15
Resumen costo general de bloque	0	0	0	0	8	8
Resumen costo de comercialización de finca	0	0	2	2	8	8
Generar reporte de tabla de labores	0	0	1	1	6	6
Generar reporte de colaboradores hoy	0	0	1	1	6	6
Detalles generales de la finca	0	0	4	4	12	12
Dashboard	0	3	3	6	18	18
Detalles generales del bloque	0	0	4	4	12	12
Totales	0	7	27	32	132	132

Tabla 19 Diseño y ejecución de pruebas funcionales para el módulo de reportes

Funcionalidad:	Resumen costo general de finca											
	1	2	3	4	5	6	7	8	9	10	11	12
Usuario tiene registros de costos por producto	S	S	S	S	N	N	N	N	N	N	N	N
Usuario tiene registros de costos de comercialización	S	N	S	S	S	N	S	S	N	N	N	N
Usuario tiene registros de otros costos	S	S	N	S	S	S	N	S	S	N	S	N
Usuario tiene registros de costos por labor	S	S	S	N	S	S	S	N	S	S	N	N
Resumen con graficos de todos los costos	X	-	-	-	-	-	-	-	-	-	-	-
Resumen sin graficos de costos	-	-	-	-	-	-	-	-	-	-	-	X
Resumen con graficos de todos los costos excepto los costos por producto	-	-	-	-	X	X	X	X	X	X	X	-
Resumen con graficos de todos los costos excepto los costos por labor	-	-	-	X	-	-	-	X	-	-	X	-
Resumen con graficos de todos los costos excepto los costos de comercialización	-	X	-	-	-	X	-	-	X	X	X	-
Resumen con graficos de todos los costos excepto los costos por labor	-	-	-	-	-	-	-	-	-	-	-	-
Resumen con graficos de todos los costos excepto los otros costos	-	-	X	-	-	-	X	X	-	X	-	-
Paso	S	S	S	S	S	S	S	S	N	N	N	N

Ilustración 56 Casos de prueba para la funcionalidad de detalles generales del bloque

The screenshot shows the Selenium IDE interface. The title bar reads 'Detalles generales del bloque12 (Detalles generales del bloque) - Selenium IDE 2.9.1'. The 'Test Case' pane on the left lists 11 test cases, with 'Detalles generales del bloque11' selected. The 'Table' pane on the right shows the following test data:

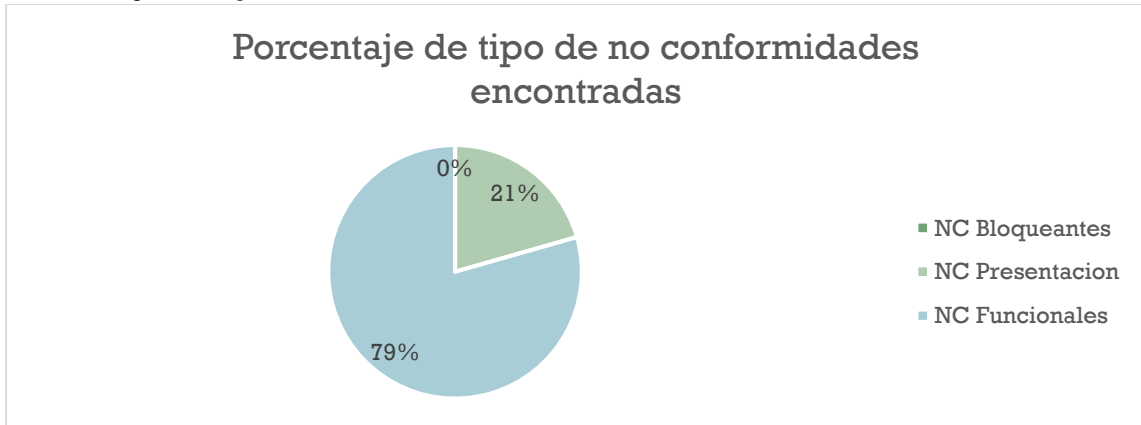
Command	Target	Value
open	https://agros1-juanda...	
verifyNotText	//div[@id='general']/d...	0
verifyNotText	//div[@id='general']/d...	0
verifyNotText	//div[@id='general']/d...	0
verifyElementPresent	css=td.legendLabel	
verifyElementPresent	//div[@id='data-donut...	
verifyElementPresent	//div[@id='data-donut...	
verifyElementPresent	//div[@id='data-donut...	
verifyElementPresent	//div[@id='data-donu...	

