

UNIVERSIDAD TÉCNICA DEL NORTE



Facultad de Ingeniería en Ciencias Aplicadas

Carrera de Ingeniería en Sistemas Computacionales

**DESARROLLO DE UN SISTEMA WEB PARA EL CONTROL DE LA
PRODUCCIÓN DE UN INVERNADERO DE TOMATE EN EL CANTÓN COTACACHI,
UTILIZANDO EL FRAMEWORK ANGULAR VERSIÓN 4**

**Trabajo de grado previo a la obtención del Título de Ingeniero en
Sistemas Computacionales**

Autor: Carlos Fernando Proaño Yépez

Director: Ing. Antonio Quiña

Ibarra – Ecuador

2019



UNIVERSIDAD TÉCNICA DEL NORTE
BIBLIOTECA UNIVERSITARIA

AUTORIZACIÓN DE USO Y PUBLICACIÓN

A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA

La Universidad Técnica del Norte dentro del proyecto repositorio Digital Institucional, determinó la necesidad de disponer de textos completos en formato digital con la finalidad de apoyar los procesos de investigación, docencia y extensión de la Universidad.

Por medio del presente documento, dejo sentada mi voluntad de participar en este proyecto, para lo cual ponemos a disposición la siguiente información.

DATOS DE CONTACTO	
CÉDULA DE IDENTIDAD	100362669-2
APELLIDOS Y NOMBRES	PROAÑO YÉPEZ CARLOS FERNANDO
DIRECCIÓN	Cotacachi, La Calera, Calle Nro. 4
EMAIL	carlosproanio@outlook.com
TELÉFONO FIJO	062916593
TELÉFONO MÓVIL	0982210740
DATOS DE LA OBRA	
TÍTULO	DESARROLLO DE UN SISTEMA WEB PARA EL CONTROL DE LA PRODUCCIÓN DE UN INVERNADERO DE TOMATE EN EL CANTÓN COTACACHI, UTILIZANDO EL FRAMEWORK ANGULAR VERSIÓN 4
AUTOR	PROAÑO YÉPEZ CARLOS FERNANDO
FECHA	08/02/2019
TÍTULO POR EL QUE OPTA	INGENIERÍA EN SISTEMAS COMPUTACIONALES
DIRECTOR	MSc. ANTONIO QUIÑA

2. AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD

Yo, Carlos Fernando Proaño Yépez, con cédula de identidad Nro. 100362669-2, en calidad de autor y titular de los derechos patrimoniales de la obra o Trabajo de Grado descrito anteriormente, hago entrega del ejemplar respectivo en forma digital y autorizo a la Universidad Técnica del Norte, la publicación de la obra en el Repositorio Digital Institucional y el uso del archivo digital en la biblioteca de la Universidad con fines académicos, para ampliar la disponibilidad del material y como apoyo a la educación, investigación y extensión, en concordancia con la Ley de Educación Superior Artículo 144.

3. CONSTANCIAS

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

EL AUTOR:



Nombre: Carlos Proaño

Cédula: 100362669-2

Ibarra, Febrero de 2019



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS
CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE INVESTIGACIÓN
A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

Yo, Carlos Fernando Proaño Yépez, con cédula de identidad Nro. 100362669-2, manifiesto mi voluntad de ceder a la Universidad Técnica del Norte los derechos patrimoniales consagrados en la ley de propiedad intelectual del Ecuador, artículo 4, 5 y 6 en calidad de autor del Trabajo de Grado denominado: “DESARROLLO DE UN SISTEMA WEB PARA EL CONTROL DE LA PRODUCCIÓN DE UN INVERNADERO DE TOMATE EN EL CANTÓN COTACACHI, UTILIZANDO EL FRAMEWORK ANGULAR VERSIÓN 4”, que ha sido desarrollado para optar el título de Ingeniería en Sistemas Computacionales, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En mi condición de autor me reservo los derechos morales de la obra antes mencionada. En concordancia suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la biblioteca de la Universidad Técnica de Norte de la ciudad de Ibarra.

EL AUTOR:

Nombre: Carlos Proaño

Cédula: 100362669-2

Ibarra, Febrero de 2019

CERTIFICADO DEL DIRECTOR DE TRABAJO DE GRADO

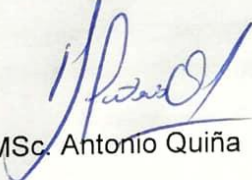


UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CERTIFICADO DEL DIRECTOR DE TRABAJO DE GRADO

Certifico que el trabajo de grado titulado "DESARROLLO DE UN SISTEMA WEB PARA EL CONTROL DE LA PRODUCCIÓN DE UN INVERNADERO DE TOMATE EN EL CANTÓN COTACACHI, UTILIZANDO EL FRAMEWORK ANGULAR VERSIÓN 4", ha sido realizado en su totalidad con interés profesional y responsabilidad por el señor Carlos Fernando Proaño Yépez, con cédula de identidad Nro. 100362669-2; previo a la obtención del título de Ingeniería en Sistemas Computacionales.


MSc. Antonio Quiña

Director de Tesis

MSc. Antonio Quiña
DOCENTE FICA UTN

CERTIFICACIÓN DE LA EMPRESA AUSPICIANTE



Finca Agrícola del Milenio

COTACACHI - ECUADOR

CERTIFICACIÓN DE AUSPICIO

Ibarra, 28 de enero del 2019

Señores
UNIVERSIDAD TÉCNICA DEL NORTE
Presente

De nuestras consideraciones.-

Siendo auspiciantes del proyecto de Tesis del Señor Egresado **CARLOS FERNANDO PROAÑO YÉPEZ** con cédula **1003626692** quien desarrollo su trabajo de titulación con el tema **"DESARROLLO DE UN SISTEMA WEB PARA EL CONTROL DE LA PRODUCCIÓN DE UN INVERNADERO DE TOMATE EN EL CANTÓN COTACACHI, UTILIZANDO EL FRAMEWORK ANGULAR VERSIÓN 4"**, nos es grato informar que se ha superado a satisfacción las pruebas técnicas y la revisión del cumplimiento de requerimientos funcionales, por lo que se recibe el proyecto como culminado por parte del señor. Una vez recibida la capacitación y documentación respectiva nos comprometemos a continuar utilizando el aplicativo en beneficio de nuestra institución.

El señor egresado **CARLOS FERNANDO PROAÑO YÉPEZ** puede hacer uso del presente documento como estime conveniente dentro del marco legal.

Atentamente,

Sr. Milton Vallejo

CI: 1002578159

PROPIETARIO



Dirección: La Calera, Calle Nro.4

Teléfono: 062916593 - 0961629291

DEDICATORIA

Este trabajo se los dedico enteramente a mis padres quienes cada día me han dado su apoyo y motivación para culminar mi vida profesional. A mi madre por darme el impulso necesario para incursionar en una carrera profesional y por creer en mí y en mis capacidades para terminarla. A mi padre por ayudarme incondicionalmente, sin importarle que sea día o noche, siempre estuvo dispuesto a brindarme su apoyo.

Carlos

AGRADECIMIENTO

Agradezco a mis padres por ayudarme incondicionalmente en buenos y malos momentos, a su oportuno consejo y brindarme la oportunidad de tener una carrera profesional.

A mis hermanos por haberme tenido tanta paciencia durante toda la carrera.

A los profesores que a lo largo de la vida universitaria, por compartir sus experiencias y conocimientos. Y de una manera muy especial agradezco a mi tutor Antonio Quiña y a mi asesor Fausto Salazar, que gracias a su valiosa ayuda logre culminar este trabajo.

A mi novia Eliana que durante toda la carrera me supo apoyar con mucha paciencia y tolerancia.

A mi gran amigo Alexander Orbes quien fue que me apoyo en desarrollo de la aplicación.

Carlos

RESUMEN

El presente trabajo de grado comprende el desarrollo de una aplicación web enfocada en la gestión de procesos dentro de la Finca Agrícola del Milenio, con el fin de usar la tecnología como un soporte para las labores ejecutadas diariamente, disminuyendo así el consumo de tiempo.

En el primer capítulo expone la situación actual de la finca y de la metodología de desarrollo y características de herramientas a ser utilizadas.

El segundo capítulo se detalla el desarrollo de la aplicación a través de las fases establecidas por la metodología SCRUM.

En el tercer capítulo se muestra los resultados obtenidos tras la implantación del sistema.

ABSTRACT

The present work of degree comprises the development of a web application focused on the management of processes within the Agricultural Millennium Farm, in order to use technology as a support for the tasks performed daily, thus decreasing the consumption of time.

In the first chapter he exposes the current situation of the farm and the methodology of development and characteristics of tools to be used.

The second chapter details the development of the application through the phases established by the SCRUM methodology.

In the third chapter the results obtained after the implementation of the system are shown.

TABLAS DE CONTENIDO

AUTORIZACIÓN DE USO Y PUBLICACIÓN.....	I
CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE INVESTIGACIÓN	III
CERTIFICADO DEL DIRECTOR DE TRABAJO DE GRADO	IV
CERTIFICACIÓN DE LA EMPRESA AUSPICIANTE	V
DEDICATORIA	VI
AGRADECIMIENTO	VII
RESUMEN	VIII
ABSTRACT	IX
TABLAS DE CONTENIDO.....	X
ÍNDICE DE FIGURAS.....	XV
ÍNDICE DE TABLAS.....	XVIII
INTRODUCCIÓN.....	1
ANTECEDENTES.....	1
SITUACIÓN ACTUAL	2
PROSPECTIVA	3
PLANTEAMIENTO DEL PROBLEMA	4
OBJETIVOS.....	4
1. Objetivo General.....	4
2. Objetivos Específicos.....	4
ALCANCE	5
1. Módulo de gestión de usuario	5
2. Módulo de inventario de materiales e insumos	5
3. Módulo de control de producción	5
4. Módulo de calendarios.....	5
5. Módulo de control de químicos	6

6.	Módulo de consultas y reportes	6
	JUSTIFICACIÓN	6
	CAPÍTULO 1.....	7
1.	MARCO TEÓRICO	7
1.1	ACTIVIDAD AGRÍCOLA.....	7
1.2	MICROEMPRESAS AGRÍCOLAS EN COTACACHI	7
1.3	SISTEMAS DE CONTROL DE PRODUCCIÓN.....	8
1.3.1	¿Qué es un sistema de control de producción?.....	8
1.3.2	Actividades guía del Justo a Tiempo	8
1.3.3	Control de la producción informático	10
1.3.4	¿Qué son sistemas web?.....	11
1.3.5	Ventajas de las aplicaciones web.....	11
1.3.6	Desventajas de las aplicaciones web	12
1.4	ENTORNO DE DESARROLLO	13
1.4.1	Base de datos MySQL	13
1.4.2	Framework Angular.....	14
	Arquitectura de una aplicación en Angular 4.....	14
	• Módulo	14
	• Componente.....	15
	• Template	15
	• Metadatos	15
	• Data Binding.....	17
	• Directiva	17
	• Servicios	17
	• Dependency Injection.....	17
1.4.3	Framework SpringBoot.....	18
	• Características	18

• ¿Cómo crear un proyecto SpringBoot?	19
1.4.4 Eclipse	19
• Arquitectura de Eclipse	20
1.5 METODOLOGÍA DE DESARROLLO DE SOFTWARE.....	21
1.5.1 Metodología de desarrollo ágil	21
Ventajas de utilizar las metodologías ágiles	21
1.5.2 Metodología SCRUM	21
1.5.3 ¿Cómo funciona SCRUM?	22
1.5.4 Reuniones de SCRUM.....	23
1.5.5 Roles de SCRUM.....	24
CAPÍTULO 2.....	26
2. DESARROLLO DE LA APLICACIÓN.....	26
2.1 ROLES DE SCRUM.....	26
2.2 REQUERIMIENTOS Y PLANIFICACIÓN DE SPRINTS	26
2.2.1 Historias de usuario.....	26
2.2.2 Product backlog	35
2.2.3 Planificación del proyecto.....	37
2.3 EJECUCIÓN	37
2.3.1 Sprint 0 Análisis y planificación del proyecto	38
a. Planificación del sprint	38
b. Revisión del sprint	39
c. Revisión retrospectiva.....	42
2.3.2 Sprint 1 Gestión de Plantas e Invernaderos	42
a. Planificación del sprint	42
b. Revisión del sprint	45
c. Revisión retrospectiva.....	49
2.3.3 Sprint 2 Gestión de Insumos	50
a. Planificación del sprint	50

b.	Revisión del sprint	52
c.	Revisión retrospectiva.....	56
2.3.4	Sprint 3 Gestión de Tratamientos y Mantenimientos	57
a.	Planificación del sprint	57
b.	Revisión del sprint	60
c.	Revisión retrospectiva.....	65
2.3.5	Sprint 4 Gestión de Siembras y Tratamientos Aplicados	66
a.	Planificación del sprint	66
b.	Revisión del sprint	69
c.	Revisión retrospectiva.....	73
2.3.6	Sprint 5 Gestión de Cosechas y Ventas	74
a.	Planificación del sprint	74
b.	Revisión del sprint	77
c.	Revisión retrospectiva.....	80
2.3.7	Sprint 6 Gestión del Dashboard	81
a.	Planificación del sprint	81
b.	Revisión del sprint	82
c.	Revisión retrospectiva.....	83
2.3.8	Sprint 7 Gestión de Acceso	84
a.	Planificación del sprint	84
b.	Revisión del sprint	85
c.	Revisión retrospectiva.....	86
CAPÍTULO 3.....		87
3.	RESULTADOS	87
3.1	PRUEBAS DE ACEPTACIÓN	87
3.2	BENEFICIOS DE LA IMPLEMENTACIÓN.....	88
3.2.1	Indicadores de beneficios.....	88
3.3	ANÁLISIS DE IMPACTOS.....	90

3.3.1	Impacto socio cultural.....	91
3.3.2	Impacto económico.....	91
3.3.3	Impacto ambiental.....	92
3.4	CONCLUSIONES Y RECOMENDACIONES.....	92
	• Conclusiones.....	92
	• Recomendaciones.....	93
	REFERENCIAS.....	94

ÍNDICE DE FIGURAS

Figura 1 Árbol de Problemas	4
Figura 2 Metadatos en Angular	16
Figura 3 Dependencias en Angular	18
Figura 4 Creación de proyectos de SpringBoot.....	19
Figura 5 Proceso de la Metodología de SCRUM.....	22
Figura 6 Roles de SCRUM.....	25
Figura 7 Arquitectura del Proyecto	39
Figura 8 Diagrama Entidad Relación	40
Figura 9 Diagrama de procesos	41
Figura 10 SideBar de Plantas e Invernaderos.....	45
Figura 11 Listado de Plantas Registradas.....	46
Figura 12 Criterios de Búsqueda registro de plantas.....	46
Figura 13 Formulario de registro de plantas.....	47
Figura 14 Registro de invernaderos	47
Figura 15 Criterios de Búsqueda registro de invernaderos.....	48
Figura 16 Formulario de registro de invernaderos.....	48
Figura 17 Exportación de archivos, plantas e invernaderos	49
Figura 18 SideBar para Materiales y Químicos	52
Figura 19 Registros de Materiales	53
Figura 20 Criterio de Búsqueda registro de materiales	53
Figura 21 Formulario de registro de materiales.....	54
Figura 22 Registros de químicos.....	54
Figura 23 Criterios de Búsqueda registro de químicos	55
Figura 24 Formulario de registro de químicos	55
Figura 25 Exportación de archivos, materiales y químicos.....	56
Figura 26 SideBar para Tipo de tratamientos y Mantenimientos	60
Figura 27 Registros de mantenimientos.....	60

Figura 28 Materiales usados.....	61
Figura 29 Formulario de registro de mantenimientos	61
Figura 30 Popup para agregar materiales.....	62
Figura 31 Listado de materiales a registrarse	62
Figura 32 Registros de tratamientos	63
Figura 33 Criterios de Búsqueda, registro de tratamientos.....	63
Figura 34 Formulario de registro de tratamientos.....	64
Figura 35 PopUp para agregar químicos	64
Figura 36 Listado de químicos a usarse.....	65
Figura 37 Exportación de archivos, tratamientos y mantenimientos.....	65
Figura 38 SideBar para Siembras y Tratamientos aplicados.....	69
Figura 39 Registros de siembras	69
Figura 40 Criterios de Búsqueda, registro de siembras.....	70
Figura 41 Formulario de registro de siembras.....	70
Figura 42 Registros de Tratamientos aplicados	71
Figura 43 Criterios de Búsqueda, registro de tratamientos aplicados.....	71
Figura 44 Formulario de registro, tratamientos.....	72
Figura 45 Formulario de registro, tratamientos.....	72
Figura 46 Formulario de registro, tratamientos.....	73
Figura 47 Exportación de archivos, siembras y tratamientos aplicados.....	73
Figura 48 SideBar para Cosechas y Siembras.....	77
Figura 49 Registros de cosechas.....	77
Figura 50 Criterios de Búsqueda, registro de cosechas	78
Figura 51 Formulario de registro de cosechas	78
Figura 52 Registros de ventas	79
Figura 53 Criterios de Búsqueda, registro de ventas.....	79
Figura 54 Formulario de registro de ventas.....	80
Figura 55 Exportación de archivos, cosechas y ventas.....	80

Figura 56 SideBar para el Dashboard	82
Figura 57 Dashboard, calendario y stock	82
Figura 58 Dashboard, ventas	83
Figura 59 Iniciar Sesión	85
Figura 60 Cerrar Sesión.....	85

ÍNDICE DE TABLAS

Tabla 1 Versiones de Angular	14
Tabla 2 Roles de SCRUM	26
Tabla 3 Historias de usuario.....	28
Tabla 4 Product backlog.....	35
Tabla 5 Planificación del proyecto.....	37
Tabla 6 Planificación del sprint 0.....	38
Tabla 7 Sprint 0.....	38
Tabla 8 Planificación del sprint 1	42
Tabla 9 Sprint backlog del sprint 1	42
Tabla 10 Planificación del Sprint 2	50
Tabla 11 Sprint backlog del sprint 2	50
Tabla 12 Planificación del sprint 3.....	57
Tabla 13 Sprint backlog del sprint 3	57
Tabla 14 Planificación del sprint 4.....	66
Tabla 15 Sprint backlog del sprint 4	66
Tabla 16 Planificación del sprint 5.....	74
Tabla 17 Sprint backlog del sprint 5	74
Tabla 18 Planificación del sprint 6.....	81
Tabla 19 Sprint backlog del sprint 6	81
Tabla 20 Planificación del sprint 7	84
Tabla 21 Sprint backlog del sprint 7	84
Tabla 22 Pruebas de aceptación.....	87
Tabla 23 Tabla de eficiencia.....	89
Tabla 24 Tabla de eficacia	90
Tabla 25 Análisis de impactos.....	90
Tabla 26 Impacto socio cultural.....	91
Tabla 27 Impacto económico	91

Tabla 28 Impacto ambiental 92

INTRODUCCIÓN

ANTECEDENTES

El framework Angular nació en el año 2011 con el nombre de AngularJS el cual esta embebido en JavaScript, con el pasar de los tiempo esta funcionalidad comenzó a quedar un poco obsoleta y así mismo incrementaban las inseguridades dentro de las aplicaciones empresariales, es por eso que fue mejorado por un grupo de ingenieros de Google con el fin de crear una herramienta para el desarrollo de aplicaciones que funcionen de manera universal, es decir, que funcionen correctamente en ambientes web, de escritorio y móviles.

Debido al extenso alcance que tiene Angular en el desarrollo de aplicaciones, es necesario mencionar una metodología de desarrollo, la cual será SCRUM. Las siglas de SCRUM, en sí, no tienen un significado o traducción alguna, puesto que es un término utilizado en el juego de rugby, el cual consiste en que todos los miembros del equipo trabajen de manera equitativa para obtener el mejor resultado posible, tomando esto como punto de partida en los inicios de la década de los 80's Ikujiro Nanoka e Hirotaka Takeuchi formularon nuevos procesos del trabajo en equipo dentro de las organizaciones(Sutherland & Schwaber, 2013), no fue hasta el año de 1995, que Ken Schwaber formalizo el término de SCRUM como una metodología de desarrollo ágil con su trabajo "Proceso de Desarrollo SCRUM " en OOPSLA o Object Oriented Programming Systems & Aplicaciónlication Conference(Palacio, Menzinsky, & López, 2016).

En las áreas rurales del cantón Cotacachi existen pequeñas microempresas cuya actividad principal es la producción y venta de productos orgánicos los cuales crecen dentro de invernaderos o a la intemperie, las personas que están a cargo de las plantas se rigen a procesos repetitivos y monótonos independientemente de la diversidad de la planta, sin tomar en cuenta que cada especie necesita cuidados específicos en las diferentes etapas de crecimiento y estaciones del año, es por eso que en muchos casos la producción total no es la esperada y en su lugar representan una perdida.

SITUACIÓN ACTUAL

Actualmente Angular es uno de los frameworks más completos ya que ofrece una amplia variedad de ambientes en los cuales trabajar, tales como aplicaciones web, escritorio y móviles, mediante sus TypeScript es posible una sintaxis muy a la par con otros lenguajes de programación orientados a objetos por ejemplo Java o C, dando como resultado una curva de aprendizaje relativamente pequeña, permitiendo al personal enfocarse en la lógica del negocio (Díaz, 2016). Así mismo es posible una perfecta integración con otras tecnologías como React, Polymer e incluso X-tag de Microsoft, también se pueden desarrollar aplicaciones híbridas, como en el caso de Ionic2 para el desarrollo de aplicaciones móviles (Oriol, 2016). Hoy en día esta herramienta es una de las más utilizadas, puesto que su resultado son aplicaciones modernas y escalables por la parte del cliente, esto se debe ya que trabaja de manera independiente la interfaz del usuario y la lógica del negocio, es muy diferentes a la mayoría de frameworks, ya que mezclan todos los componentes en un solo archivo.

En la actualidad las metodologías ágiles son las más utilizadas y populares en el mercado por su fácil aprendizaje, y versatilidad para gestionar cambios de requisitos en el proceso de desarrollo de software. Tomando en cuenta lo antes mencionado y conforme al estudio realizado en el 2011 por Javier Garzas, se puede apreciar que la metodología más utilizada es SCRUM (Garzas, 2012).