At the bottom, the 'Log' pane shows 'open(url)' with the argument 'url: the URL to open: must be relative or absolute'.

Ilustración 57 Ejecución de casos de prueba para la funcionalidad de detalles generales del bloque

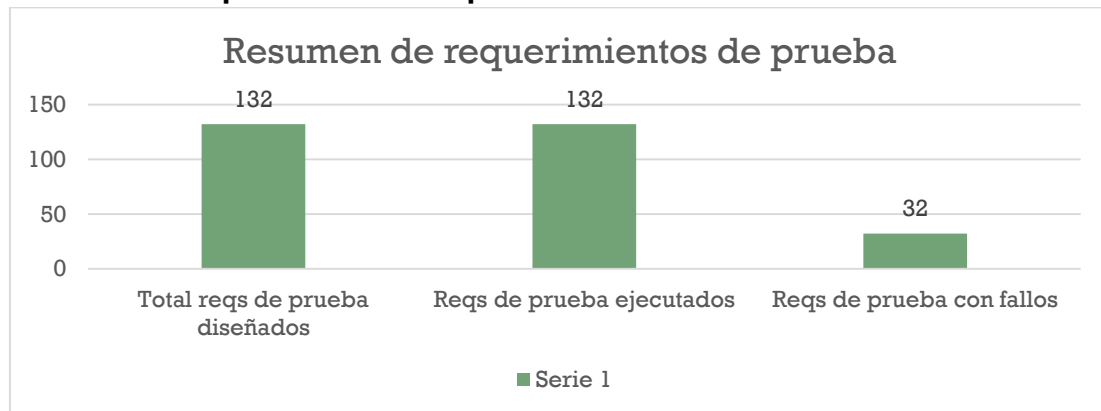
A continuación se presentan gráficos estadísticos con los datos obtenidos del resumen de ejecución.

- **Porcentaje de tipo de no conformidades encontradas**



En el gráfico anterior podemos observar el porcentaje de no conformidades encontradas según su tipo, podemos ver que no se encontró ninguna no conformidad bloqueante, lo que habla bien de los componentes que se evaluaron del sistema, pues no se detiene la ejecución de este. También podemos observar que el 21% de las no conformidades fueron funcionales y el otro 79% de presentación, esto quiere decir que la mayoría de no conformidades encontradas fueron de funcionales y en este caso se debieron a malos formatos y datos enviados o recibidos a la hora de generar gráficos o tablas.

- **Resumen de requerimientos de prueba**



En el gráfico anterior podemos observar un resumen de requerimientos de pruebas, podemos ver cuál fue el total de requerimientos de prueba diseñados y ejecutados para este módulo, además cuáles de estos presentaron fallos de algún tipo, del gráfico de barras podemos concluir que se ejecutaron la misma cantidad de requerimientos de prueba que diseños realizados, esto quiere decir que no quedaron diseños pendientes por ejecutar, también podemos observar que de 132 requerimientos de prueba ejecutado 32 tuvieron fallos de algún tipo, esto quiere decir que este módulo presentó un número considerable de errores en su funcionamiento y presentación al momento de realizar las pruebas. Es importante resaltar que este fue el módulo que más fallos presentó en la ejecución de sus pruebas funcionales.

13 Conclusiones

La recolección de datos de muestras de campo en el sector agrícola es compleja y costosa, pues los agricultores son convocados pero responden con desinterés, debido a que la utilización de la tecnología puede cambiar su forma de realizar algunos procesos. Es importante concientizar a los empleados y empleadores que se desempeñan en este sector, que la tecnología no es algo que este lejos de su alcance y que les va a permitir mejorar en cuanto a su organización y producción.

Este documento presenta el desarrollo de una aplicación web/móvil que permite hacer el seguimiento del proceso de planeación y producción agrícola de una red de agricultores. En específico, la aplicación gestiona los gastos y costos así como también permite generar reportes que le permiten al agricultor tener una visión crítica del comportamiento de la producción en sus propios lotes. Esta aplicación puede ser accedida desde cualquier lugar y mediante diversos dispositivos ya que es una aplicación web/móvil.

Al finalizar la realización de la aplicación se logró el cumplimiento de todos los objetivos específicos planteados al inicio, pues se desarrollaron seis módulos, uno para a gestión de usuarios, otros para la gestión de lotes y cultivos, adicionalmente dos más para gestionar los costos y la producción. Por último se desarrolló un módulo de reportes para que el agricultor accediera a la información de forma gráfica. Habiendo logrado desarrollar los módulos para el cumplimiento de los objetivos específicos, podemos concluir que se logró también el cumplimiento del objetivo general ya que se desarrolló exitosamente una aplicación web para el control de producción agrícola.

Además del cumplimiento de los objetivos propuestos, también se realizaron pruebas de campo con el apoyo de un agricultor de nombre José David Palma, quien después de evaluar el funcionamiento de la aplicación indicó que está supero sus expectativas, pues según su opinión por sus interfaces intuitivas es muy fácil de usar. Además resalto la utilidad que tiene el software biométrico auxiliar, que le permitió llevar un mejor control con respecto a las labores que se realizaron en sus lotes.

Como trabajo extra se desarrolló un programa auxiliar biométrico en Java para permitir a los agricultores registrar las huellas dactilares de sus colaboradores y así realizar registros más precisos de las labores y así generar de mejor forma los costos por la realización de las labores que se realizan en los lotes.

Dentro de las competencias adquiridas en el transcurso del programa de Ingeniería de Sistemas que se utilizaron para el desarrollo de este trabajo de grado, están las que fueron adquiridas desde las materias básicas de programación como fundamentos de programación o programación orientada a objetos, hasta las de desarrollo de software incluidas entre ellas el diseño web, el diseño de bases de datos, el diseño de interfaces de usuario y pruebas de software. Además de las asignaturas complementarias como el Inglés para la lectura de artículos e investigaciones y las de administración de proyectos para la estimación de recursos y tiempo para la realización de la plataforma.

Se adquirieron nuevos conocimientos en el sector agrícola (organización, proceso de producción y costeo), en la utilización de sistemas de información geo referenciada, en desarrollo de software utilizando hardware biométrico y en visualización de información utilizando librerías especializadas como D3.js.

Es interesante conocer y explorar de qué manera la ingeniería de sistemas puede apoyar otros sectores, en este caso a sector agrícola. La interdisciplinariedad de este proyecto me permitió acercarme al sector agrícola y además de adquirir conocimiento, también se presentó la oportunidad de compartir el conocimiento adquirido en mi carrera. Además este trabajo resaltó la utilidad que tiene el desarrollo de software aplicado con otras áreas de conocimiento.

13.1 Trabajo futuro

Queda como trabajo futuro continuar profundizando en el manejo de costos del proceso de producción agrícola y así mejorar la gestión de los datos. Por ejemplo cuando se mezclan varios productos para realizar una aplicación múltiple en diferentes bloques.

Implementar el software auxiliar de biometría para que pueda ser utilizado en dispositivos móviles, pues permitirían hacer el registro de labores de una manera más práctica y eficiente.