SCRUM es conocida como una metodología de desarrollo ágil que se apoya firmemente en el trabajo en equipo con el objetivo de realizar pequeños ciclos de desarrollo llamados sprints (API_MARKET, 2017), es por esta razón que muchas organizaciones han utilizado SCRUM, entre las cuales se nombrara a las más conocidas como: Apple, Amazon, Adobe, Google, Yahoo, Blizzard, etc (Aranda, 2017).

Actualmente las personas que deciden tener un sembradío lo hacen sin un adecuado cuidado o algún tipo de control, es por este motivo que las plantas no reciben un tratamiento adecuado lo que en consecuencia produce su muerte dando como resultado la pérdida de capital al agricultor, estos tienen registros en papel de las características y necesidades básicas de los cultivos, así como una contabilidad aproximada de pérdidas y ganancias obtenidas, lo cuales no permiten al agricultor sacar el máximo provecho del negocio. Adicionalmente en la mayoría de los casos no se cuenta con conocimientos específicos acerca de la manipulación adecuada de las plantas.

En la comunidad de La Calera, ubicada en Cotacachi, hay una microempresa que se dedica a la producción de tomates dentro de invernaderos, la cual lleva a cabo diferentes procesos tradicionales, mencionando algunos, como su sistema de fumigación, registro de productos químicos utilizados para tener a la planta en buen estado, inventario de los materiales e insumos con los que cuenta, cabe recalcar que todos estos procesos son registrados de manera física, en papel, y debido a esto en muchas ocasiones se pierden dichos registros, por lo cual al momentos de necesitarlos demoran mucho en encontrarlas. Cuentan con la ayuda de un ingeniero Agrónomo el cual los guía para tener un adecuado cuidado de las plantas, así mismo con las combinaciones de los distintos químicos, tanto para regarlas como para fumigarlas.

PROSPECTIVA

La implementación del sistema de control de productividad dentro de los invernaderos facilitará el cuidado de las plantas ya que por medio del mismo se contara con los debidos registros de riego y fumigación, y se llevará a cabo una anotación por cada variedad de planta sembrada, así como también un listado de los gastos realizados para la sustentar a la planta y uno de ganancias una vez se inicie la comercialización de los productos a cosechar, de esta manera se evitara la pérdida de los registros que se anotan en papel, evitando así una extensa aglomeración de documentación.

Mediante la utilización de la metodología de desarrollo ágil SCRUM se efectiviza el desarrollo de la aplicación en el tiempo estimado, y atreves de sus buenas prácticas de la metodología se mejorará la calidad del producto.

El uso Angular en la elaboración del sistema web contribuirá a un desarrollo debidamente estructurado, ya que sus microservicios garantizarán un sistema completamente autónomo y funcional.

PLANTEAMIENTO DEL PROBLEMA

La Finca Agrícola del Milenio, necesita llevar un control de la producción de los invernaderos que tiene. A continuación, se define el árbol de problemas:



Figura 1 Árbol de Problemas

Fuente: Propia

Por lo cual se define al problema: "¿Cómo mejorar el manejo de los procesos para aumentar la producción de un invernadero?".

OBJETIVOS

1. Objetivo General

Desarrollar un Sistema Web para el Control de Producción de un Invernadero de tomate en el Cantón Cotacachi, utilizando el Framework Angular versión 4.

2. Objetivos Específicos

- Analizar los procesos de producción de tomate que se automatizarán en el sistema web.
- Aplicar la metodología SCRUM en el desarrollo del sistema.
- Implantar el sistema web de control de producción en el invernadero Cotacachi.
- Validar los beneficios de la implementación del sistema.

ALCANCE

El proyecto tiene como fin desarrollar un sistema web que permita tener un control de la producción de tomate dentro de un invernadero. El sistema contará con los siguientes módulos:

1. Módulo de gestión de usuario

Permite controlar que únicamente las personas registradas ingresen al sistema, con el fin de resguardar la información de sus sembradíos.

2. Módulo de inventario de materiales e insumos

Permite llevar un listado de los materiales e insumos con los que se cuente en ese momento.

3. Módulo de control de producción

a) Proceso de siembra

Permite registrar las plantas y el tipo de variedad que van a ser plantadas dentro del invernadero.

b) Proceso de fumigación

Permite registrar las distintas fechas de fumigación y las diferentes mezclas de químicos que se realizaron para dicha fumigación, acorde a las específicas de cada planta.

c) Proceso de registro de enfermedades de las plantas.

Se anotarán los diferentes problemas que se presenten en el crecimiento de las plantas.

d) Proceso de cosechas

Almacenará los datos referentes a la cantidad del producto a cosechar, y el cómo va aumentando o decreciendo dicha cantidad.

4. Módulo de calendarios

Permitirá llevar un registro de los días más óptimos para sembrar las diferentes plantas.

5. Módulo de control de químicos

Permite guardar la información pertinente a los diferentes químicos utilizados para mantener a las plantas saludables, así como también, un registro de los días en lo que es necesario alimentar a la plantas con dichas sustancias.

6. Módulo de consultas y reportes

El usuario podrá obtener reportes tabulares de los químicos usados, así como de las plantas cosechadas en el periodo actual.

JUSTIFICACIÓN

Un sistema de control de producción es necesario para cualquier grupo de personas u organizaciones que desean ejecutar procesos y almacenar registros de forma estructurada y que estos mismos sean accesibles de una manera rápida y espontánea.

En el cantón Cotacachi hay agricultores y microempresas que laboran con dificultad sus diferentes actividades administrativas tales como el registro adecuado de los ingresos y egresos que requiere una actividad agrícola, así como también las diferentes cantidades y combinaciones de químicos requeridas para un adecuado desarrollo y crecimiento de las plantas con la finalidad de obtener productos de calidad al momento de cosecharlas, es notable que la existencia de una brecha en este aspecto, la cual se pretende subsanar con la introducción de tecnología, recalcando que difícilmente los procesos manuales superen las facilidades que ofrece las herramientas tecnológicas, ya que mediante esta es posible ahorrar tiempo, y teniendo esta información a la mano se prevé mejorar la toma de decisiones.

El desarrollo de una aplicación web enfocada en resolver los problemas de papeleo y contabilidad, permitirá un procesamiento eficiente y estructurado con la información más importante para los agricultores, con esto facilitando el registro de los mismos, de esta manera se tendría los datos organizados de modo que estos registros ayuden en la toma de decisiones.

Mediante la utilización del sistema web se desea disminuir el consumo de papel, ya que se evitará tener registros físicos, y de esta manera hacer un aporte al medio ambiente.

Al utilizar Angular en el desarrollo del sistema se logrará una aplicación de calidad en menor tiempo, debido que es una herramienta que permite incorporar diferentes servicios, dependiendo de las necesidades del usuario, organizándolos de una manera estructurada y ordenada, además la parte de la seguridad está cubierta por complementos de mismo framework.

CAPÍTULO 1

1. MARCO TEÓRICO

1.1 ACTIVIDAD AGRÍCOLA

El Ecuador debido a su situación geográfica, es uno de los países que cuenta con una amplia diversidad de suelos, los cuales se encuentran dispersos a lo largo de las regiones Costa, Sierra y Oriente.

Cabe recalcar que cada región del Ecuador es apta para una específica actividad agrícola, como es el caso de la Costa, que debido a su cercanía con el mar, los sembradíos que predominan son aquellos que tengan un ciclo corto, a diferencia de la Sierra, es aquí en donde se encuentra una inmensa variedad cultivos sembrados dentro de plantaciones o a la intemperie, y finalmente tenemos a la Amazonía que es considerada como un área tropical, ya que debido a su clima de constantes lluvias y elevadas temperaturas, nos da una enorme diversidad de flora y fauna, por lo cual deforestar este ambiente traería complicaciones graves, tales como la erosión de la tierra, trayendo como consecuencia la infertilidad de las tierras.

Es importante mencionar que las cifras del Instituto Nacional de Estadística y Censos (INEC) consideran a la actividad agrícola como es la principal fuente de empleo en Ecuador y eje de seguridad alimentaria, permitiendo aliviar el impacto económico causado por los bajos precios del petróleo, generando divisas al país y suministrando materias primas a la industria nacional (Hora, 2013).

La última Encuesta Nacional de Empleo, Desempleo y Subempleo (Enemdu), elaborada por el INEC, indica que la rama de la agricultura, ganadería, caza y silvicultura, y pesca concentra la mayor participación del empleo en el país, con el 29,3%, en el primer trimestre del año. A esta le siguen las actividades de comercio y manufactura (incluida refinación de petróleo) con el 17,9% y 10,3%, respectivamente, señala el INEC (Telégrafo, 2017).

1.2 MICROEMPRESAS AGRÍCOLAS EN COTACACHI

En la provincia de Imbabura, en las zonas rurales, cada vez hay más personas u organizaciones que incursionan con la agricultura, debido a que la tierra de la región es rica en nutrientes y debido a su ubicación geográfica tiene facilidad de acceder a corrientes o pozos de agua.

Un ejemplo de estos emprendimientos, es la Asociación de Desarrollo Integral “El Viejo San Martín”, está conformada por gran número de integrantes, debido a esto dentro de la asociación se encuentran diferentes cultivos, tales como pimiento, tomate de árbol, maíz, y efectivamente cultivos de tomate de riñón dentro de invernaderos, a los cuales los nutren y protegen con estrictos procesos de fertilización, fumigación y riego.

La Finca Agrícola del Milenio (FAM), es una pequeña organización que día a día se abre paso dentro del mercado, con sus productos de alta calidad, se ha hecho acreedor a una clientela que, al igual que la Asociación “El Viejo San Martín”, FAM tiene estrictos procesos para cuidar a las plantas.

1.3 SISTEMAS DE CONTROL DE PRODUCCIÓN

1.3.1 ¿Qué es un sistema de control de producción?

Un sistema de control de producción es una herramienta, que nos permite controlar la actividad productiva. Basándose en el concepto de Just In Time (JIT), justo a tiempo, que permite definir la manera en la que se debe optimizar un sistema de producción, debido a que se enfoca en la demanda y no en manejar cantidades enormes de entregas o materia prima (Barcelona, 2002).

El justo a tiempo trabaja sobre cuatro pilares fundamentales:

- Atacar a los problemas que son originados debido al mal control de los procesos.
- Deshacerse de los costos que no generen valor a los productos.
- Realizar los procesos con simplicidad.
- Diseñar sistemas para reconocer los diferentes problemas, y así implantar sistemas que estén orientados al control total de la calidad.

1.3.2 Actividades guía del Justo a Tiempo

- **Servicio al cliente**

En términos generales el servicio al cliente es aquel que detalla la disponibilidad de artículos al cliente, cuando este los necesite. Teniendo en cuenta que un cliente puede ser el consumidor de un producto determinado, un distribuidor, una planta dentro de una organización, etc (Barcelona, 2002).

- **Procesamiento de ordenes**

Es la acción que pone en marcha el proceso de distribución y gestiona las acciones a ser tomadas para satisfacer las peticiones de los usuarios, mediante la comunicación de actividades que enlaza el proceso de la implantación de calidad y la disminución de costos, una perfecta comunicación es una sólida base para un exitoso JIT (Editores Noriega, 2015)(Barcelona, 2002).

- **Control de inventarios**

El control de inventarios es uno de los puntos más cruciales dentro de JIT, ya que es crucial tener un balance que permita satisfacer desde la demanda más pequeña, hasta la más alta. El triunfo de este dependerá de la correcta definición del nivel necesario de inventario para alcanzar el nivel deseado de servicio al cliente (Editores Noriega, 2015).

- **Planeación de demanda**

En este proceso se involucran las cantidades existentes del producto y los respectivos servicios que requieran los usuarios a futuro. La predicción de esta demanda se puede definir mediante estudios de marketing, ya que el conocimiento de esta proporciona a los administradores la información necesaria para asignar los distintos recursos a las distintas peticiones (Editores Noriega, 2015).

- **Tráfico y transporte**

El movilizar y manipular los materiales de un lugar a otro representa una de las más importantes actividades de JIT, ya que con una debida aplicación, da mayor eficacia a todos los procesos. Otro punto que se ve integra en este ámbito es el de almacenamiento ya es sumamente necesario preservar los productos que conforman al inventario (Editores Noriega, 2015).

- **Localización de planta y almacenes**

Al ubicar las instalaciones de la empresa u organización en lugares estratégicos, es un factor sumamente importante, ya que con una adecuada ubicación permite reducir gastos como el transporte, este siendo necesario para poder movilizar los productos (Editores Noriega, 2015).

- **Manejo de mercancías**

La correcta manipulación de productos trae consigo un adecuado movimiento tanto de la materia prima como el de inventario, el objetivo principal del manejo de mercancías es: minimizar la distancia, aminorar la manipulación de la materia prima y reducir al máximo la pérdida por desperdicios (Editores Noriega, 2015).

- **Abastecimiento**

Para un adecuado abastecimiento se deben tener proveedores, con los cuales se tenga perfectamente claro, el cómo ha sido adquirido el material, el proceso de compras, el control de calidad, etc. Debido a todos estos procesos es que adquirir productos de excelentes calidad es de vital importancia para el proceso productivo de la empresa (Editores Noriega, 2015).

- **Empaque**

En este proceso se desarrollan dos funciones fundamentales, la función de calidad y la función de mercados, en la parte de mercados este actúa como un medio de impulso, puesto que a sus diferentes tamaños, peso, color e información impresa atrae a posibles clientes y por el sentido de calidad empaquetado protege al producto de sufrir daños y facilita su movilidad y manipulación (Editores Noriega, 2015).

- **Disposición del desperdicio y material reciclable**

Este proceso debe considerarse como una responsabilidad tras implantar un sistema de calidad, ya que debido a la naturaleza de los estos sistemas, siempre están presentes materiales de desperdicio y estos deben ser almacenados o desechados en un lugar en donde no represente ningún peligro (Editores Noriega, 2015).

Para una correcta aplicación de JIT, es necesario tener una disciplina y previo a esto un cambio de mentalidad, que se puede obtener mediante una implantación de una manera de pensar dirigida hacia la calidad, enfocada en el mejoramiento continuo, y así mismo tener una flexibilidad a los distintos cambios que se presenten (Editores Noriega, 2015).

1.3.3 Control de la producción informático

Un sistema de control de producción es de gran apoyo al encargado de la producción, ya que representa una herramienta muy versátil con la cual se pueden registrar diversos procesos, y ya con los respectivos datos, se pueden generar reportes en el momento en el que el usuario los necesite (Editores Noriega, 2015)

La implantación de sistema de control de producción mediante la tecnología, traza un cambio significativo en la manera de realizar todos los procesos. Dando lugar a que intervenga un ordenador, en la manera de recolección de datos y como obtener información crítica a modo de reportes.

Dentro de FAM se consta con las herramientas básicas para un sistema de control de producción, el cual es un ordenador, debido a esto, se desarrollará un sistema web, con el fin de tener un acceso rápido a los registros y reportes que sean necesarios para llevar un adecuado procedimiento de control.

1.3.4 ¿Qué son sistemas web?

Los sistemas web reciben este nombre porque se ejecutan en el internet. Es decir que los datos o los archivos en los que trabajas son procesados y almacenados dentro de la web. Estas aplicaciones, por lo general, no necesitan ser instaladas en un computador.

El concepto de sistemas web está relacionado con el almacenamiento en la nube. Toda la información se guarda de forma permanente en grandes servidores de internet y nos envían a nuestros dispositivos o equipos los datos que requerimos en un momento específico.

Estos grandes servidores de internet que prestan el servicio de alojamiento están ubicados alrededor de todo el mundo, así hacen que el servicio prestado no sea tan costoso o gratuito en la mayoría de los casos y es extremadamente seguro.

Las aplicaciones web son un tipo de software que se codifica en un lenguaje soportado por los navegadores web y cuya ejecución es llevada a cabo por el navegador en Internet o de una intranet, ahí que reciban el nombre de Aplicación web (Barzanallana, 2019).

1.3.5 Ventajas de las aplicaciones web

- Ahorro de tiempo: Son aplicaciones sencillas de gestionar, por lo que permiten realizar tareas de forma fácil sin necesidad de descargar ni instalar ningún programa o componente adicional.
- Completa compatibilidad: Son totalmente compatibles con navegadores para poder utilizarlas. Sólo se suele requerir que el navegador web esté debidamente actualizado.
- Actualización continua e inmediata: Debido a que es el propio desarrollador el que gestiona y controla el software, la versión que se descarguen, instalen y utilicen los usuarios, siempre será la última que haya lanzado dicho desarrollador aplicación.

- Recuperación de datos: Una de las mayores ventajas de una Aplicación Web es que, en la mayoría de ocasiones el usuario puede recuperar su información desde cualquier dispositivo y lugar con su nombre de usuario y contraseña.
- Ahorro de recursos en equipos y dispositivos: Las aplicaciones web, generalmente tiene un bajo consumo de recursos dado que toda o gran parte de la aplicación se encuentra en un servidor web y no en el ordenador.
- Gran parte de las tareas que realiza el software no consumen recursos propios del equipo o dispositivo, ya que se realizan desde el mismo servidor o servidores en el que se encuentre alojada la aplicación.
- Compatibilidad con múltiples plataformas: se pueden usar desde cualquier dispositivo móvil o equipo informático, ya que son válidas para cualquier sistema operativo ya que sólo es necesario que cuente con un navegador web y acceso a internet.
- Portabilidad: Las aplicaciones web son compatibles con toda la plataforma, ya que con tener acceso a internet se puede acceder a las mismas.
- Virus inexistentes: Los virus no afectan a los datos guardados en el servidor de la aplicación, ya que cuentan con potentes sistemas de seguridad informáticos y garantías ante este tipo de situaciones (Alegsa.com.ar, 2019).

1.3.6 Desventajas de las aplicaciones web

- Las aplicaciones web requieren navegadores web totalmente compatibles para funcionar. Incluso muchas veces requieren las extensiones apropiadas y actualizadas para operar.
- Muchas veces requieren una conexión a internet para funcionar, si la misma se interrumpe, no es posible utilizarla. De todas maneras, en ocasiones, pueden ser descargadas e instaladas localmente para su uso offline.
- Muchas no son de código abierto, perdiendo flexibilidad.
- La aplicación web desaparece si así lo requiere el desarrollador o si el mismo se extingue. Las aplicaciones tradicionales, en general, pueden seguir usándose en esos casos (Alegsa.com.ar, 2019).

Los beneficios son muchos, en comparación a las desventajas que claramente son fáciles de solucionar, y con una debida capacitación los beneficios no tardarán en mostrarse hacia los usuarios.

1.4 ENTORNO DE DESARROLLO

1.4.1 Base de datos MySQL

MySQL es una de las bases de datos de código abierto más popular del mundo, lo que permite una entrega rentable de aplicaciones de bases de datos integradas y confiables, de alto rendimiento y escalables.

Es una base de datos integrada y segura para transacciones, compatible con ACID con capacidades completas de compromiso, retrotracción, recuperación de fallas y bloqueo a nivel de fila.

Ofrece la facilidad de uso, la escalabilidad y el alto rendimiento, así como un conjunto completo de controladores de base de datos y herramientas visuales para ayudar a los desarrolladores y administradores de bases de datos a crear y administrar sus aplicaciones de MySQL críticas para un negocio específico. MySQL es desarrollado, distribuido y respaldado por Oracle (Oracle, 2019).

MySQL proporciona las siguientes características:

- Alto rendimiento y escalabilidad para satisfacer las demandas de usuarios y cargas de datos en crecimiento exponencial.
- Clústeres de replicación para mejorar la escalabilidad, el rendimiento y la disponibilidad.
- Cambio de esquema en línea para cumplir con los cambiantes requisitos comerciales.
- Esquema de rendimiento para supervisar la rentabilidad a nivel de usuario, aplicación.
- Consta de acceso a SQL y NoSQL para realizar operaciones simples y complejas mediante una estructura de clave y valor.
- Brinda la flexibilidad para desarrollar e implementar en múltiples sistemas operativos (Oracle, 2019).

1.4.2 Framework Angular

Con el transcurso del tiempo, angular, ha desarrollado varias versiones.

Tabla 1 Versiones de Angular

Versión	Fecha de Lanzamiento
4	Marzo 2017
5	Septiembre/Octubre 2017
6	Marzo 2018
7	Septiembre/Octubre 2018

Fuente: (CampusMVP, 2019).

Para el desarrollo del sistema de FAM se hará uso de la versión 4 de angular, debido a que tiene una gran estabilidad y compatibilidad con muchos componentes.

Angular 4, normalmente llamado Angular, es uno de los framework más completos para elaborar aplicaciones en cliente con HTML y JavaScript, es decir, que su objetivo principal está en que el peso de la lógica y el renderizado lo realice el propio navegador, en lugar del servidor (Oriol, 2019a).

Arquitectura de una aplicación en Angular 4

Dentro de angular podemos identificar ocho módulos principales que conformar a una aplicación.

- **Módulo**

Las aplicaciones desarrolladas con angular son modulares, es decir está compuesto por otros módulos más, actualmente no es necesario escribir una sintaxis para definir a los módulos, sino que hace uso del estándar ECMAScript 2015.

ECMAScript 2015 es una orientación a clases y herencias, que incluyen, el uso de módulos, parámetros por defecto, integra a las variables let y const. Teniendo como un punto a considerar es que los principales navegadores ya tienen implementadas estas funciones, lo cual asegura la ejecución del código de nuestras aplicaciones en todos los navegadores.

Generalmente un módulo es un conjunto de código dedicado a únicamente un objetivo, exportando algo representativo como una clase

También podemos encontrar módulos que son librerías de un conjunto de módulos, un claro ejemplo lo podemos ver en el mismo Angular, ya que empaqueta una colección de librerías vinculadas a diferentes paquetes, de tal modo que importamos aquellos servicios o

módulos que necesitemos, haciendo así a la aplicación más ligera al momento de navegar dentro de la aplicación (Oriol, 2019a).

Tenemos como principales librerías:

- @angular/core
- @angular/common
- @angular/router
- @angular/http

- **Componente**

Un componente controla la zona de espacio de una pantalla, que la denominaremos vista. El componente sería el equivalente a un controlador que siempre va ligado a una vista.

Define propiedades y métodos que se encuentran disponibles en su template. Es crucial seguir el modelo de diseño S.O.L.I.D (Oriol, 2019a).

S.O.L.I.D. nos presta cinco principios fundamentales sobre el diseño orientado a objetos, basándose en la gestión de dependencias (Uniwebsidad, 2019).

- S, Single responsibility principle o Principio de responsabilidad única
- O, Open/close principle o Principio de abierto/cerrado
- L, Liskov substitution principle o Principio de sustitución de Liskov
- I, Interface segregation principle o Principio de segregación de la interfaz
- D, Dependency inversion principle o Principio de la inversión de la dependencia

- **Template**

Nos permite definir la vista de un componente, este template tiene un formato de HTML, pero se encuentra embebido con componentes y expresiones de angular que enriquecen el comportamiento del template (Oriol, 2019a).

- **Metadatos**

En una clase de angular podemos encontrar únicamente clases JavaScript, no obstante, se puede indicar que se trata de un componente, con la intervención de los metadatos de angular (Oriol, 2019a), y son:

- Selector: Es un selector de CSS que indica a Angular que debe crear e instanciar el componente cuando se encuentra un elemento con ese nombre en el HTML

- **ModuleId:** Este parámetro, que viene de la sintaxis CommonJS, es opcional y utilizando la asignación exacta `moduleId: module.id`, permite que Angular busque los templates y css a partir de urls relativas en lugar de absolutas. Inicialmente estas url tenían que ser absolutas, pero gracias al `moduleId` ya no es necesario, logrando así una implementación más modular.
- **TemplateUrl:** La url en la que se encuentra el template que se quiere vincular al componente.
- **StyleUrl:** Un array con urls a archivos de estilos que se quieren aplicar al componente.
- **Directives:** Este parámetro es para facilitar la Inyección de Dependencias.
- **Providers:** Igual que con las directivas, aquí se definen los servicios de los que depende el componente.

```
import { Component, Input } from '@angular/core';
import { Todo } from '../shared/todo.model';

@Component({
  moduleId: module.id,
  selector: 'todo-detail',
  template: ` <div *ngIf="todo">
    <h2>Selected todo:</h2>
    <div class="subject">{{todo.subject}}</div>
    <div>{{todo.content}}</div>
  </div>`,
  styles: [
    .subject{
      font-weight: 600;
      font-style: italic;
      margin-bottom: 0.2em;
    }
  ]
})
export class TodoDetailComponent {
  @Input()
  todo: Todo;

  constructor() { }
}
```

Figura 2 Metadatos en Angular

Fuente: (Oriol, 2019b).

- **Data Binding**

Uno de los principales valores de Angular es que abstrae de la lógica pull/push asociada a insertar y actualizar valores en el HTML con lo cual convierte las respuestas de los usuarios (inputs, clicks, etc) en acciones concretas. Escribir toda esa lógica a mano es tedioso y propenso a errores, angular lo resuelve gracias al Data Binding (Oriol, 2019b).

- **Directiva**

Una directiva se define como una clase que utiliza el decorador @Directive. Un Componente es un caso concreto de directiva que siempre va asociado a un template y al que por ser un elemento tan importante en angular se le ha dado un decorador propio (Oriol, 2019b).

- **Servicios**

Se definen mediante simples clases. Todo valor, función o característica que la aplicación necesita, desde constantes a la lógica de negocio, se encapsula dentro de un servicio. Los Componentes son grandes consumidores de servicios. No recuperan datos del servidor, ni validan inputs de usuario. Delegan todo este tipo de tareas a los Servicios (Oriol, 2019c).

- **Dependency Injection**

Es un mecanismo para proporcionar nuevas instancias de una clase con todas aquellas dependencias que requiere plenamente formadas. La mayoría de dependencias son servicios, y Angular usa la DI para proporcionar nuevos componentes con los servicios que necesitan. Entre otras cosas, la DI facilita la modularidad y testabilidad del código (Oriol, 2019d).

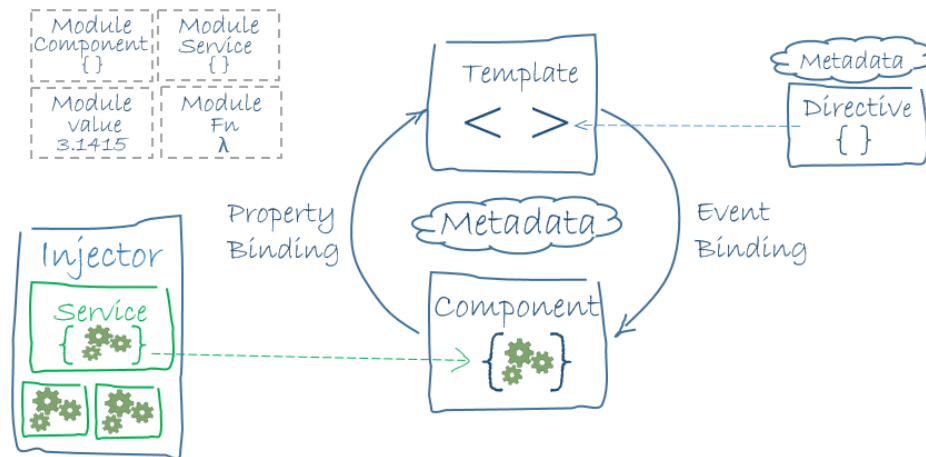


Figura 3 Dependencias en Angular

Fuente: (Oriol, 2016).

1.4.3 Framework SpringBoot

Spring Boot es el proyecto más reciente de spring que nos ayuda a iniciar un proyecto utilizando los diferentes proyectos de spring de una manera ágil y evitar la configuración, pcon la cual se ha tenido varios inconvenientes por mucho tiempo. Toda esta magia se debe a la configuración por defecto que trae dentro y que puede ser configurada vía propiedades. Spring Boot nos permite poner enfoque en agregar valor y mejorar la experiencia del desarrollador (MELÉNDEZ, 2019).

- **Características**

- Convención sobre configuración: En lugar de escribir la configuración necesaria y validar si es correcta, Spring Boot provee las configuraciones necesarias bajo diferentes escenarios. De esta manera evita la tarea repetitiva de estar agregando las configuraciones una y otra vez.
- Gestion de dependencias: Spring Boot provee un análisis de las dependencias que los proyectos alrededor de spring utilizan de manera que los desarrolladores solo denben que indicar que dependencia necesita, sin necesidad de indicar la versión. Al mismo tiempo, ofrece una manera práctica manipula el versionamiento de las dependencias en caso se requiera hacer una actualización o desactualización de la misma.

- Auto-configuración: Spring y sus diferentes proyectos necesitan ser configurados para que funcionen de manera adecuada y se pueda continuar agregando valor a las aplicaciones. Spring Boot es lo suficientemente autónomo como para poder activar las configuraciones necesarias si cumple con ciertas condiciones como si las clases están en el classpath o los beans han sido creados o los properties correspondientes han sido habilitados.
- Dependencias iniciales: SpringBoot provee dependencias que traen consigo las dependencias necesarias para empezar un proyecto.

- **¿Cómo crear un proyecto SpringBoot?**

Spring nos brinda un servicio el cual nos permite generar un proyecto, mediante una interfaz en la que podemos escoger la versión de Java a utilizar, la versión de spring, el servidor sobre el cual trabajará, el lenguaje a utilizar, los paquetes a generar. De esta manera se obtiene una base sólida para comenzar a desarrollar (MELÉNDEZ, 2019).

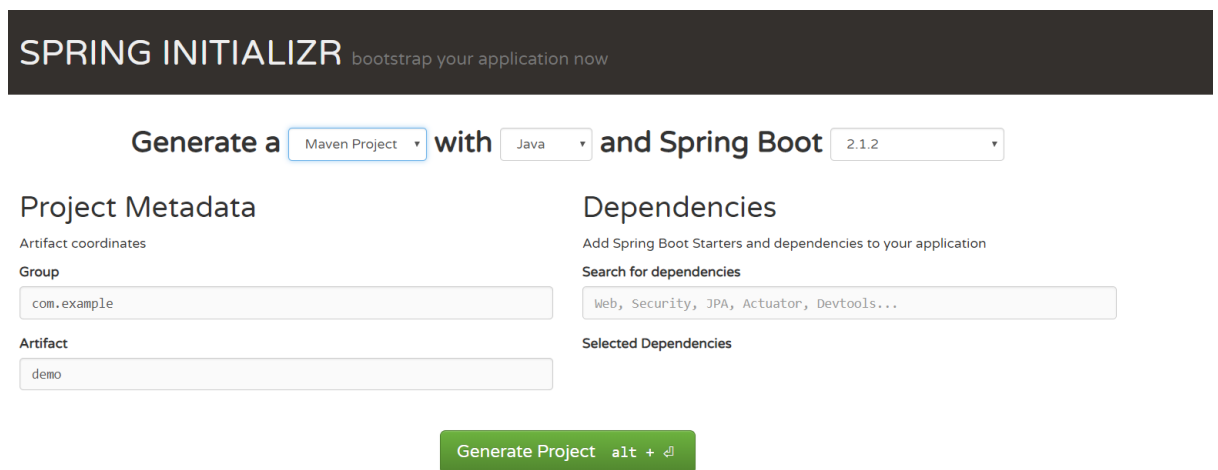


Figura 4 Creación de proyectos de SpringBoot

Fuente: (Spring, 2019).

1.4.4 Eclipse

Eclipse es una plataforma de software compuesto por un conjunto de herramientas de programación de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores.

Esta plataforma, típicamente ha sido usada para desarrollar entornos de desarrollo integrados (IDE), como el ambiente de Java llamado Java Development Toolkit (JDT) y el compilador que se entrega como parte de Eclipse, y son usados también para desarrollar el mismo.

- **Arquitectura de Eclipse**

La base para Eclipse es la Plataforma de cliente enriquecido, del inglés Rich Client Platform RCP. Los siguientes componentes constituyen la plataforma de cliente enriquecido.

- Plataforma principal - inicio de Eclipse, ejecución de plugins.
- OSGi - una plataforma para bundling estándar.
- El Standard Widget Toolkit (SWT) - Un widget toolkit portable.
- JFace - manejo de archivos, manejo de texto, editores de texto.
- El Workbench de Eclipse - vistas, editores, perspectivas, asistentes.

Eclipse provee al programador con frameworks muy ricos para el desarrollo de aplicaciones gráficas, definición y manipulación de modelos de software, aplicaciones web, etc. Por ejemplo, Graphic Editing Framework o Framework para la edición gráfica (GEF) es un plugin de Eclipse para el desarrollo de editores visuales que pueden ir desde procesadores de texto wysiwyg hasta editores de diagramas UML, Interfaces Gráficas para el Usuario (GUI).

Recursos es un término colectivo para los proyectos, carpetas y archivos que existen en Workbench. Las vistas de navegación proporcionan una vista jerárquica de recursos y le permite abrirlos para editarlos. Otras herramientas pueden mostrar y manejar estos recursos de manera diferente.

Para comenzar a desarrollar en eclipse debemos considerar el espacio de trabajo o también conocido como Workbench, el cual se compone de tres recursos básicos, y son:

- Archivos: se comparan a los archivos de un sistema de IDE.
- Carpetas: es similar a los directorios de un sistema de archivos, dentro del IDE, las carpetas contienen proyectos u otras carpetas, y estas, pueden archivos o más carpetas.
- Proyectos: contiene a las carpetas y archivos antes mencionados, estos, son utilizados para compilaciones, administración de versiones, intercambio y organización de recursos. Al igual que las carpetas, los proyectos se asignan a directorios en el sistema de archivos.

Un proyecto puede estar abierto o cerrado. Cuando se cierra un proyecto, no se puede manipular en Workbench. Los recursos de un proyecto cerrado no aparecerán en Workbench, pero los recursos aún residen en el sistema de archivos local, los proyectos cerrados requieren menos memoria. Dado que no se examinan durante las compilaciones, cerrar un proyecto puede mejorar el tiempo de compilación.

Cuando un proyecto está abierto, la estructura del proyecto se puede cambiar y verá los contenidos.

Las carpetas y los archivos se pueden vincular a ubicaciones en el sistema de archivos fuera de la ubicación del proyecto. Estas carpetas y archivos especiales se denominan recursos vinculados (Eclipse, 2019).

1.5 METODOLOGÍA DE DESARROLLO DE SOFTWARE

1.5.1 Metodología de desarrollo ágil

Las metodologías de desarrollo son aquellas que nos dejan integrar la forma de trabajo a las necesidades del proyecto, con el fin de tener flexibilidad e inmediatez, con el fin de adecuar el desarrollo del proyecto a circunstancias específicas del proyecto (lebschool, 2019).

Ventajas de utilizar las metodologías ágiles

- Mejora la satisfacción del cliente, debido a que se lo compromete desde el inicio del proyecto.
- Enriquece la motivación e implicación del equipo de desarrollo al proyecto, ya que todos los miembros del grupo están al tanto del estado del proyecto.
- Permite ahorrar tiempo y costes, ya que trabaja de manera más rápida y eficiente, haciendo cumplir de manera estricta el presupuesto y los periodos pactados.
- Debido a entregas continuas y a la participación del cliente es posible depurar el proyecto eliminando características innecesarias del producto (lebschool, 2019).

1.5.2 Metodología SCRUM

Scrum es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos.

En Scrum se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales (Softeng, 2019).

1.5.3 ¿Cómo funciona SCRUM?

En Scrum un proyecto se ejecuta en ciclos temporales cortos y de duración fija. Cada iteración tiene que proporcionar un resultado completo, un incremento de producto final que sea susceptible de ser entregado con el mínimo esfuerzo al cliente cuando lo solicite.

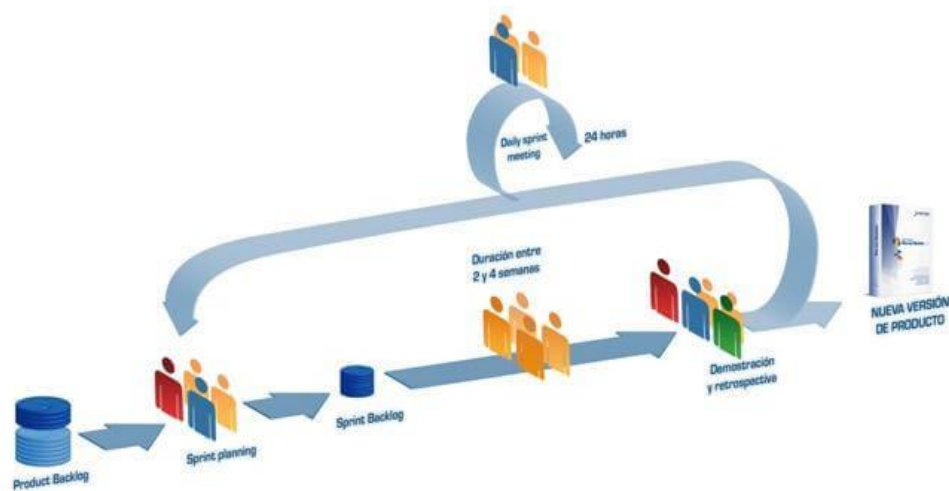


Figura 5 Proceso de la Metodología de SCRUM

Fuente: (Softeng, 2019)

El proceso parte de la lista de objetivos priorizada del producto, que actúa como plan del proyecto. En esta lista el cliente prioriza los objetivos balanceando el valor que le aportan respecto a su coste (que el equipo estima considerando la Definición de Hecho) y quedan repartidos en iteraciones y entregas (Proyectos ágiles, 2019).

- Planificación de la iteración: El primer día de la iteración se realiza la reunión de planificación de la iteración. Tiene dos partes:
 - Selección de requisitos. El cliente presenta al equipo la lista de requisitos priorizada del producto o proyecto. El equipo pregunta al cliente las dudas que surgen y selecciona los requisitos más prioritarios que se compromete a completar en la iteración, de manera que puedan ser entregados si el cliente lo solicita.

- Planificación de la iteración. El equipo elabora la lista de tareas de la iteración necesarias para desarrollar los requisitos a que se ha comprometido. La estimación de esfuerzo se hace de manera conjunta y los miembros del equipo se autoasignan las tareas.
- Ejecución de la iteración: Cada día el equipo realiza una reunión de sincronización, normalmente delante de un tablero físico o pizarra. En la reunión cada miembro del equipo responde a tres preguntas:
 - ¿Qué he hecho desde la última reunión de sincronización?
 - ¿Qué voy a hacer a partir de este momento?
 - ¿Qué impedimentos tengo o voy a tener?
- Inspección y adaptación: El último día de la iteración se realiza la reunión de revisión de la iteración. Tiene dos partes:
 - Demostración. El equipo presenta al cliente los requisitos completados en la iteración, en forma de incremento de producto preparado para ser entregado con el mínimo esfuerzo. En función de los resultados mostrados y de los cambios que haya habido en el contexto del proyecto, el cliente realiza las adaptaciones necesarias de manera objetiva, ya desde la primera iteración, replanificando el proyecto.
 - Retrospectiva. El equipo analiza cómo ha sido su manera de trabajar y cuáles son los problemas que podrían impedirle progresar adecuadamente, mejorando de manera continua su productividad. El Facilitador se encargará de ir eliminando los obstáculos identificados.

1.5.4 Reuniones de SCRUM

- Planificación del Backlog

Se definirá un documento en el que se reflejarán los requisitos del sistema por prioridades. En esta fase se definirá también la planificación del Sprint 0, en la que se decidirá cuáles van a ser los objetivos y el trabajo que hay que realizar para esa iteración. Se obtendrá además en esta reunión un Sprint Backlog, que es la lista de tareas y que es el objetivo más importante del Sprint (Manuel Trigas Gallego, 2019).

- Seguimiento del Sprint

En esta fase se hacen reuniones diarias en las que las 3 preguntas principales para evaluar el avance de las tareas serán: ¿Qué trabajo se realizó desde la reunión anterior? ¿Qué trabajo se hará hasta una nueva reunión? Inconvenientes que han surgido y qué hay que solucionar para poder continuar (Manuel Trigas Gallego, 2019).

- Revisión del Sprint

Cuando se finaliza el Sprint se realizará una revisión del incremento que se ha generado. Se presentarán los resultados finales y una demo o versión, esto ayudará a mejorar el feedback con el cliente (Manuel Trigas Gallego, 2019).

1.5.5 Roles de SCRUM

En Scrum, el equipo se enfoca en construir software de calidad. La gestión de un proyecto Scrum se centra en definir cuáles son las características que debe tener el producto a construir, qué construir, qué no y en qué orden, y en vencer cualquier obstáculo que pudiera entorpecer la tarea del equipo de desarrollo (Manuel Trigas Gallego, 2019).

El equipo Scrum está formado por los siguientes roles:

- Scrum master: Persona que lidera al equipo guiándolo para que cumpla las reglas y procesos de la metodología. Gestiona la reducción de impedimentos del proyecto y trabaja con el Product Owner para maximizar el Retorno de Inversión – Return of Investmen (ROI).
- Product owner: Representante de los accionistas y clientes que usan el software. Se focaliza en la parte de negocio y es el responsable del ROI del proyecto, entregar un valor superior al dinero invertido. Traslada la visión del proyecto al equipo, formaliza las prestaciones en historias a incorporar en el Product Backlog y las reprioriza de forma regular.
- Team: Grupo de profesionales con los conocimientos técnicos necesarios y que desarrollan el proyecto de manera conjunta llevando a cabo las historias a las que se comprometen al inicio de cada sprint.

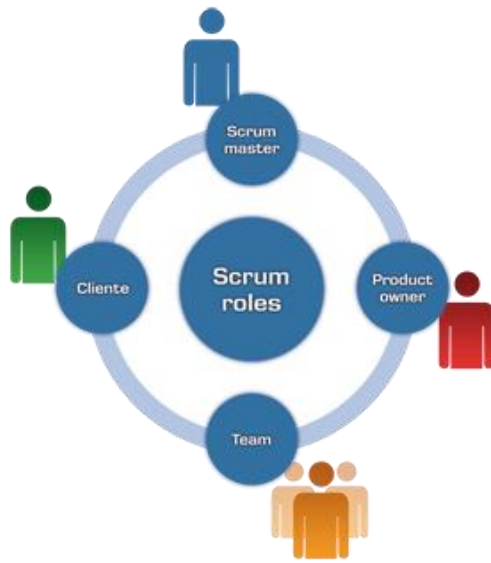


Figura 6 Roles de SCRUM

Fuente: (Softeng, 2019).

CAPÍTULO 2

2. DESARROLLO DE LA APLICACIÓN

En el presente capítulo detalla el proceso de las acciones ejecutadas durante el desarrollo del sistema FAM, siguiendo la metodología SCRUM, con la cual se ha agilizado el proceso de desarrollo y facilitar la documentación del mismo.

2.1 ROLES DE SCRUM

En la tabla se muestran los roles de las personas que se han involucrado en el desarrollo de la aplicación de FAM.

Tabla 2 Roles de SCRUM

ROL	NOMBRE	FUNCIÓN
Product Owner	Sr. Milton Vallejo Propietario de la FAM	Verificar las funcionalidades del sistema.
Scrum Master	MSc. Antonio Quiña	Verificar y facilitar el avance del desarrollo del sistema, así como el uso correcto de SCRUM.
Equipo Scrum	Carlos Proaño	Desarrollar el sistema en base a las especificaciones y requerimientos establecidos por el Product Owner.

Fuente: Propia

2.2 REQUERIMIENTOS Y PLANIFICACIÓN DE SPRINTS

Los requerimientos son establecidos como historias de usuario, realizadas al usuario final.

2.2.1 Historias de usuario

Las historias de usuario permiten detallar los requerimientos en base a las necesidades del Usuario, detalladas en la tabla 3.

- **Identificador:** identifica a la historia de usuario, la cual está compuesta por un número de tres cifras.
- **Enunciado de Historia:** está compuesto por subsecciones:
 - Rol: nos especifica el rol del usuario que va a actuar en la historia.
 - Característica/Funcionalidad: detalla la acción a realizar.
 - Razón/Resultado: da a conocer por qué necesita realizar dicho proceso.