14 Referencias

- Abrantes, J. F., & Travassos, G. H. (2011). Common Agile Practices in Software Processes. *2011 International Symposium on Empirical Software Engineering and Measurement*, 355–358. <https://doi.org/10.1109/ESEM.2011.47>
- Adrian Holovaty, J. K.-M. (2008). El libro de Django 1.0. Retrieved October 31, 2018, from https://librosweb.es/libro/django_1_0/
- Agile Alliance. (2017). Agile Alliance. Retrieved October 14, 2017, from <https://www.agilealliance.org/>
- Agroptima. (2017). Software y Aplicación de gestión agrícola fácil de usar | Agroptima®. Retrieved October 17, 2017, from <https://www.agroptima.com/>
- Amazon Web Services. (2018). AWS Cloud9 – Amazon Web Services. Retrieved October 31, 2018, from <https://aws.amazon.com/es/cloud9/>
- Beck, K., Beedle, M., & Bennekum, A. van. (2001). Manifiesto Agil. Retrieved October 14, 2017, from <http://agilemanifesto.org/iso/es/manifesto.html>
- Bootstrap. (2017). Bootstrap · The most popular HTML, CSS, and JS library in the world. Retrieved November 28, 2017, from <https://getbootstrap.com/>
- DigitalPersona Inc. (2010). One Touch ® for Windows SDK, 119.
- Django Software Foundation. (2017a). GeoDjango | Django documentation | Django. Retrieved November 27, 2017, from <https://docs.djangoproject.com/en/1.11/ref/contrib/gis/>
- Django Software Foundation. (2017b). The Web framework for perfectionists with deadlines | Django. Retrieved November 11, 2017, from <https://www.djangoproject.com/>
- EcuRed. (2018). Producción agrícola. Retrieved November 10, 2018, from https://www.ecured.cu/Producción_agrícola
- EL TIEMPO. (2000). CON EL REGISTRO, CUENTAS CLARAS EN LA FINCA. *ELTIEMPO*.
- ESRI. (2017). ArcGIS | Principal. Retrieved November 27, 2017, from <https://www.arcgis.com/features/index.html>
- Gobierno de España. (2016). Visualización de datos, (C).
- González, A. B. (2005). *Gestión de Proyectos de Software*.

- Google. (2017). Material Design. Retrieved November 28, 2017, from <https://material.io/>
- InSoft. (2017). Software Agrícola - AgroWin. Retrieved September 24, 2017, from <http://www.agrowin.com/>
- Mora, S. L. (2013). Programación de aplicaciones web: historia, principios básicos y clientes web. *Editorial Club Universitario.*, 53(9), 1689–1699. <https://doi.org/10.1017/CBO9781107415324.004>
- Mozilla Developer Network. (2017). Documentación web de MDN. Retrieved November 28, 2017, from <https://developer.mozilla.org/es/>
- Oracle Corporation. (2018a). ¿Qué es Java y para qué es necesario? Retrieved October 3, 2018, from https://www.java.com/es/download/faq/whatis_java.xml
- Oracle Corporation. (2018b). NetBeans IDE. Retrieved October 31, 2018, from <https://netbeans.org/features/index.html>
- Osorio, O. (1995). Los Costos Y Las Decisiones En Agricultura Una Actividad Olvidada. *Congreso Internacional De Custos.* Retrieved from <http://intercostos.org/documentos/TEX10-14.pdf>
- pgadmin.org. (2018). pgAdmin - PostgreSQL Tools. Retrieved October 31, 2018, from <https://www.pgadmin.org/>
- Rojas, K. J. (2016). La agricultura colombiana en el contexto de la globalización. Retrieved September 5, 2017, from <http://www.elcampesino.co/la-agricultura-colombiana-en-el-contexto-de-la-globalizacion/>
- Schwaber, K., & Sutherland, J. (2013). La Guía de Scrum. *Scrumguides.Org*, 1, 21. Retrieved from <http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-ES.pdf>
- seleniumhq.org. (2018). Selenium - Web Browser Automation. Retrieved November 7, 2018, from <https://www.seleniumhq.org/>
- Socialmood. (2017). ¿Qué es el Diseño Responsive? Retrieved November 28, 2017, from <https://www.40defiebre.com/que-es/disenio-responsive/>
- SynAgro. (2016). Synagro - Software Agropecuario. Retrieved October 16, 2017, from <http://synagroweb.com/>
- The jQuery Foundation. (2017). jQuery. Retrieved November 28, 2017, from <https://jquery.com/>
- Veléz Quiroga, S. N. (2011). Herramienta Para Control Y Programación De La Producción En El Sector Agrícola.
- Vertabelo. (2016). Vertabelo. Retrieved November 9, 2018, from

<https://my.vertabelo.com/drive>

W3C. (2017). HTML & CSS - W3C. Retrieved November 28, 2017, from <https://www.w3.org/standards/webdesign/htmlcss.html>

Zhao, A., Xie, F., Zhu, J., Tang, Q., Zhang, G., Xiong, L., & Gong, J. (2010). Assessment of inquiry-based learning geographic course - A case study of "GIS introduction." *2010 International Conference on Computational Intelligence and Software Engineering, CiSE 2010*. <https://doi.org/10.1109/CISE.2010.5677160>

Zube. (2018). Retrieved October 3, 2018, from <https://zube.io/>