- **Criterios de Aceptación:**
 - Criterio de aceptación: detalla los comportamientos de una función establecida.
 - Contexto: la posible situación en la que puede ocurrir un evento.
 - Evento: explica la acción a realizar
 - Resultado: es la respuesta del sistema ante un comportamiento determinado

Tabla 3 Historias de usuario

Enunciado de la historia				Criterios de aceptación				
ID	Rol	Característica / Funcionalidad	Razón / Resultado	#	Criterio de aceptación (Escenario)	Contexto	Evento	Resultado / Comportamiento esperado
001	Como administrador	Necesito registrar las plantas que serán sembradas	Con la finalidad de contar con un registro histórico de las plantas sembradas	1	Datos ingresados no válidos	En caso que los datos ingresados sean incorrectos o vacíos	Cuando presiono el botón de guardar del formulario de registro, o cambio de un campo a otro	El sistema muestra una advertencia, y señala el campo erróneo o faltante
				2	Datos ingresados válidos	En caso que la información ingresada sea válida	Cuando presiono el botón de guardar del formulario de registro	El sistema muestra un mensaje de confirmación, y redirige al usuario al listado plantas
002	Como administrador	Necesito editar/actualizar los registros de las plantas	Con la finalidad de que la información que se muestra sea la indicada	1	Datos ingresados no válidos	En caso que los datos ingresados sean incorrectos o vacíos	Cuando envío el formulario de registro, o cambio de un campo a otro	El sistema muestra una advertencia, y señala el campo erróneo o faltante
				2	Datos ingresados válidos	En caso que la información ingresada sea válida	Cuando envío formulario de registro	El sistema muestra un mensaje de confirmación, y redirige al usuario un listado
003	Como administrador	Necesito registrar los invernaderos en los cuales se sembrarán las plantas	Con la finalidad de contar con un registro de los invernaderos en los cuales se han sembrado las plantas	1	Datos ingresados no válidos	En caso que los datos ingresados sean incorrectos o vacíos	Cuando envío el formulario de registro, o cambio de un campo a otro	El sistema muestra una advertencia, y señala el campo erróneo o faltante
				2	Datos ingresados válidos	En caso que la información ingresada sea válida	Cuando envío formulario de registro	El sistema muestra un mensaje de confirmación, y redirige al usuario un listado
004	Como administrador	Necesito editar/actualizar los registros de los invernaderos	Con la finalidad de que la información que se muestra sea la indicada	1	Datos ingresados no válidos	En caso que los datos estén duplicados o incompletos	Cuando envío el formulario de registro, o cambio de un campo a otro	El sistema muestra una advertencia, y señala el campo erróneo o faltante

				2	Datos ingresados válidos	En caso que la información ingresada sea válida	Cuando envío formulario de registro	El sistema muestra un mensaje de confirmación, y redirige al usuario un listado
005	Como administrador	Necesito registrar los materiales que se van comprando/adquiriendo	Con la finalidad de contar con el stock de materiales en bodega	1	Datos ingresados no válidos	En caso que los datos ingresados sean incorrectos o vacíos	Cuando envío el formulario de registro, o cambio de un campo a otro	El sistema muestra una advertencia, y señala el campo erróneo o faltante
				2	Datos ingresados válidos	En caso que la información ingresada sea válida	Cuando envío formulario de registro	El sistema muestra un mensaje de confirmación, y redirige al usuario un listado
006	Como administrador	Necesito editar/actualizar los registros de los materiales adquiridos	Con la finalidad de que la información que se muestra sea la indicada	1	Datos ingresados no válidos	En caso que los datos ingresados sean incorrectos o vacíos	Cuando envío el formulario de registro, o cambio de un campo a otro	El sistema muestra una advertencia, y señala el campo erróneo o faltante
				2	Datos ingresados válidos	En caso que la información ingresada sea válida	Cuando envío formulario de registro	El sistema muestra un mensaje de confirmación, y redirige al usuario un listado
007	Como administrador	Necesito registrar los químicos que se van comprando/adquiriendo	Con la finalidad de recaudar valores y restar el el saldo de aportes de socio	1	Datos no válidos	En caso que los datos ingresados sean incorrectos o incompletos	Cuando envío formulario de registro	El sistema muestra error respectivo
				2	Socio con saldo insuficiente	En caso que el socio tenga saldo inferior al valor a registrar	Cuando envío formulario de registro	El sistema muestra error respectivo
008	Como administrador	Necesito editar/actualizar los registros de los químicos adquiridos	Con la finalidad de que la información que se muestra sea la indicada	1	Datos ingresados no válidos	En caso que los datos ingresados sean incorrectos o vacíos	Cuando envío el formulario de registro, o cambio de un campo a otro	El sistema muestra una advertencia, y señala el campo erróneo o faltante

				2	Datos ingresados válidos	En caso que la información ingresada sea válida	Cuando envió formulario de registro	El sistema muestra un mensaje de confirmación, y redirige al usuario un listado
009	Como administrador	Necesito registrar las siembras realizadas	Con la finalidad de contar con un histórico de siembras	1	Datos ingresados no válidos	En caso que los datos ingresados sean incorrectos o vacíos	Cuando envió el formulario de registro, o cambio de un campo a otro	El sistema muestra una advertencia, y señala el campo erróneo o faltante
				2	Datos ingresados válidos	En caso que la información ingresada sea válida	Cuando envió formulario de registro	El sistema muestra un mensaje de confirmación, y redirige al usuario un listado
010	Como administrador	Necesito editar/actualizar los registros de las siembras realizadas	Con la finalidad de que la información que se muestra sea la indicada	1	Datos ingresados no válidos	En caso que los datos ingresados sean incorrectos o incompletos	Cuando envió el formulario de registro, o cambio de un campo a otro	El sistema muestra error respectivo
				2	Datos ingresados válidos	En caso que la información ingresada sea válida	Cuando envió formulario de registro	El sistema muestra ingreso, gasto o donación creada
011	Como administrador	Necesito registrar las cosechas realizadas	Con la finalidad de contar con un histórico de cosechas	1	Datos ingresados no válidos	En caso que los datos ingresados sean incorrectos o vacíos	Cuando envió el formulario de registro, o cambio de un campo a otro	El sistema muestra una advertencia, y señala el campo erróneo o faltante
				2	Datos ingresados válidos	En caso que la información ingresada sea válida	Cuando envió formulario de registro	El sistema muestra un mensaje de confirmación, y redirige al usuario un listado
012	Como administrador	Necesito editar/actualizar los registros de las cosechas realizadas	Con la finalidad de que la información que se muestre sea la indicada	1	Datos ingresados no válidos	En caso que los datos ingresados sean incorrectos o vacíos	Cuando envió el formulario de registro, o cambio de un campo a otro	El sistema muestra una advertencia, y señala el campo erróneo o faltante
				2	Datos ingresados válidos	En caso que la información ingresada sea válida	Cuando envió formulario de registro	El sistema muestra un mensaje de confirmación, y redirige al usuario un listado

013	Como administrador	Necesito registrar las ventas realizadas	Con la finalidad de contar con un histórico de ventas	1	Datos ingresados no válidos	En caso que los datos ingresados sean incorrectos o vacíos	Cuando envío el formulario de registro, o cambio de un campo a otro	El sistema muestra una advertencia, y señala el campo erróneo o faltante
				2	Datos ingresados válidos	En caso que la información ingresada sea válida	Cuando envío formulario de registro	El sistema muestra un mensaje de confirmación, y redirige al usuario un listado
014	Como administrador	Necesito editar/actualizar los registros de las ventas realizadas	Con la finalidad de que la información que se muestre sea la indicada	1	Datos ingresados no válidos	En caso que los datos ingresados sean incorrectos o vacíos	Cuando envío el formulario de registro, o cambio de un campo a otro	El sistema muestra una advertencia, y señala el campo erróneo o faltante
				2	Datos ingresados válidos	En caso que la información ingresada sea válida	Cuando envío formulario de registro	El sistema muestra un mensaje de confirmación, y redirige al usuario un listado
015	Como administrador	Necesito registrar tratamientos que van a ser usados para nutrir, cuidar y mantener saludable a la planta	Con la finalidad de contar con un histórico de tratamientos	1	Datos ingresados no válidos	En caso que los datos ingresados sean incorrectos o vacíos	Cuando envío el formulario de registro, o cambio de un campo a otro	El sistema muestra una advertencia, y señala el campo erróneo o faltante
				2	Datos ingresados válidos	En caso que la información ingresada sea válida	Cuando envío el formulario de registro, o cambio de un campo a otro	El sistema muestra un mensaje de confirmación, y redirige al usuario un listado
016	Como administrador	Necesito editar/actualizar los registros de las tratamientos registrados	Con la finalidad de que la información que se muestre sea la indicada	1	Datos ingresados no válidos	En caso que los datos ingresados sean incorrectos o vacíos	Cuando envío el formulario de registro, o cambio de un campo a otro	El sistema muestra una advertencia, y señala el campo erróneo o faltante
				2	Datos ingresados válidos	En caso que la información ingresada sea válida	Cuando envío formulario de registro	El sistema muestra un mensaje de confirmación, y redirige al usuario un listado

017	Como administrador	Necesito registrar las aplicaciones de los tratamientos	Con la finalidad de tener un registro adecuado, de que tratamiento está siendo aplicado en una siembra específica	1	Datos ingresados no válidos	En caso que los datos ingresados sean incorrectos o vacíos	Cuando envío el formulario de registro, o cambio de un campo a otro	El sistema muestra una advertencia, y señala el campo erróneo o faltante
				2	Datos ingresados válidos	En caso que la información ingresada sea válida	Cuando envío formulario de registro	El sistema muestra un mensaje de confirmación, y redirige al usuario un listado
018	Como administrador	Necesito editar/actualizar los registros de los tratamientos aplicados realizadas	Con la finalidad de que la información que se muestre sea la indicada	1	Datos ingresados no válidos	En caso que los datos ingresados sean incorrectos o vacíos	Cuando envío el formulario de registro, o cambio de un campo a otro	El sistema muestra una advertencia, y señala el campo erróneo o faltante
				2	Datos ingresados válidos	En caso que la información ingresada sea válida	Cuando envío formulario de registro	El sistema muestra un mensaje de confirmación, y redirige al usuario un listado
019	Como administrador	Necesito registrar los mantenimientos realizados a los invernaderos	Con la finalidad de tener un registro adecuado de los mantenimientos que han sido realizados	1	Datos ingresados no válidos	En caso que los datos ingresados sean incorrectos o vacíos	Cuando envío el formulario de registro, o cambio de un campo a otro	El sistema muestra una advertencia, y señala el campo erróneo o faltante
				2	Datos ingresados válidos	En caso que la información ingresada sea válida	Cuando envío formulario de registro	El sistema muestra un mensaje de confirmación, y redirige al usuario un listado
020	Como administrador	Necesito editar/actualizar los registros de los mantenimientos realizadas	Con la finalidad de que la información que se muestre sea la indicada	1	Datos ingresados no válidos	En caso que los datos ingresados sean incorrectos o vacíos	Cuando envío el formulario de registro, o cambio de un campo a otro	El sistema muestra una advertencia, y señala el campo erróneo o faltante
				2	Datos ingresados válidos	En caso que la información ingresada sea válida	Cuando envío formulario de registro	El sistema muestra un mensaje de confirmación, y redirige al usuario un listado

021	Como administrador	Necesito visualizar los datos críticos en un listado	Con la finalidad de poder ver datos cruciales al ingresar a la pantalla de listados	1	Visualización de datos	En caso de ingresar a un página de listado, pueden ser de materiales, químicos, invernaderos, plantas, siembras, cosechas, ventas, tratamientos y tratamientos aplicados	Cuando ingrese a la página de listado de materiales, químicos, invernaderos, plantas, siembras, cosechas, ventas, tratamientos y tratamientos aplicados	El sistema muestra un listado teniendo como primera fila, al último registro ingresado, y se irán listando acorde a este criterio
022	Como administrador	Necesito exportar las tablas de los datos críticos	Con la finalidad de tener un respaldo extra	1	Exportación de tablas	En caso de necesitar exportar los datos	Al presionar la opción de exportar	El sistema mostrará un botón "Exportar" para que pueda generar un archivo pdf o excel.
023	Como administrador	Necesito una página principal en la que se pueda apreciar datos importantes	Con la finalidad de tener como primera instancia datos que son cruciales para no tener la necesidad de ir a revisar un listado en específico	1	Calendario con la fecha de las siembras	En caso que prefiera dar un breve vistazo al sistema	Cuando se ingresa en el sistema	El sistema muestra automáticamente un calendario, que se visualiza por semanas
				2	Gráfico de barras con la cantidad de materiales en stock	En caso que prefiera dar un breve vistazo al sistema	Cuando se ingresa en el sistema	El sistema muestra automáticamente un gráfico con los materiales existentes
				3	Gráfico de barras con la cantidad de químicos en stock	En caso que prefiera dar un breve vistazo al sistema	Cuando se ingresa en el sistema	El sistema muestra automáticamente un gráfico con los químicos existentes
				4	Gráfico de barras de las ventas realizadas	En caso que prefiera dar un breve vistazo al sistema	Cuando se ingresa en el sistema	El sistema muestra automáticamente con las ventas realizadas
				5	Gráfico de barras con la sumatoria de ventas por meses	En caso que prefiera dar un breve vistazo al sistema	Cuando se ingresa en el sistema	El sistema muestra automáticamente con la sumatoria de ventas realizadas, ordenadas por meses

024	Como administrador	Necesito tener acceso al sistema	Con la finalidad de hacer uso del sistema, confiando de que ninguna persona desconocida pueda hacerlo	1	Datos ingresados no válidos	En caso que los datos ingresados sean incorrectos o vacíos	Cuando envío el formulario de registro, o cambio de un campo a otro	El sistema muestra una advertencia, y señala el campo erróneo o faltante
				2	Datos ingresados válidos	En caso que la información ingresada sea válida	Cuando envío formulario de registro	El sistema muestra un mensaje de confirmación, y redirige al usuario a la página principal
025	Como administrador	Necesito terminar la sesión en el sistema	Con la finalidad de evitar intrusiones	1	Menú salir	-	Cuando da clic en la opción de salir	El sistema cierra la sesión y redirige al usuario al página de ingreso al sistema

Fuente: Propia

2.2.2 Product backlog

Tomando como base las historias de usuario, se detalla el esfuerzo en términos de horas, al cual ha sido asignado, de acuerdo a la lógica del negocio.

Tabla 4 Product backlog

Id de Historia	Nombre	Descripción	Dimensión/ Esfuerzo	Prioridad
001	Registro de plantas	Como administrador, Necesito registrar las plantas que serán sembradas , Con la finalidad de contar con un registro histórico de las plantas sembradas	60	1
002	Actualizar plantas	Como administrador, Necesito editar/actualizar los registros de las plantas , Con la finalidad de que la información que se muestra sea la indicada	60	2
003	Registro de invernaderos	Como administrador, Necesito registrar los invernaderos en los cuales se sembrarán los plantas , Con la finalidad de contar con un registro de los invernaderos en los cuales se han sembrado las plantas	43	5
004	Actualizar invernaderos	Como administrador, Necesito editar/actualizar los registros de los invernaderos , Con la finalidad de que la información que se muestra sea la indicada	45	3
005	Registro de materiales	Como administrador, Necesito registrar los materiales que se van comprando/adquiriendo , Con la finalidad de contar con el stock de materiales en bodega	60	4
006	Actualizar materiales	Como administrador, Necesito editar/actualizar los registros de los materiales adquiridos , Con la finalidad de que la información que se muestra sea la indicada	60	6
007	Registro de químicos	Como administrador, Necesito registrar los químicos que se van comprando/adquiriendo , Con la finalidad de contar con el stock de químicos en bodega	60	7
008	Actualizar químicos	Como administrador, Necesito editar/actualizar los registros de los químicos adquiridos , Con la finalidad de que la información que se muestra sea la indicada	60	8
009	Registro de siembras	Como administrador, Necesito registrar las siembras realizadas , Con la finalidad de contar con un histórico de siembras	75	11
010	Actualizar siembras	Como administrador, Necesito editar/actualizar los registros de las siembras realizadas , Con la finalidad de que la información que se muestra sea la indicada	75	12
011	Registro de cosechas	Como administrador, Necesito registrar las cosechas realizadas , Con la finalidad de contar con un histórico de cosechas	75	13
012	Actualizar cosechas	Como administrador, Necesito editar/actualizar los registros de las cosechas realizadas , Con la finalidad de que la información que se muestre sea la indicada	75	14

013	Registro de ventas	Como administrador, Necesito registrar las ventas realizadas , Con la finalidad de contar con un histórico de ventas	75	10
014	Actualizar ventas	Como administrador, Necesito editar/actualizar los registros de las ventas realizadas , Con la finalidad de que la información que se muestre sea la indicada	75	15
015	Registro de tratamientos	Como administrador, Necesito registrar tratamientos que van a ser usados para nutrir, cuidar y mantener saludable a la planta , Con la finalidad de contar con un histórico de tratamientos	90	20
016	Actualizar tratamientos	Como administrador, Necesito editar/actualizar los registros de las tratamientos registrados , Con la finalidad de que la información que se muestre sea la indicada	90	16
017	Registro de tratamientos aplicados	Como administrador, Necesito registrar las aplicaciones de los tratamientos , Con la finalidad de tener un registro adecuado, de que tratamiento está siendo aplicado en una siembra específica	90	21
018	Actualizar tratamientos aplicados	Como administrador, Necesito editar/actualizar los registros de los tratamientos aplicados realizadas , Con la finalidad de que la información que se muestre sea la indicada	90	17
019	Registro de mantenimientos	Como administrador, Necesito registrar los mantenimientos realizados a los invernaderos, Con la finalidad de contar con un histórico de mantenimientos	80	18
020	Actualizar mantenimientos	Como administrador, Necesito editar/actualizar los registros de los mantenimientos realizadas , Con la finalidad de que la información que se muestre sea la indicada	80	19
021	Visualización de datos	Como administrador, Necesito visualizar los datos críticos en un listado , Con la finalidad de poder ver datos cruciales al ingresar a la pantalla de listados	90	22
022	Exportación de los datos visualizados	Como administrador, Necesito exportar las tablas de los datos críticos , Con la finalidad de tener un respaldo extra	70	24
023	Creación de un dashboard	Como administrador, Necesito una página principal en la que se pueda apreciar datos importantes , Con la finalidad de tener como primera instancia datos que son cruciales para no tener la necesidad de ir a revisar un listado en específico	75	23
024	Iniciar sesión	Como administrador, Necesito tener acceso al sistema, Con la finalidad de hacer uso del sistema, confiando de que ninguna persona desconocida pueda hacerlo	80	25
025	Cerrar sesión	Como administrador, Necesito terminar la sesión en el sistema, Con la finalidad de evitar intrusiones	80	26

Fuente: propia

2.2.3 Planificación del proyecto

En este proceso se determina resumen, las fechas de inicio y fin, tomando en cuenta jornadas de trabajo de ocho horas diarias por cinco días a la semana. Se establece que los sprints tendrán una duración de 4 semanas.

Tabla 5 Planificación del proyecto

ID	NOMBRE DE ITERACIÓN	FECHA INICIAL	FECHA FINAL	DIAS	HORAS
Sprint 0	Análisis y planificación del proyecto	29/01/2018	23/2/2018	20	100
Sprint 1	Gestión de plantas e invernaderos	05/3/2018	30/3/2018	20	160
Sprint 2	Gestión de insumos	30/04/2018	25/05/2018	20	160
Sprint 3	Gestión de Tratamientos y Mantenimientos	28/05/2018	22/06/2018	20	160
Sprint 4	Gestión de Siembras y Tratamientos Aplicados	09/07/2018	03/08/2018	20	160
Sprint 5	Gestión de Cosechas y Ventas	13/08/2018	07/09/2018	20	160
Sprint 6	Gestión del Dashboard	17/09/2018	12/10/2018	20	160
Sprint 7	Gestión de Acceso	05/11/2018	30/11/2018	20	160

Fuente: Propia

2.3 EJECUCIÓN

Se definieron los roles de usuario y responsabilidades del proyecto, el siguiente paso es desarrollar los sprints para obtener un incremento del desarrollo del sistema hasta llegar a obtener el producto final.

2.3.1 Sprint 0 Análisis y planificación del proyecto

El sprint 0, es una fase en donde se configuran las herramientas, dejando una sólida base para comenzar con los demás sprints.

a. Planificación del sprint

Se la realizo con el Product Owner, Scrum Master y el equipo de desarrollo la fecha de 29/01/2018.

Tabla 6 Planificación del sprint 0

ID	Sprint 0
NOMBRE DE ITERACIÓN	Análisis y planificación del proyecto
FECHA INICIAL	29/01/2018
FECHA FINAL	23/02/2018

Fuente: Propia

De la reunión se determinó el sprint backlog, en el que se desarrollan las actividades propuestas en el sprint 0.

Tabla 7 Sprint 0

Sprint	S0 Análisis y planificación del proyecto	Inicio	29/01/2018
Responsable	Carlos Proaño	Fin	23/02/2018
Estado	TERMINADO	Duración	160
HISTORIA DE USUARIO	TAREAS	TIPO	
	Descarga e Instalación de Maven	Análisis	
	Descarga e Instalación de Java	Análisis	
	Descarga e Instalación de Node.js	Análisis	
	Descarga e Instalación de MySQL versión 5.7	Análisis	
	Descarga e Instalación de Eclipse Neon.3 Release	Análisis	
	Descarga e Instalación de Angular - cli	Análisis	
	Prueba de plantilla básica de Angular 4	Análisis	
	Diseño de modelo entidad relación	Análisis	
	Aprendizaje de Spring Boot	Análisis	
	Aprendizaje de Angular 4	Análisis	
	Corrección de modelo entidad relación	Revisión	
	Instalación de plantilla avanzada de Angular 4	Análisis	
	Configuración del proyecto	Prueba	
	Definición y configuración del Navbar	Desarrollo	
	Definición y configuración del Sidebar	Desarrollo	
	Configuración de la rutas - routes	Desarrollo	
	Edición de Visualización de Sitio Principal	Desarrollo	

Fuente: Propia

b. Revisión del sprint

Realizado el debido levantamiento de requisitos, se deja definida la arquitectura del proyecto, el diagrama de base de datos entidad – relación y el diagrama de procesos, piezas clave para comenzar con el desarrollo de la aplicación.

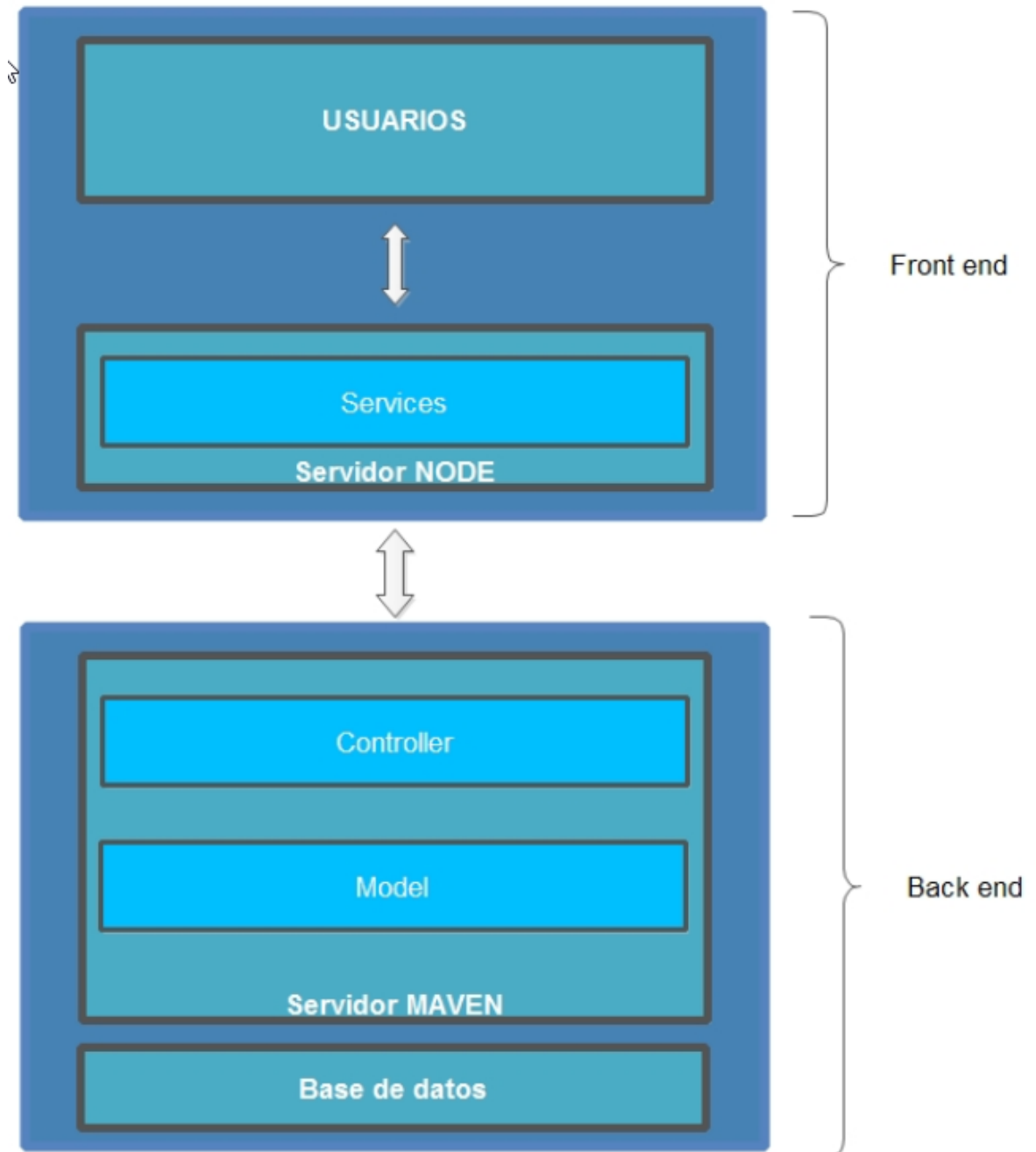


Figura 7 Arquitectura del Proyecto

Fuente: Propia

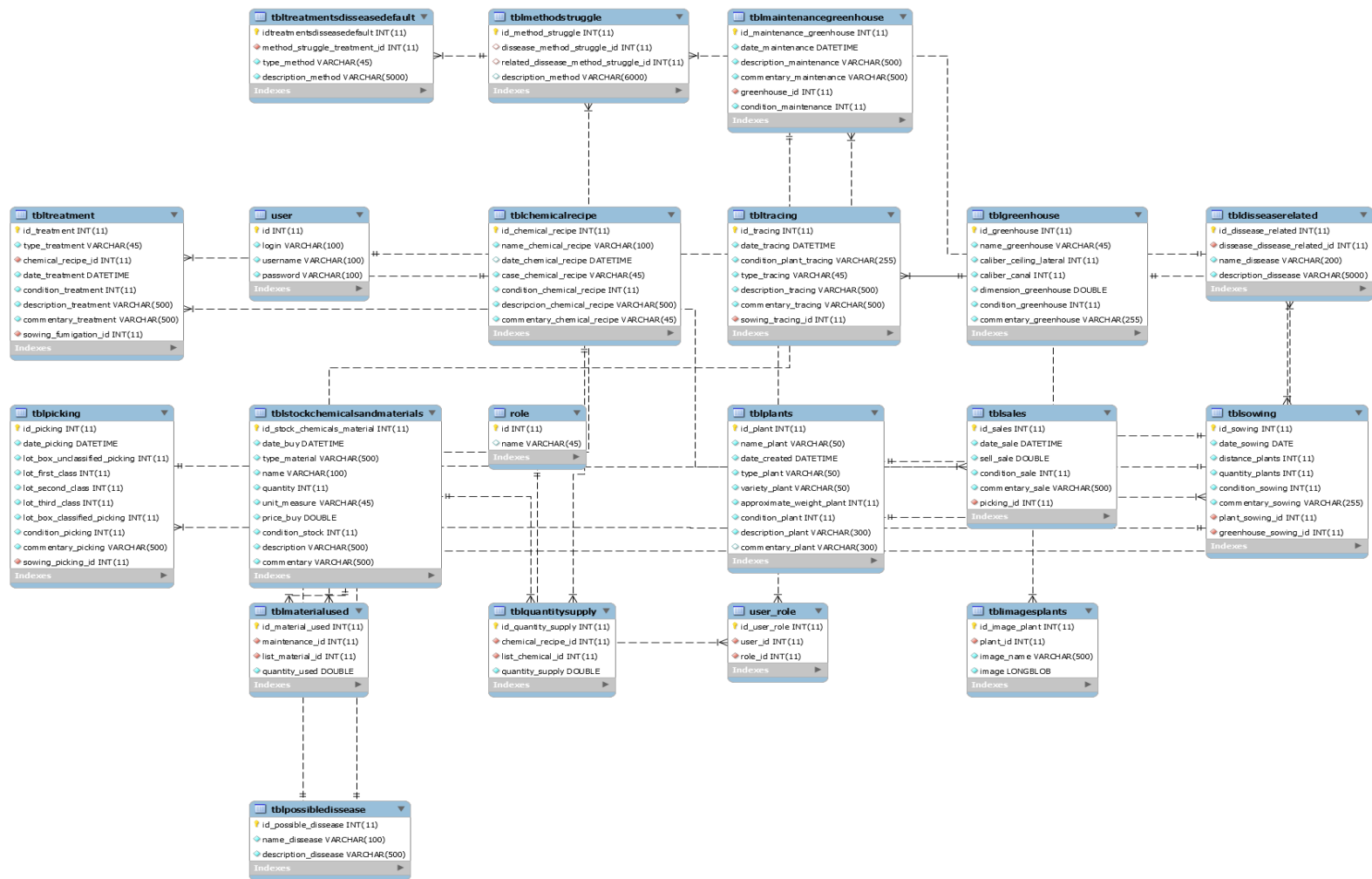


Figura 8 Diagrama Entidad Relación

Fuente: Propia

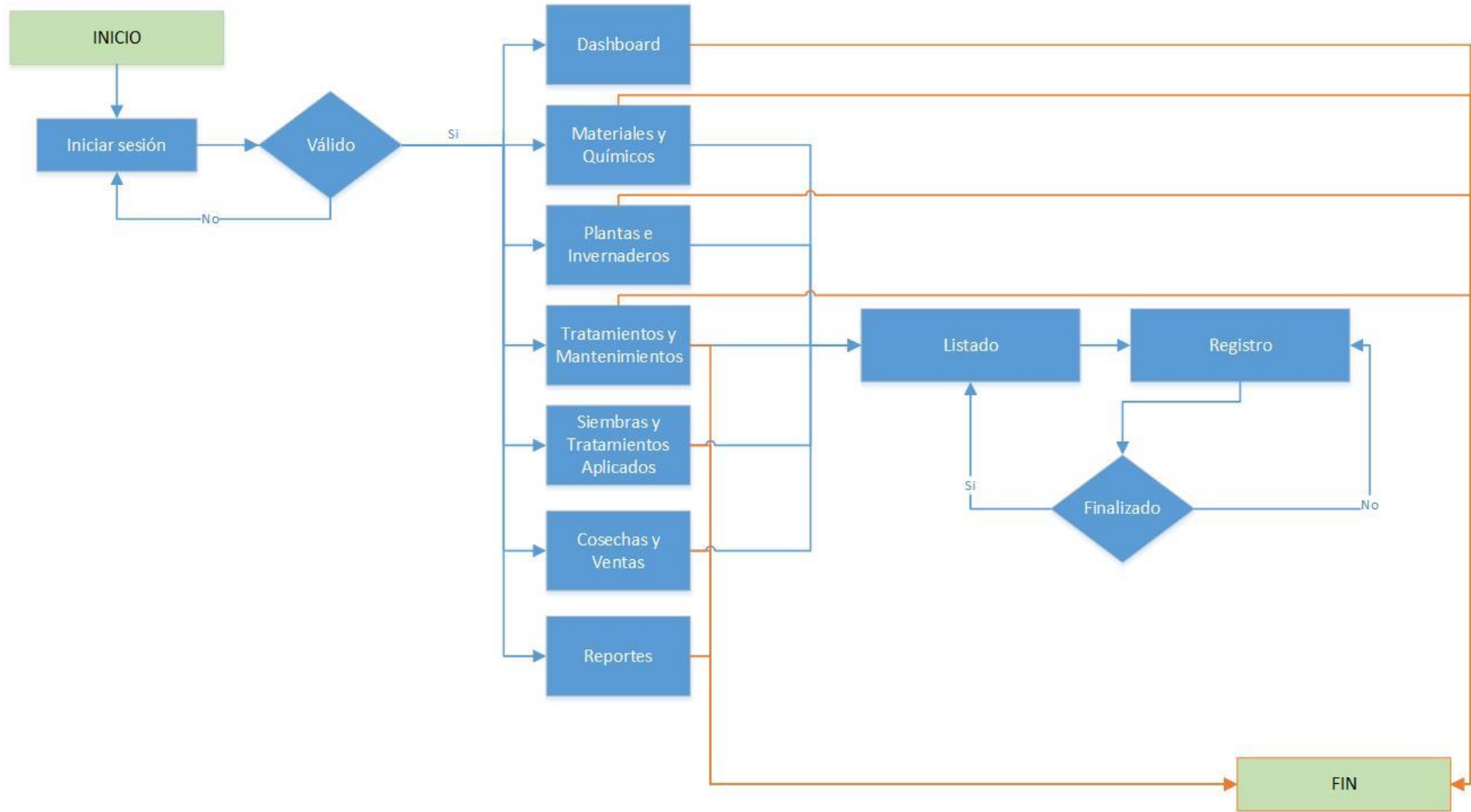


Figura 9 Diagrama de procesos

Fuente: Propia

c. Revisión retrospectiva

El product owner esta gustoso de que se involucre en el proceso de levantamiento de requerimientos, con lo cual nos ofrece su ayuda para el desarrollo del sistema.

Se presentaron muchos problemas al momento de realizar la integración de las herramientas, debido a las versiones usadas en primera instancia.

Se debe realizar una exhaustiva investigación para saber que versiones se pueden utilizar.

2.3.2 Sprint 1 Gestión de Plantas e Invernaderos

En este sprint se procede a desarrollar las funcionalidades del módulo de Plantas e Invernaderos.

a. Planificación del sprint

Se la realizo con el Product Owner, Scrum Master y el equipo de desarrollo la fecha de 05/03/2018.

Tabla 8 Planificación del sprint 1

ID	Sprint 1
NOMBRE DE ITERACIÓN	Gestión de plantas e invernaderos
FECHA INICIAL	05/03/2018
FECHA FINAL	30/03/2018

Fuente: Propia

De la reunión se determinó el sprint backlog, que se conforman por las historias de usuario que se desarrollarán en el Sprint 1.

Tabla 9 Sprint backlog del sprint 1

Sprint	S1 Gestión de plantas e invernaderos	Inicio	05/3/2018
Responsable	Carlos Proaño	Fin	30/3/2018
Estado	TERMINADO	Duración	160
HISTORIA DE USUARIO	TAREAS	TIPO	
Registro de plantas	Creación del modelo "Plants" de la tabla "tblplants" en el model de springBoot	Desarrollo	
	Creación del repositorio para el modelo "Plants"	Desarrollo	
	Creación del servicio web rest "save" para el registro de las plantas	Desarrollo	
	Instalación de los servicios web	Desarrollo	
	Creación de la clase "plant.service" y agregar el método de "guardar" para que capture el servicio web	Desarrollo	
	Creación del módulo de Registro de Plantas para el front-end	Desarrollo	

	Creación y agregamiento de las clases module.ts y routing.ts al módulo de registro de plantas del front-end	Desarrollo
	Creación del routing para la navegación dentro del sistema web	Desarrollo
	Creación del formulario para el registro de plantas en el front-end	Desarrollo
	Validación de los campos del registro	Desarrollo
	Creación del modelo "plant.model" en el front-end	Desarrollo
	Asignación del modelo "plant.model" a cada campo del formulario	Desarrollo
	Asignación de la clase "plant.service" al module.ts del registro de plantas	Desarrollo
	Creación del método guardar	Desarrollo
Actualizar plantas	Creación del servicio web rest para la actualización del registro de las plantas	Desarrollo
	Instalación de los servicios web	Desarrollo
	Agregar el método de "editar" en la clase "plant.service" para que capture el servicio web	Desarrollo
	Creación del método para que los datos a editar se muestren en el registro de plantas en el front-end	Desarrollo
	Asignación del modelo "plant.model" a cada campo del formulario	Desarrollo
	Creación del método editar	Desarrollo
Visualización de los registros de plantas	Creación del servicio web rest para la visualización del listado plantas	Desarrollo
	Instalación de los servicios web	Desarrollo
	Agregar el método de "findAll" en la clase "plant.service" para que capture el servicio web	Desarrollo
	Creación del módulo de Listado de Plantas para el front-end	Desarrollo
	Creación y agregamiento de las clases module.ts y routing.ts al módulo de listado de plantas del front-end	Desarrollo
	Creación del routing para la navegación dentro del sistema web	Desarrollo
	Creación del formulario para la visualización de los registros de plantas en el front-end	Desarrollo
	Asignación de la clase "plant.service" al module.ts del registro de listado de plantas	Desarrollo
	Creación del método "getPlants"	Desarrollo
	Importación del componente "ngx-data-table"	Desarrollo
	Mostrar los datos dentro del componente	Desarrollo
Filtros de búsqueda en el registros de plantas	Creación del servicio web rest para la visualización del listado plantas	Desarrollo
	Instalación de los servicios web	Desarrollo
	Agregar el método de "getPlantByCriteria" en la clase "plant.service" para que capture el servicio web	Desarrollo
	Creación de los campos de búsqueda en el módulo de listado de plantas	Desarrollo
	Creación del método para realizarla búsqueda en el front-end	Desarrollo
Registro de invernaderos	Creación del modelo "Greenhouse" de la tabla "tblgreenhouse" en el model de springBoot	Desarrollo
	Creación del repositorio para el modelo "Greenhouse"	Desarrollo
	Creación del servicio web rest "save" para el registro de los invernaderos	Desarrollo
	Instalación de los servicios web	Desarrollo
	Creación de la clase "greenhouse.service" y agregar el método de "guardar" para que capture el servicio web	Desarrollo
	Creación del módulo de Registro de invernaderos para el front-end	Desarrollo
	Creación y agregamiento de las clases module.ts y routing.ts al módulo de registro de invernaderos del front-end	Desarrollo

	Creación del routing para la navegación dentro del sistema web	Desarrollo
	Creación del formulario para el registro de invernaderos en el front-end	Desarrollo
	Validación de los campos del registro	Desarrollo
	Creación del modelo "greenhouse.model" en el front-end	Desarrollo
	Asignación del modelo "greenhpuse.model" a cada campo del formulario	Desarrollo
	Asignación de la clase "greenhouse.service" al module.ts del registro de invernaderos	Desarrollo
	Creación del método guardar	Desarrollo
Actualizar plantas	Creación del servicio web rest para la actualización del registro de invernaderos	Desarrollo
	Instalación de los servicios web	Desarrollo
	Agregar el método de "editar" en la clase "greenhouse.service" para que capture el servicio web	Desarrollo
	Creación del método para que los datos a editar se muestren en el registro de invernaderos en el front-end	Desarrollo
	Asignación del modelo "greenhouse.model" a cada campo del formulario	Desarrollo
	Creación del método editar	Desarrollo
Visualización de los registros de invernaderos	Creación del servicio web rest para la visualización del listado de invernaderos	Desarrollo
	Instalación de los servicios web	Desarrollo
	Agregar el método de "findAll" en la clase "greenhouse.service" para que capture el servicio web	Desarrollo
	Creación del módulo de Listado de Invernaderos para el front-end	Desarrollo
	Creación y agregamiento de las clases module.ts y rounting.ts al módulo de listado de invernaderos del front-end	Desarrollo
	Creación del routing para la navegación dentro del sistema web	Desarrollo
	Creación del formulario para la visualización de los registros de invernaderos en el front-end	Desarrollo
	Asignación de la clase "plant.service" al module.ts del registro de listado de invernaderos	Desarrollo
	Creación del método "getGreenhouse"	Desarrollo
	Importación del componente "ngx-data-table"	Desarrollo
	Mostrar los datos dentro del componente	Desarrollo
Filtros de búsqueda en el registros de invernaderos	Creación del servicio web rest para la visualización del listado de invernaderos	Desarrollo
	Instalación de los servicios web	Desarrollo
	Agregar el método de "getGreenhouseByCriteria" en la clase "greenhouse.service" para que capture el servicio web	Desarrollo
	Creación de los campos de búsqueda en el módulo de listado de invernaderos	Desarrollo
	Creación del método para realizarla búsqueda en el front-end	Desarrollo
Exportación de datos	Instalación de componentes para la exportación de datos	Desarrollo
	Creación de los métodos	Desarrollo

Fuente: Propia

b. Revisión del sprint

Ya realizadas las tareas del sprint y ejecutado las pruebas pertinentes, se entrega al módulo como terminado, a continuación, se muestran las pantallas del módulo finalizado.

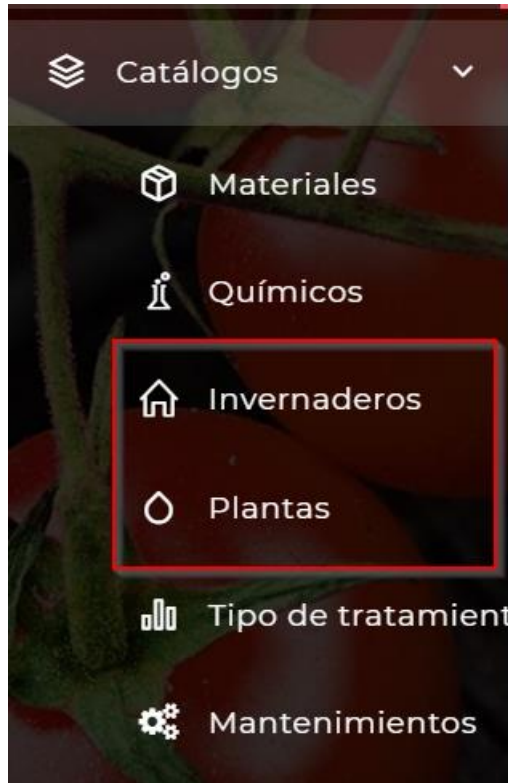


Figura 10 SideBar de Plantas e Invernaderos

Fuente: Propia

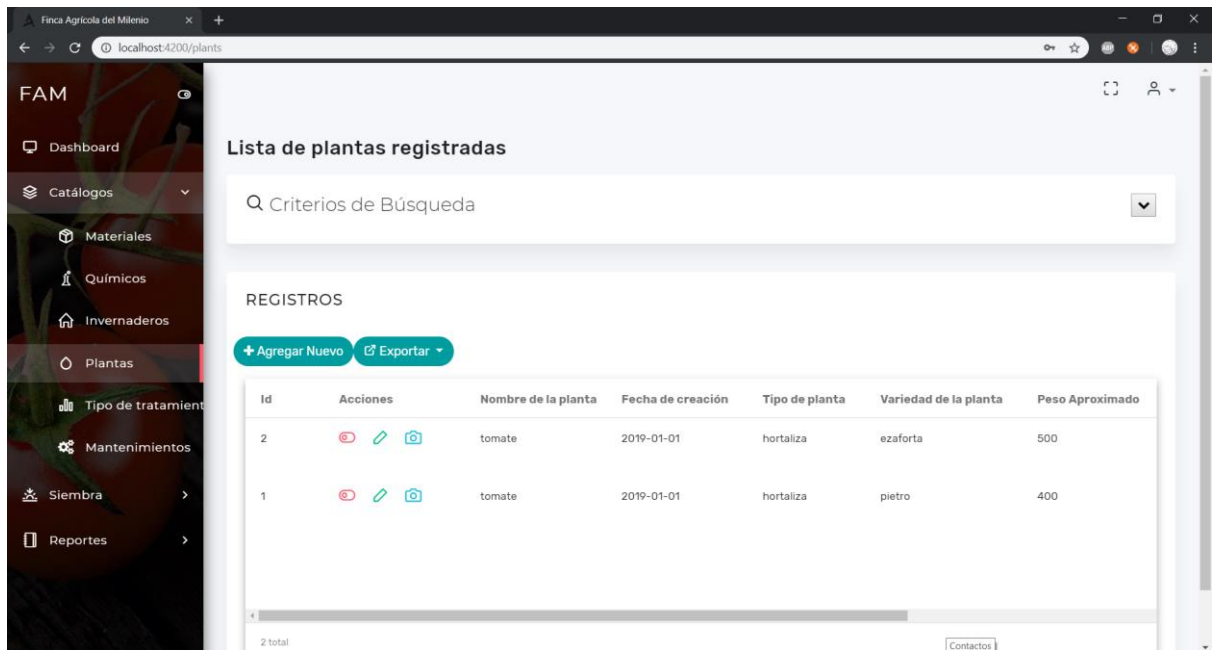


Figura 11 Listado de Plantas Registradas

Fuente: Propia

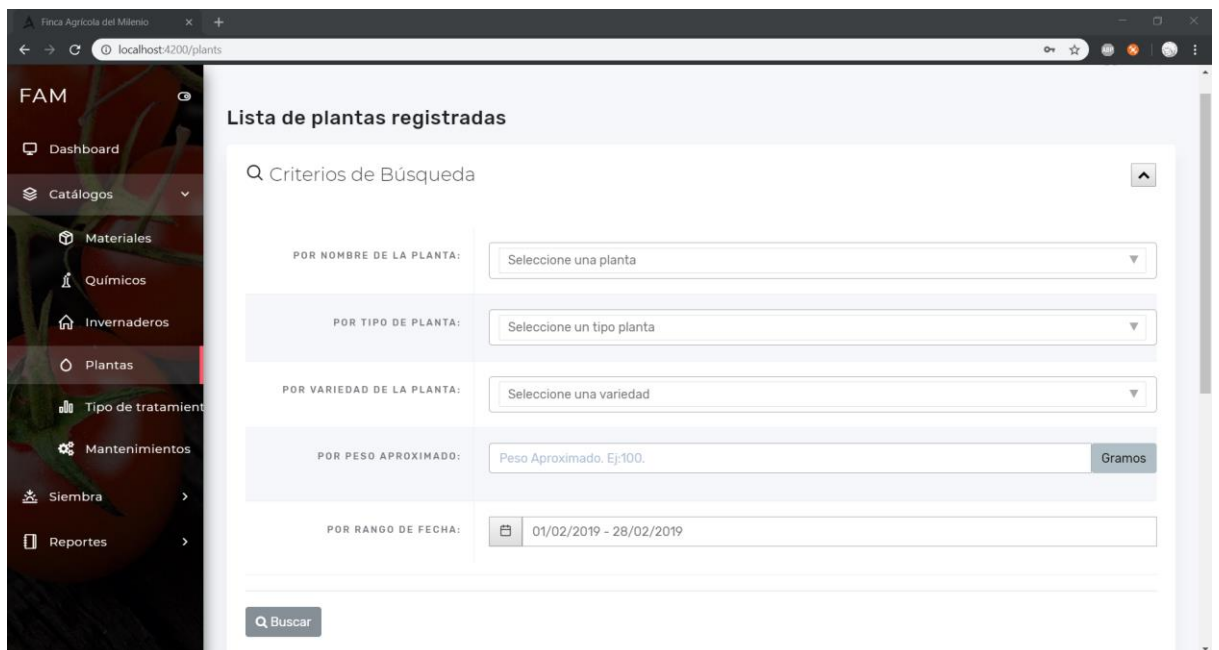


Figura 12 Criterios de Búsqueda registro de plantas

Fuente: Propia

INGRESO DE PLANTAS

Información General

NOMBRE DE LA PLANTA: tomate

FECHA DE VENTA: 2019-02-01

TIPO DE PLANTA: hortaliza

VARIEDAD DE LA PLANTA: pietro

PESO APROXIMADO DEL FRUTO: -4 Gramos
Este campo es obligatorio y debe ser mayor a cero.

DESCRIPCIÓN DE LA PLANTA: tiene demanda en el mercado

COMENTARIO DE LA PLANTA: primera siembra de esta variedad de tomate

IMÁGENES DE LA PLANTA (OPCIONAL): Click para subir las imágenes

Cancelar Guardar

Figura 13 Formulario de registro de plantas

Fuente: Propia

Finca Agrícola del Milenio

localhost:4200/listGreenhouse

FAM

- Dashboard
- Catálogos
- Materiales
- Químicos
- Invernaderos
- Plantas
- Tipo de tratamiento
- Mantenimientos
- Siembra
- Reportes

Lista de invernaderos registrados

Criterios de Búsqueda

REGISTROS

+ Agregar Nuevo Exportar +

Id	Acciones	Nombre invernadero	Calibre Techo & Laterales	Calibre Canal	Dimensiones en m ²	Descripción del invernadero
3		modulo3	8	9	1500	invernadero metálico con semiredondas
2		modulo2	7	8	600	invernadero metálico con semiredondas
1		modulo1	7	8	400	invernadero metálico con semiredondas

3 total

Figura 14 Registro de invernaderos

Fuente: Propia

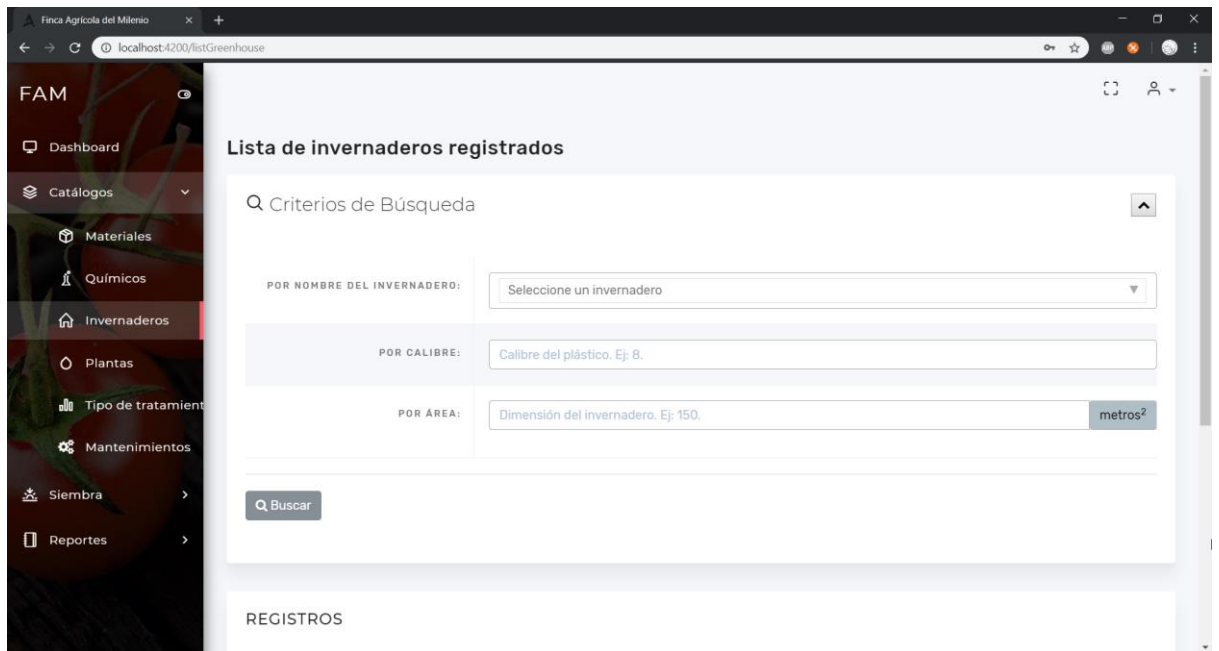


Figura 15 Criterios de Búsqueda registro de invernaderos.

Fuente: Propia



Figura 16 Formulario de registro de invernaderos

Fuente: Propia

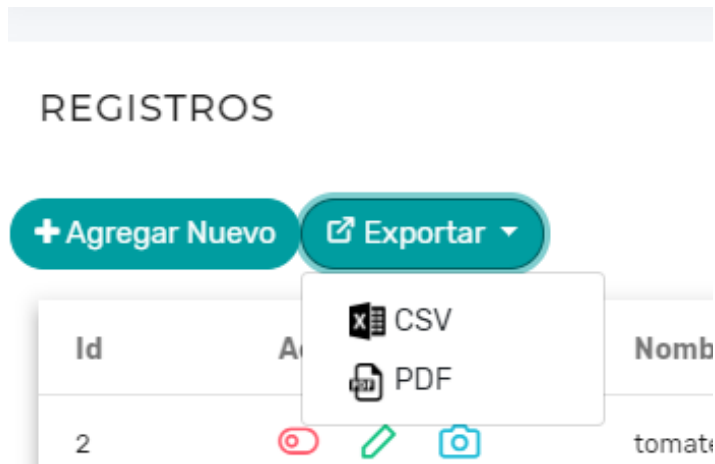


Figura 17 Exportación de archivos, plantas e invernaderos

Fuente: Propia

c. Revisión retrospectiva

Durante el desarrollo del sprint las reuniones planificadas se efectuaron sin ninguna demora, lo cual facilitó el ágil desarrollo de las tareas propuestas.

Existieron problemas en el desarrollo, al momento de guardar las imágenes de las plantas, ya que el componente seleccionado tenía problemas con la versión de angular usada, y debido a esto se tuvo que realizar varias pruebas con diferentes componentes.

Es recomendable realizar una investigación previa de la compatibilidad de los componentes que se serán usados y la versión de angular seleccionada para el desarrollo, con el fin de evitar problemas de compatibilidad.

2.3.3 Sprint 2 Gestión de Insumos

En este sprint se desarrollan los módulos de Químicos y Materiales.

a. Planificación del sprint

Se la realizo con el Product Owner, Scrum Master y el equipo de desarrollo la fecha de 30/04/2018.

Tabla 10 Planificación del Sprint 2

ID	Sprint 2
NOMBRE DE ITERACIÓN	Gestión de insumos
FECHA INICIAL	30/04/2018
FECHA FINAL	25/05/2018

Fuente: Propia

De la reunión se determinó el sprint backlog, que se conforman por las historias de usuario que se desarrollarán en el Sprint 2.

Tabla 11 Sprint backlog del sprint 2

Sprint	S2 Gestión de insumos	Inicio	30/04/2018
Responsable	Carlos Proaño	Fin	25/05/2018
Estado	TERMINADO	Duración	160
HISTORIA DE USUARIO	TAREAS	TIPO	
Registro de insumo	Creación del modelo "StockChemicalsAndMaterials" de la tabla "tblstockchemicalsandmaterials" en el model de springBoot	Desarrollo	
	Creación del repositorio para el modelo "StockChemicalsAndMaterials"	Desarrollo	
	Creación del servicio web rest "save" para el registro de insumos	Desarrollo	
	Instalación de los servicios web	Desarrollo	
	Creación de la clase "stock.service" y agregar el método de "guardar" para que capture el servicio web	Desarrollo	
	Creación del módulos de Registro de Materiales para el front-end	Desarrollo	
	Creación del módulos de Registro de Químicos para el front-end	Desarrollo	
	Creación y agregamiento de las clases module.ts y routing.ts al módulo de registro de materiales del front-end	Desarrollo	
	Creación y agregamiento de las clases module.ts y routing.ts al módulo de registro de químicos del front-end	Desarrollo	
	Creación del routing para la navegación dentro del sistema web	Desarrollo	
	Creación del formulario para el registro de materiales en el front-end	Desarrollo	
Creación del formulario para el registro de químicos en el front-end	Desarrollo		

	Validación de los campos del registro	Desarrollo
	Creación del modelo "stock.model" en el front-end	Desarrollo
	Asignación del modelo "stock.model" a cada campo del formulario	Desarrollo
	Asignación de la clase "stock.service" al module.ts del registro de materiales	Desarrollo
	Asignación de la clase "stock.service" al module.ts del registro de químicos	Desarrollo
	Creación del método guardar	Desarrollo
Actualizar insumos	Creación del servicio web rest para la actualización del registro de los insumos (materiales y químicos)	Desarrollo
	Instalación de los servicios web	Desarrollo
	Agregar el método de "editar" en la clase "stock.service" para que capture el servicio web	Desarrollo
	Creación del método para que los datos a editar se muestren en el registro de materiales en el front-end	Desarrollo
	Creación del método para que los datos a editar se muestren en el registro de químicos en el front-end	Desarrollo
	Asignación del modelo "stock.model" a cada campo del formulario	Desarrollo
	Creación del método editar	Desarrollo
Visualización de los registros de los insumos (Materiales y químicos)	Creación del servicio web rest para la visualización del listado insumos	Desarrollo
	Instalación de los servicios web	Desarrollo
	Agregar el método de "findAll" en la clase "stock.service" para que capture el servicio web	Desarrollo
	Creación del módulo de Listado de Materiales para el front-end	Desarrollo
	Creación del módulo de Listado de Químicos para el front-end	Desarrollo
	Creación y agregamiento de las clases module.ts y routing.ts al módulo de listado de materiales del front-end	Desarrollo
	Creación y agregamiento de las clases module.ts y routing.ts al módulo de listado de químicos del front-end	Desarrollo
	Creación del routing para la navegación dentro del sistema web	Desarrollo
	Creación del formulario para la visualización de los registros de materiales en el front-end	Desarrollo
	Creación del formulario para la visualización de los registros de químicos en el front-end	Desarrollo
	Asignación de la clase "stock.service" al module.ts del registro de listado de materiales	Desarrollo
	Asignación de la clase "stock.service" al module.ts del registro de listado de químicos	Desarrollo
	Creación del método "getMaterials"	Desarrollo
	Creación del método "getChemicals"	Desarrollo
	Importación del componente "ngx-data-table"	Desarrollo
	Mostrar los datos dentro del componente	Desarrollo
Filtros de búsqueda en el registros de insumos	Creación del servicio web rest para la visualización del listado insumos	Desarrollo
	Instalación de los servicios web	Desarrollo
	Agregar el método de "getStockByCriteria" en la clase "stock.service" para que capture el servicio web	Desarrollo
	Creación de los campos de búsqueda en el módulo de listado de materiales	Desarrollo

	Creación de los campos de búsqueda en el módulo de listado de químicos	Desarrollo
	Creación del método para realizarla búsqueda en el front-end	Desarrollo
Exportación de datos	Instalación de componentes para la exportación de datos	Desarrollo
	Creación de los métodos	Desarrollo

Fuente: Propia

b. Revisión del sprint

Ya realizadas las tareas del sprint y ejecutado las pruebas pertinentes, se entrega al módulo como terminado, a continuación, se muestran las pantallas del módulo finalizado.



Figura 18 SideBar para Materiales y Químicos

Fuente: Propia

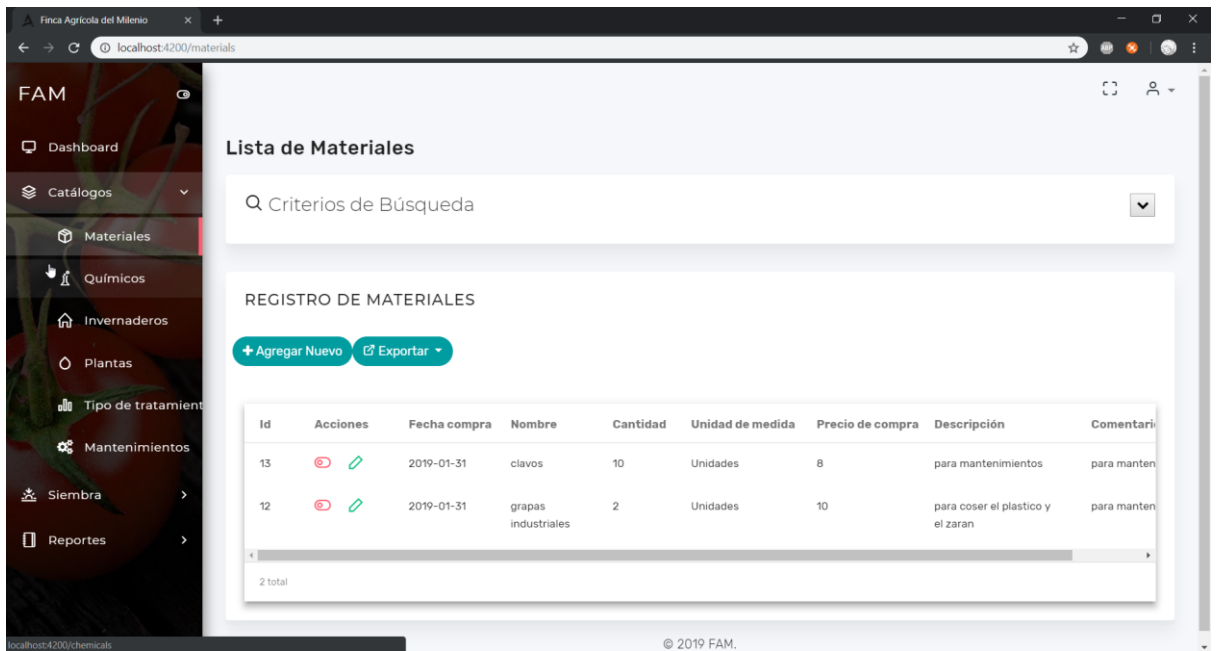


Figura 19 Registros de Materiales

Fuente: Propia

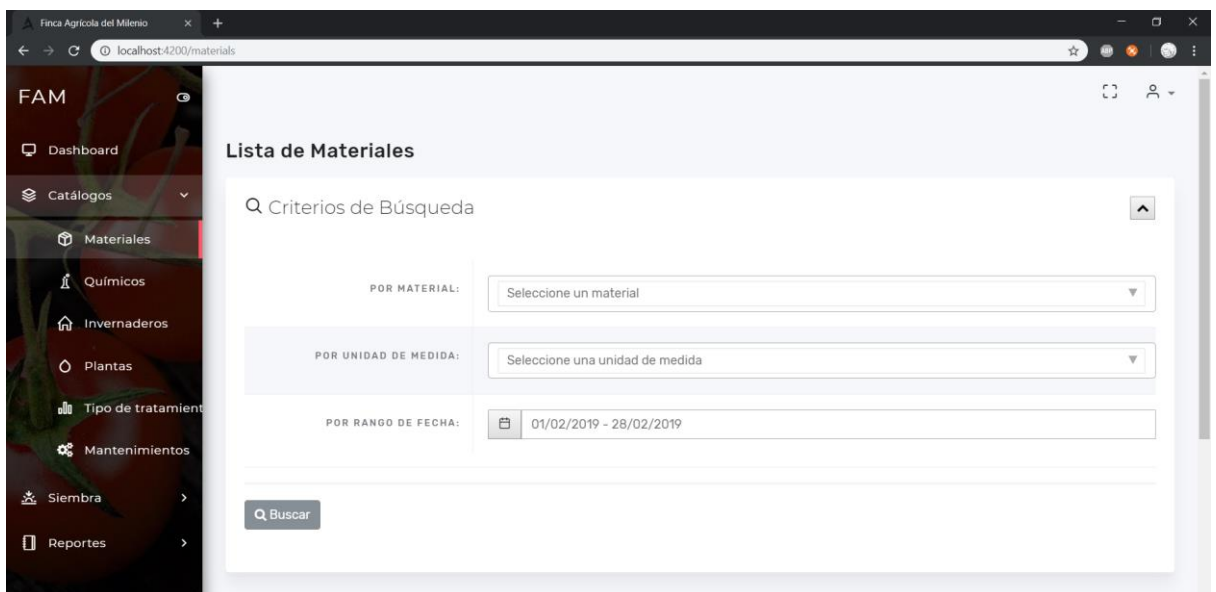


Figura 20 Criterio de Búsqueda registro de materiales

Fuente: Propia

Registro de materiales

INGRESO DE MATERIALES

Información General

FECHA DE COMPRA: 2019-02-02

NOMBRE: manguera

CANTIDAD: 25

UNIDAD DE MEDIDA: Metros

PRECIO DE COMPRA: -24 \$
Este campo es obligatorio y debe ser mayor a cero.

DESCRIPCIÓN DEL MATERIAL: tiene perforaciones cada 10 centímetros

COMENTARIO DEL MATERIAL: es utilizada para regar a las plantas

Figura 21 Formulario de registro de materiales

Fuente: Propia

Lista de Químicos

Criterios de Búsqueda

REGISTRO DE QUÍMICOS

+ Agregar Nuevo Exportar

Id	Acciones	Fecha compra	Nombre	Cantidad	Unidad de medida	Precio de compra	Descripción	Comentario
16		2019-01-31	neometin	2400	Gramos	24	fertilizante	fertilizante
15		2019-01-31	indicate	1900	Mililitros	14	fertilizante	fertilizante
14		2019-01-31	herbisec	5000	Gramos	23	fertilizante	fertilizante
11		2019-01-31	acidos humicos	1000	Mililitros	9	para neutralizar las sales de los suelos	para la ferti

14 total

Figura 22 Registros de químicos

Fuente: Propia

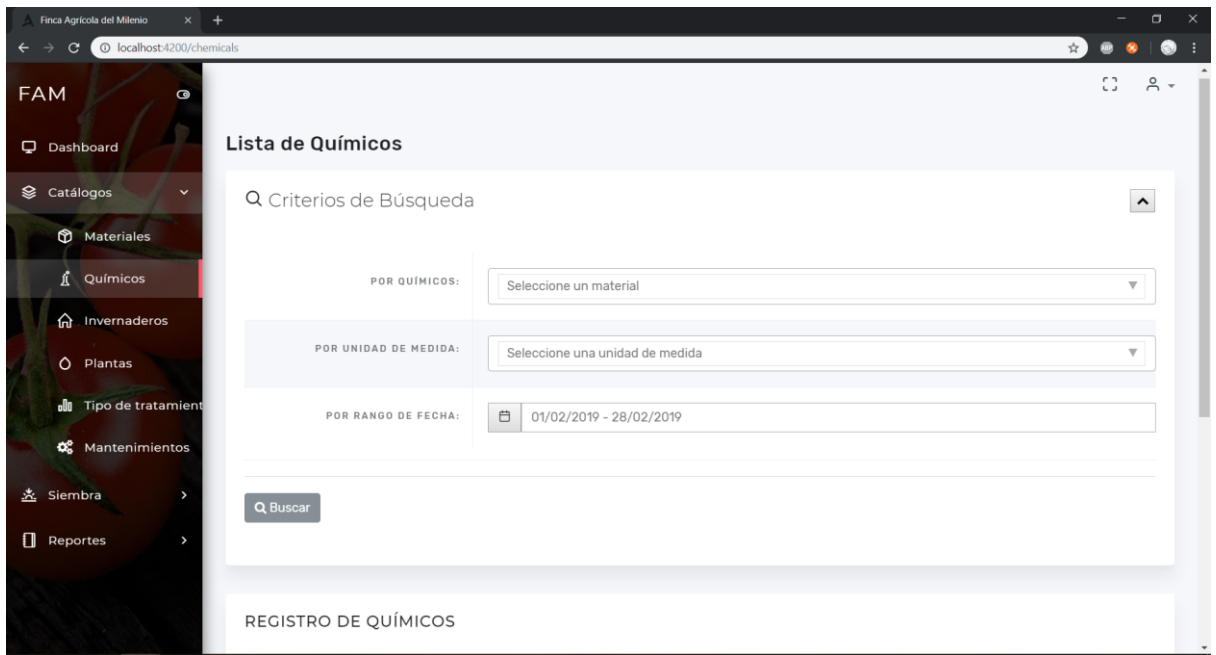


Figura 23 Criterios de Búsqueda registro de químicos

Fuente: Propia

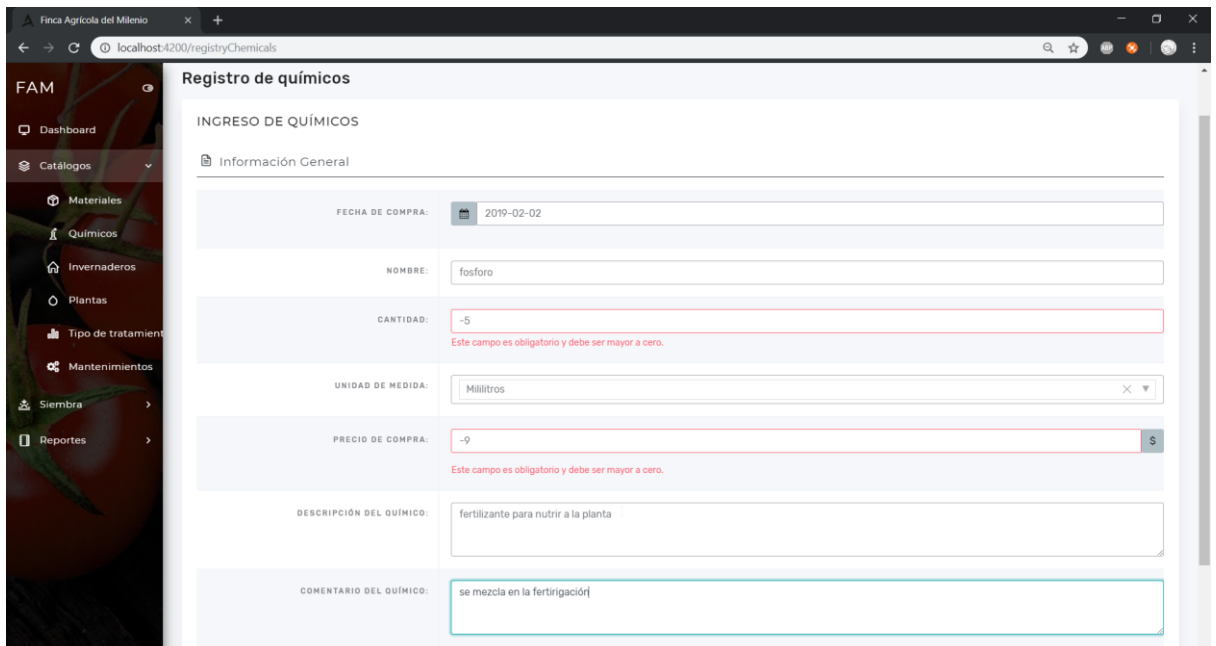


Figura 24 Formulario de registro de químicos

Fuente: Propia



Figura 25 Exportación de archivos, materiales y químicos

Fuente: Propia

c. Revisión retrospectiva

No se presentó ningún problema con las versiones de los componentes usados para la realización de este sprint, debido al previo análisis de la compatibilidad de estos con el entorno de desarrollo.

Se presentó un inconveniente con la ubicación de los listados tanto de materiales como de los químicos, ya que se presentaba a los dos listados dentro de una misma pantalla, lo cual para el dueño del proceso, no fue nada práctico, y por ende se tuvo que rehacer a las mismas.

Es vital que el product owner sepa expresarse de la manera correcta, por lo cual el equipo de desarrollo debe realizar preguntas abiertas al dueño del proceso, con el fin de que se exprese sus ideas.

2.3.4 Sprint 3 Gestión de Tratamientos y Mantenimientos

En este sprint se procede a desarrollar las funcionalidades del módulo de Tratamientos y Mantenimientos.

a. Planificación del sprint

Se la realizo con el Product Owner, Scrum Master y el equipo de desarrollo la fecha de 28/05/2018.

Tabla 12 Planificación del sprint 3

ID	Sprint 3
NOMBRE DE ITERACIÓN	Gestión de Tratamientos y Mantenimientos
FECHA INICIAL	28/05/2018
FECHA FINAL	22/06/2018

Fuente: Propia

De la reunión se determinó el sprint backlog, que se conforman por las historias de usuario que se desarrollarán en el Sprint 3.

Tabla 13 Sprint backlog del sprint 3

Sprint	S3 Gestión de Tratamientos y Mantenimientos	Inicio	28/052018
Responsable	Carlos Proaño	Fin	22/06/2018
Estado	TERMINADO	Duración	160
HISTORIA DE USUARIO	TAREAS	TIPO	
Registro de mantenimientos	Creación del modelo "Maintenance" de la tabla "tblmaintenancegreenhouse" en el model de springBoot	Desarrollo	
	Creación del repositorio para el modelo "Maintenance"	Desarrollo	
	Creación del servicio web rest "save" para el registro de mantenimientos	Desarrollo	
	Instalación de los servicios web	Desarrollo	
	Creación de la clase "maintenance.service" y agregar el método de "guardar" para que capture el servicio web	Desarrollo	
	Creación del módulo de Registro de Mantenimientos para el front-end	Desarrollo	
	Creación y agregamiento de las clases module.ts y routing.ts al módulo de registro de mantenimientos del front-end	Desarrollo	
	Creación del routing para la navegación dentro del sistema web	Desarrollo	
	Creación del formulario para el registro de mantenimientos en el front-end	Desarrollo	
	Validación de los campos del registro	Desarrollo	
	Creación del modelo "maintenance.model" en el front-end	Desarrollo	
	Asignación del modelo "maintenance.model" a cada campo del formulario	Desarrollo	
	Asignación de la clase "maintenance.service" al module.ts del registro de mantenimientos	Desarrollo	
Creación del método guardar	Desarrollo		

Actualización de mantenimientos	Creación del servicio web rest para la actualización del registro de mantenimientos	Desarrollo
	Instalación de los servicios web	Desarrollo
	Agregar el método de "editar" en la clase "maintenance.service" para que capture el servicio web	Desarrollo
	Creación del método para que los datos a editar se muestren en el registro de mantenimientos en el front-end	Desarrollo
	Asignación del modelo "maintenance.model" a cada campo del formulario	Desarrollo
	Creación del método editar	Desarrollo
Visualización de los registros de mantenimientos	Creación del servicio web rest para la visualización del listado mantenimientos	Desarrollo
	Instalación de los servicios web	Desarrollo
	Agregar el método de "findAll" en la clase "maintenance.service" para que capture el servicio web	Desarrollo
	Creación del módulo de Listado de Mantenimientos para el front-end	Desarrollo
	Creación y agregamiento de las clases module.ts y routing.ts al módulo de listado de mantenimientos del front-end	Desarrollo
	Creación del routing para la navegación dentro del sistema web	Desarrollo
	Creación del formulario para la visualización de los registros de mantenimientos en el front-end	Desarrollo
	Asignación de la clase "maintenance.service" al module.ts del registro de listado de mantenimientos	Desarrollo
	Creación del método "getMaintenances"	Desarrollo
	Importación del componente "ngx-data-table"	Desarrollo
	Mostrar los datos dentro del componente	Desarrollo
Filtros de búsqueda en el registros de mantenimientos	Creación del servicio web rest para la visualización del listado mantenimientos	Desarrollo
	Instalación de los servicios web	Desarrollo
	Agregar el método de "getMaintenanceByCriteria" en la clase "maintenance.service" para que capture el servicio web	Desarrollo
	Creación de los campos de búsqueda en el módulo de listado de mantenimientos	Desarrollo
	Creación del método para realizarla búsqueda en el front-end	Desarrollo
Registro de tratamientos	Creación del modelo "ChemicalRecipe" de la tabla "tblchemicalrecipe" en el model de springBoot	Desarrollo
	Creación del repositorio para el modelo "ChemicalRecipe"	Desarrollo
	Creación del servicio web rest "save" para el registro de tratamientos	Desarrollo
	Instalación de los servicios web	Desarrollo
	Creación de la clase "recipes.service" y agregar el método de "guardar" para que capture el servicio web	Desarrollo
	Creación del módulo de Registro de Tratamientos para el front-end	Desarrollo
	Creación y agregamiento de las clases module.ts y routing.ts al módulo de registro de tratamientos del front-end	Desarrollo
	Creación del routing para la navegación dentro del sistema web	Desarrollo
	Creación del formulario para el registro de tratamientos en el front-end	Desarrollo
	Validación de los campos del registro	Desarrollo
	Creación del modelo "recipes.model" en el front-end	Desarrollo

	Asignación del modelo "recipes.model" a cada campo del formulario	Desarrollo
	Asignación de la clase "recipes.service" al module.ts del registro de tratamientos	Desarrollo
	Creación del método guardar	Desarrollo
Actualización de tratamientos	Creación del servicio web rest para la actualización del registro de tratamientos	Desarrollo
	Instalación de los servicios web	Desarrollo
	Agregar el método de "editar" en la clase "recipes.service" para que capture el servicio web	Desarrollo
	Creación del método para que los datos a editar se muestren en el registro de tratamientos en el front-end	Desarrollo
	Asignación del modelo "recipes.model" a cada campo del formulario	Desarrollo
	Creación del método editar	Desarrollo
Visualización de los registros de tratamientos	Creación del servicio web rest para la visualización del listado tratamientos	Desarrollo
	Instalación de los servicios web	Desarrollo
	Agregar el método de "findAll" en la clase "recipes.service" para que capture el servicio web	Desarrollo
	Creación del módulo de Listado de Tratamientos para el front-end	Desarrollo
	Creación y agregamiento de las clases module.ts y routing.ts al módulo de listado de tratamientos del front-end	Desarrollo
	Creación del routing para la navegación dentro del sistema web	Desarrollo
	Creación del formulario para la visualización de los registros de tratamientos en el front-end	Desarrollo
	Asignación de la clase "recipes.service" al module.ts del registro de listado de tratamientos	Desarrollo
	Creación del método "getRecipes"	Desarrollo
	Importación del componente "ngx-data-table"	Desarrollo
	Mostrar los datos dentro del componente	Desarrollo
Filtros de búsqueda en el registros de tratamientos	Creación del servicio web rest para la visualización del listado tratamientos	Desarrollo
	Instalación de los servicios web	Desarrollo
	Agregar el método de "getRecipesByCriteria" en la clase "recipes.service" para que capture el servicio web	Desarrollo
	Creación de los campos de búsqueda en el módulo de listado de tratamientos	Desarrollo
	Creación del método para realizarla búsqueda en el front-end	Desarrollo
Exportación de datos	Instalación de componentes para la exportación de datos	Desarrollo
	Creación de los métodos	Desarrollo

Fuente: Propia

b. Revisión del sprint

Ya realizadas las tareas del sprint y ejecutado las pruebas pertinentes, se entrega al módulo como terminado, a continuación, se muestran las pantallas del módulo finalizado.



Figura 26 SideBar para Tipo de tratamientos y Mantenimientos

Fuente: Propia

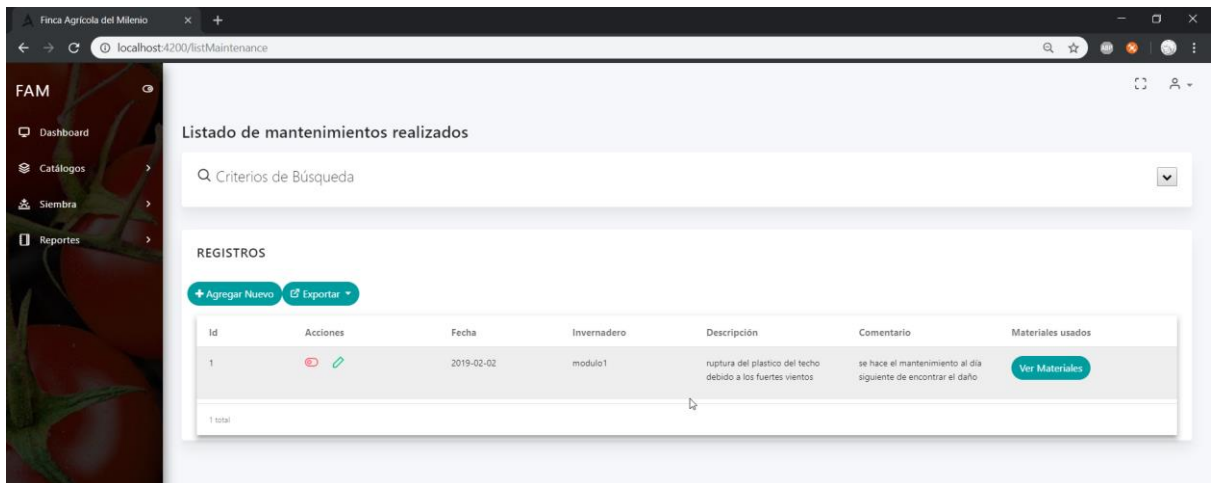


Figura 27 Registros de mantenimientos

Fuente: Propia

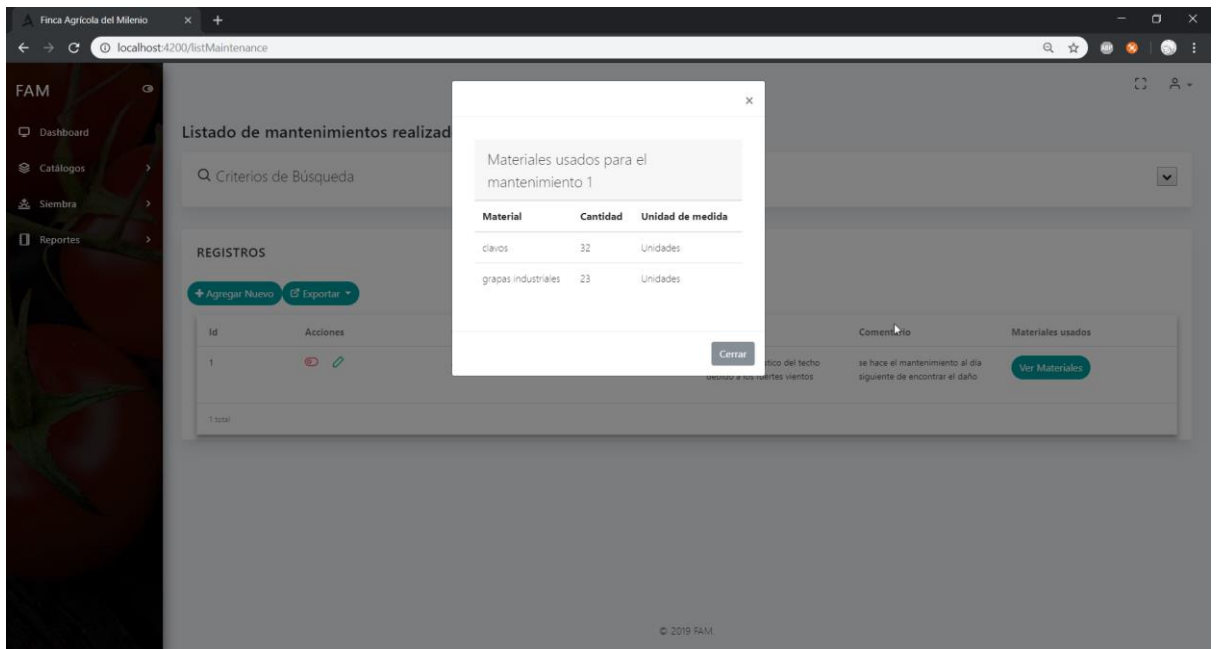


Figura 28 Materiales usados

Fuente: Propia

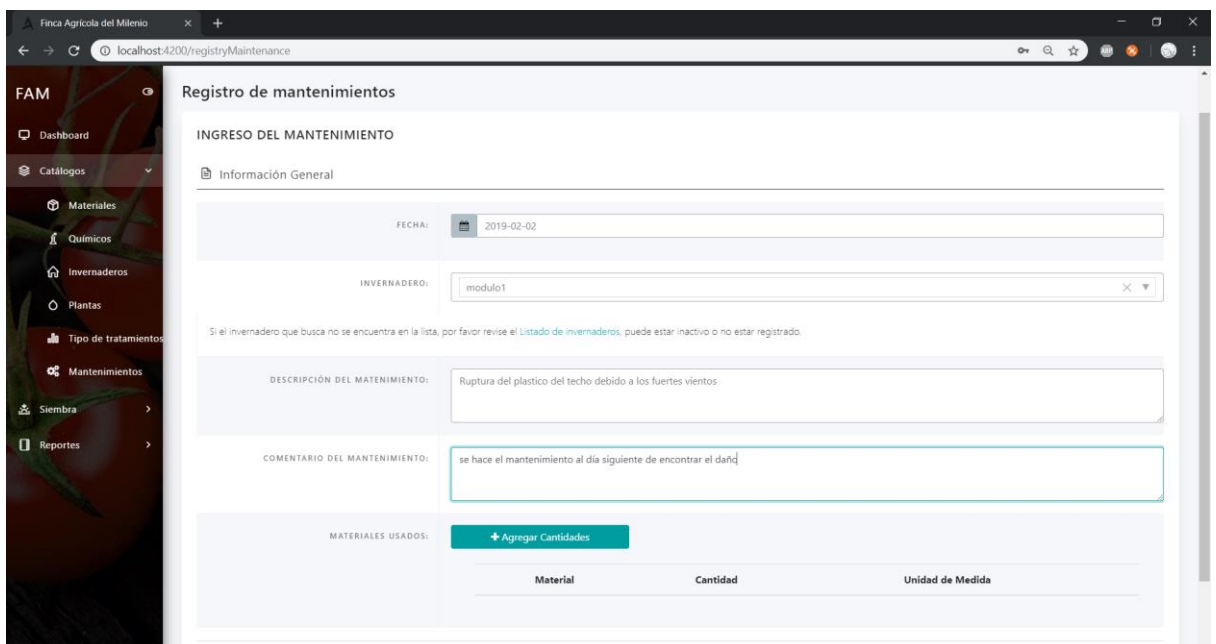


Figura 29 Formulario de registro de mantenimientos

Fuente: Propia

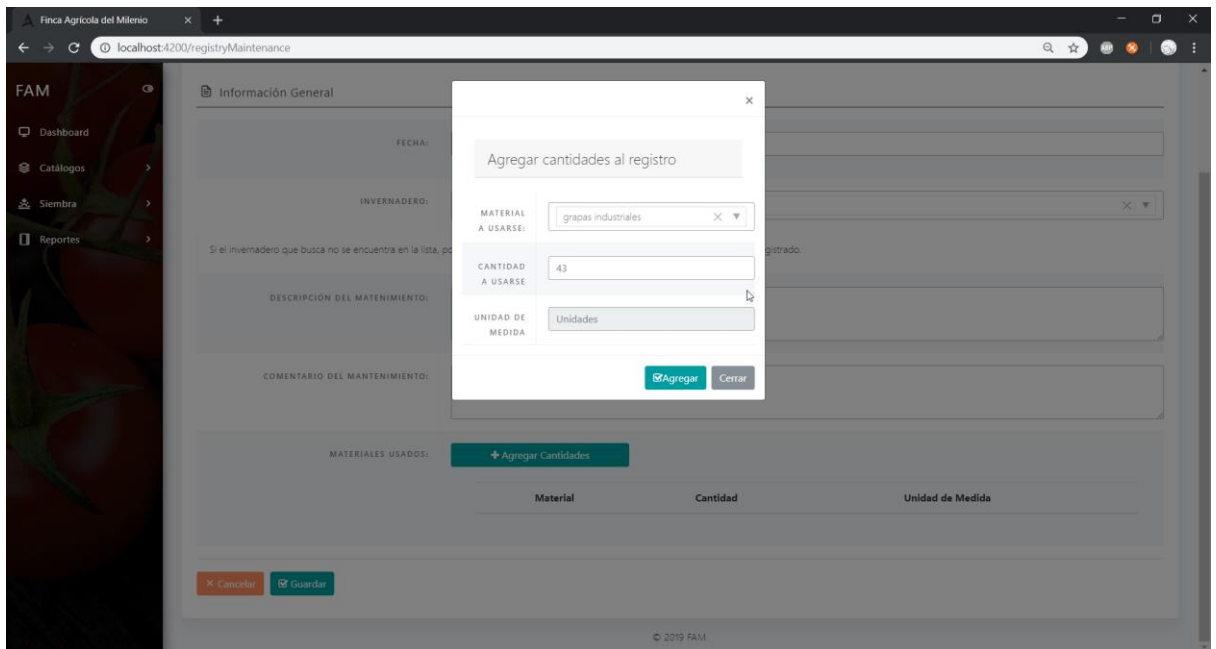


Figura 30 Popup para agregar materiales

Fuente: Propia

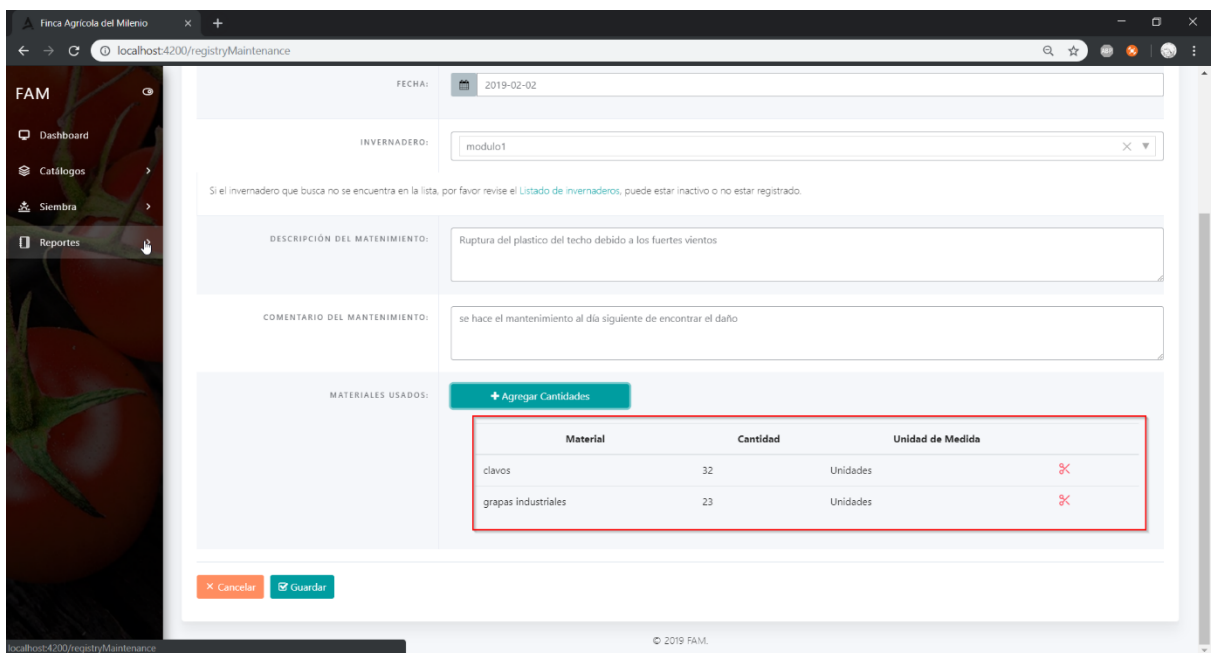


Figura 31 Listado de materiales a registrarse

Fuente: Propia

Finca Agrícola del Milenio

localhost:4200/listTreatment

FAM

Dashboard

Catálogos

Materiales

Químicos

Invernaderos

Plantas

Tipo de tratamientos

Mantenimientos

Siembra

Reportes

Listado de tratamientos, registrados por el usuario

Criterios de Búsqueda

REGISTROS

+ Agregar Nuevo Exportar

Id	Acciones	Fecha	Nombre	Caso	Descripción	Comentario	Cantidades Aplicadas
2		2019-01-31	segunda fumigación preventiva/curativa	Fumigación Preventiva	indicate 10ml/20lt de agua corrector de ph. fitoraz 50 gr/20lt de agua fungicida, neometin 10ml/20lt de agua insecticida abono foliar 50gr/ 20lt de agua, herbisecc 10 gr/20lt de agua, fungloraz 16 gr/20lt de agua: fitonormal plus 10ml/20lt de agua floracion y engrose del fruto	segunda fumigacion preventiva	Ver Cantidades
1		2019-01-31	fumigacion preventiva/curativa	Fumigación Preventiva	dosificacion inicial: indicate 10ml/20lt de agua corrector de ph. fitoraz 50 gr/20lt de agua fungicida, neometin 10ml/20lt de agua insecticida abono foliar 50gr/ 20lt de agua todo esto se aplica en 200 lts de agua	fumigacion preventiva semana	Ver Cantidades

Figura 32 Registros de tratamientos

Fuente: Propia

Finca Agrícola del Milenio

localhost:4200/listTreatment

FAM

Dashboard

Catálogos

Materiales

Químicos

Invernaderos

Plantas

Tipo de tratamientos

Mantenimientos

Siembra

Reportes

Listado de tratamientos, registrados por el usuario

Criterios de Búsqueda

POR CASO: Seleccione uno

POR NOMBRE DE TRATAMIENTO: Seleccione uno

POR RANGO DE FECHA: 01/02/2019 - 28/02/2019

Buscar

REGISTROS

+ Agregar Nuevo Exportar

Id	Acciones	Fecha	Nombre	Caso	Descripción	Comentario	Cantidades Aplicadas
2		2019-01-31	segunda fumigación preventiva/curativa	Fumigación Preventiva	indicate 10ml/20lt de agua corrector de ph. fitoraz 50 gr/20lt de agua fungicida	segunda fumigacion preventiva	Ver Cantidades

Figura 33 Criterios de Búsqueda, registro de tratamientos

Fuente: Propia

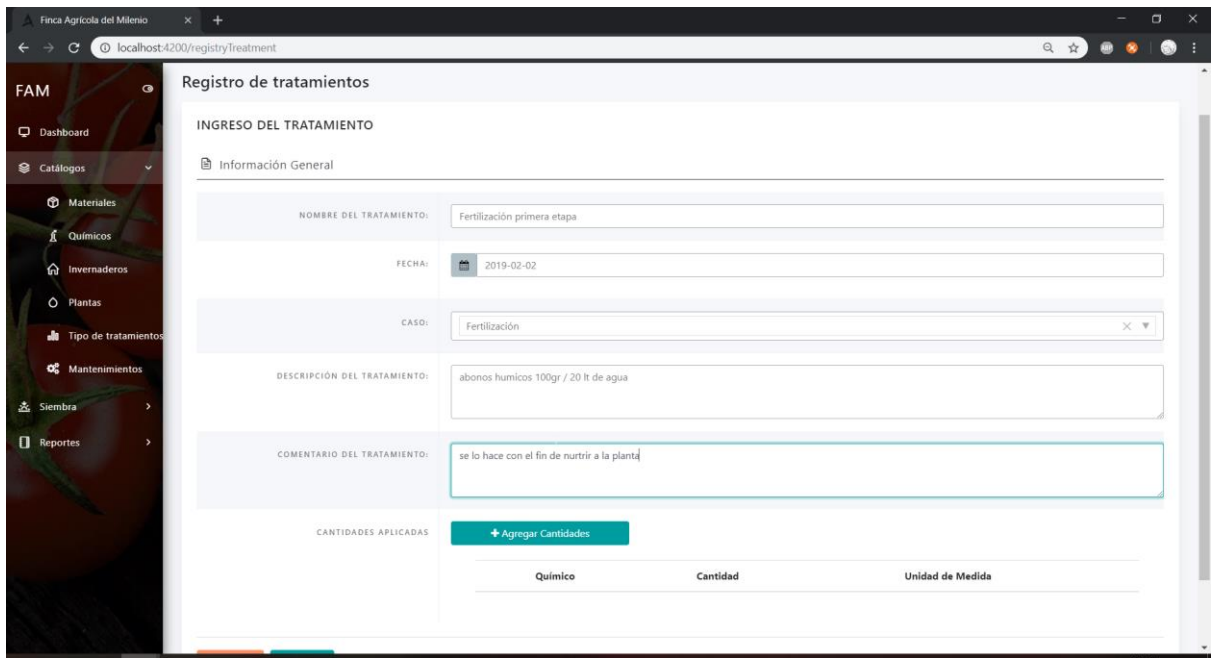


Figura 34 Formulario de registro de tratamientos

Fuente: Propia

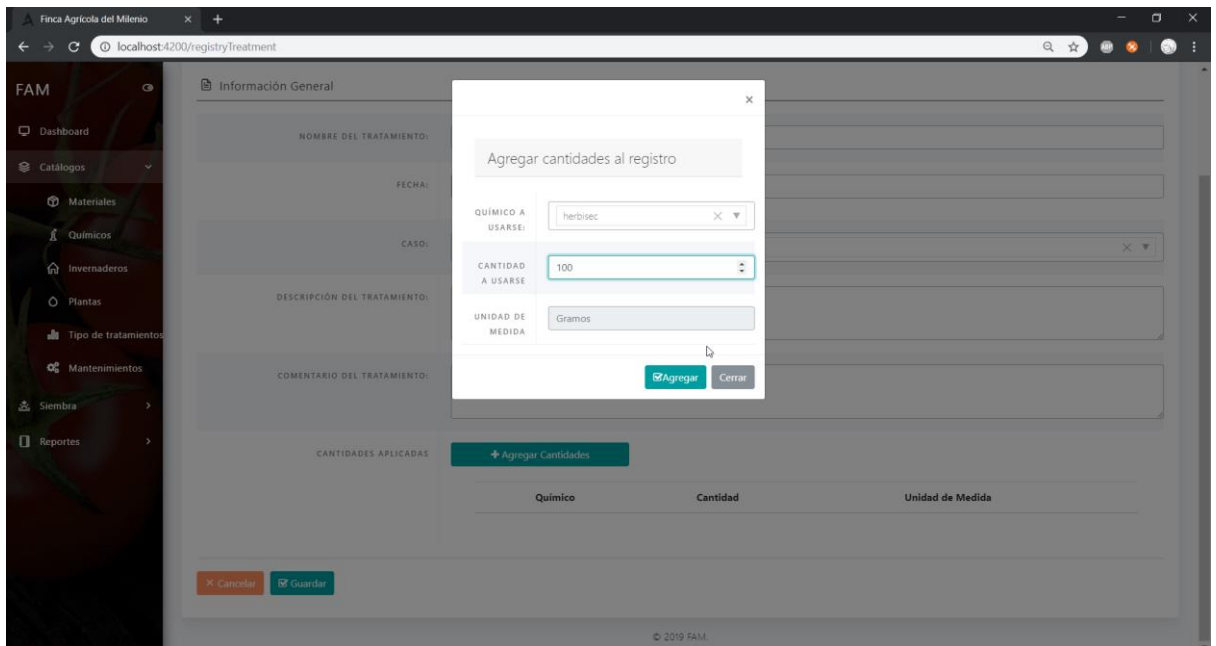


Figura 35 PopUp para agregar químicos

Fuente: Propia

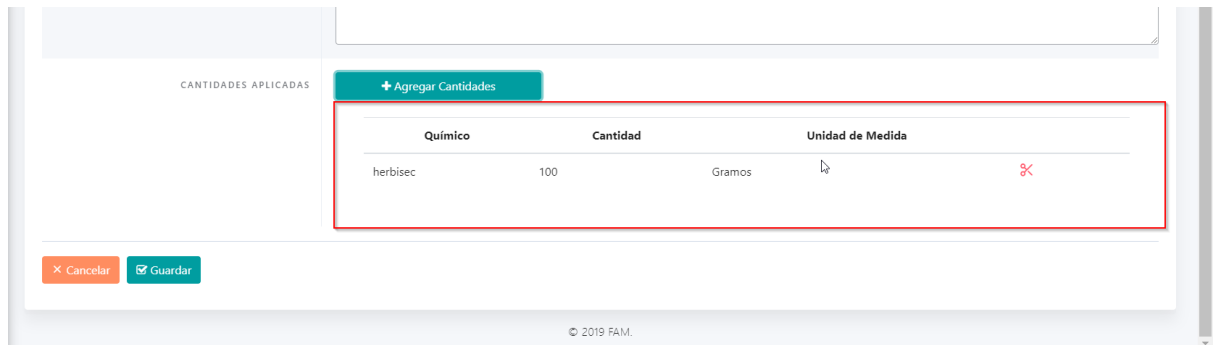


Figura 36 Listado de químicos a usarse

Fuente: Propia

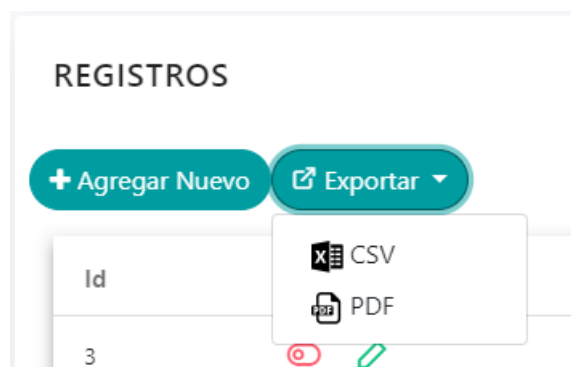


Figura 37 Exportación de archivos, tratamientos y mantenimientos

Fuente: Propia

c. Revisión retrospectiva

No se tuvo ningún problema al realizar las reuniones con el product owner, ya que se vio necesario hacer una, para explicar el proceso de registro de nuevos datos.

El desarrollo se vio estancado debido a que en los formularios de registro se debe manejar un modelo de factura, en el cual, los materiales y químicos deben ser agregados uno por uno. Por lo cual hubo un poco de inconvenientes al exponer este procedimiento al dueño del proceso, debido a que no entendía el concepto de “modelo de factura”.

Es necesario expresar los términos usados por el equipo de desarrollo, en palabras que el usuario pueda entender, con el fin de que haya una comunicación mutua y no se presenten inconvenientes por falta de entendimiento.

2.3.5 Sprint 4 Gestión de Siembras y Tratamientos Aplicados

En este sprint se procede a desarrollar las funcionalidades del módulo de Siembras y Tratamientos Aplicados.

a. Planificación del sprint

Se la realizo con el Product Owner, Scrum Master y el equipo de desarrollo la fecha de 09/07/2018.

Tabla 14 Planificación del sprint 4

ID	Sprint 4
NOMBRE DE ITERACIÓN	Gestión de Siembras y Tratamientos Aplicados
FECHA INICIAL	09/07/2018
FECHA FINAL	03/08/2018

Fuente: Propia

De la reunión se determinó el sprint backlog, que se conforman por las historias de usuario que se desarrollarán en el Sprint 4.

Tabla 15 Sprint backlog del sprint 4

Sprint	S4 Gestión de Siembras y Tratamientos Aplicados	Inicio	09/07/2018
Responsable	Carlos Proaño	Fin	03/08/2018
Estado	TERMINADO	Duración	160
HISTORIA DE USUARIO	TAREAS	TIPO	
Registro de siembras	Creación del modelo "Sowing" de la tabla "tblsowing" en el model de springBoot	Desarrollo	
	Creación del repositorio para el modelo "Sowing"	Desarrollo	
	Creación del servicio web rest "save" para el registro de siembras	Desarrollo	
	Instalación de los servicios web	Desarrollo	
	Creación de la clase "sowing.service" y agregar el método de "guardar" para que capture el servicio web	Desarrollo	
	Creación del módulo de Registro de siembras para el front-end	Desarrollo	
	Creación y agregamiento de las clases module.ts y routing.ts al módulo de registro de siembras del front-end	Desarrollo	
	Creación del routing para la navegación dentro del sistema web	Desarrollo	
	Creación del formulario para el registro de siembras en el front-end	Desarrollo	
	Validación de los campos del registro	Desarrollo	
	Creación del modelo "sowing.model" en el front-end	Desarrollo	
	Asignación del modelo "sowing.model" a cada campo del formulario	Desarrollo	
	Asignación de la clase "sowing.service" al module.ts del registro de siembras	Desarrollo	
Creación del método guardar	Desarrollo		

Actualización de siembras	Creación del servicio web rest para la actualización del registro de siembras	Desarrollo
	Instalación de los servicios web	Desarrollo
	Agregar el método de "editar" en la clase "sowing.service" para que capture el servicio web	Desarrollo
	Creación del método para que los datos a editar se muestren en el registro de siembras en el front-end	Desarrollo
	Asignación del modelo "sowing.model" a cada campo del formulario	Desarrollo
	Creación del método editar	Desarrollo
Visualización de los registros de siembras	Creación del servicio web rest para la visualización del listado siembras	Desarrollo
	Instalación de los servicios web	Desarrollo
	Agregar el método de "findAll" en la clase "sowing.service" para que capture el servicio web	Desarrollo
	Creación del módulo de Listado de siembras para el front-end	Desarrollo
	Creación y agregamiento de las clases module.ts y routing.ts al módulo de listado de siembras del front-end	Desarrollo
	Creación del routing para la navegación dentro del sistema web	Desarrollo
	Creación del formulario para la visualización de los registros de siembras en el front-end	Desarrollo
	Asignación de la clase "sowing.service" al module.ts del registro de listado de siembras	Desarrollo
	Creación del método "getsowings"	Desarrollo
	Importación del componente "ngx-data-table"	Desarrollo
	Mostrar los datos dentro del componente	Desarrollo
Filtros de búsqueda en el registros de siembras	Creación del servicio web rest para la visualización del listado siembras	Desarrollo
	Instalación de los servicios web	Desarrollo
	Agregar el método de "getSowingByCriteria" en la clase "sowing.service" para que capture el servicio web	Desarrollo
	Creación de los campos de búsqueda en el módulo de listado de siembras	Desarrollo
	Creación del método para realizarla búsqueda en el front-end	Desarrollo
Registro de tratamientos aplicados	Creación del modelo "Treatments" de la tabla "tblTreatments" en el model de springBoot	Desarrollo
	Creación del repositorio para el modelo "Treatments"	Desarrollo
	Creación del servicio web rest "save" para el registro de tratamientos aplicados	Desarrollo
	Instalación de los servicios web	Desarrollo
	Creación de la clase "treatments.service" y agregar el método de "guardar" para que capture el servicio web	Desarrollo
	Creación del módulo de Registro de tratamientos aplicados para el front-end	Desarrollo
	Creación y agregamiento de las clases module.ts y routing.ts al módulo de registro de tratamientos aplicados del front-end	Desarrollo
	Creación del routing para la navegación dentro del sistema web	Desarrollo
	Creación del formulario para el registro de tratamientos aplicados en el front-end	Desarrollo
	Validación de los campos del registro	Desarrollo
	Creación del modelo "treatments.model" en el front-end	Desarrollo
	Asignación del modelo "treatments.model" a cada campo del formulario	Desarrollo

	Asignación de la clase "treatments.service" al module.ts del registro de tratamientos aplicados	Desarrollo
	Creación del método guardar	Desarrollo
Actualización de tratamientos aplicados	Creación del servicio web rest para la actualización del registro de tratamientos aplicados	Desarrollo
	Instalación de los servicios web	Desarrollo
	Agregar el método de "editar" en la clase "treatments.service" para que capture el servicio web	Desarrollo
	Creación del método para que los datos a editar se muestren en el registro de tratamientos aplicados en el front-end	Desarrollo
	Asignación del modelo "treatments.model" a cada campo del formulario	Desarrollo
	Creación del método editar	Desarrollo
Visualización de los registros de tratamientos aplicados	Creación del servicio web rest para la visualización del listado tratamientos aplicados	Desarrollo
	Instalación de los servicios web	Desarrollo
	Agregar el método de "findAll" en la clase "treatments.service" para que capture el servicio web	Desarrollo
	Creación del módulo de Listado de tratamientos aplicados para el front-end	Desarrollo
	Creación y agregamiento de las clases module.ts y routing.ts al módulo de listado de tratamientos aplicados del front-end	Desarrollo
	Creación del routing para la navegación dentro del sistema web	Desarrollo
	Creación del formulario para la visualización de los registros de tratamientos aplicados en el front-end	Desarrollo
	Asignación de la clase "treatments.service" al module.ts del registro de listado de tratamientos aplicados	Desarrollo
	Creación del método "gettreatments"	Desarrollo
	Importación del componente "ngx-data-table"	Desarrollo
	Mostrar los datos dentro del componente	Desarrollo
Filtros de búsqueda en el registros de tratamientos aplicados	Creación del servicio web rest para la visualización del listado tratamientos aplicados	Desarrollo
	Instalación de los servicios web	Desarrollo
	Agregar el método de "gettreatmentsByCriteria" en la clase "treatments.service" para que capture el servicio web	Desarrollo
	Creación de los campos de búsqueda en el módulo de listado de tratamientos aplicados	Desarrollo
	Creación del método para realizarla búsqueda en el front-end	Desarrollo
Exportación de datos	Instalación de componentes para la exportación de datos	Desarrollo
	Creación de los métodos	Desarrollo

Fuente: Propia

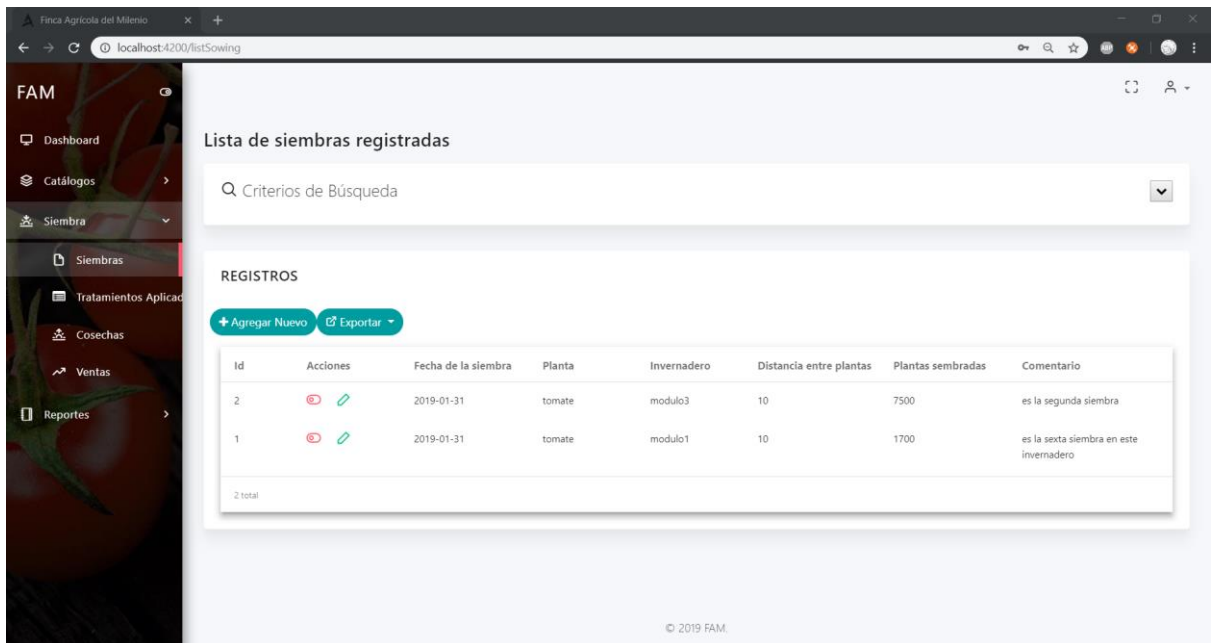
b. Revisión del sprint

Ya realizadas las tareas del sprint y ejecutado las pruebas pertinentes, se entrega al módulo como terminado, a continuación, se muestran las pantallas del módulo finalizado.



Figura 38 SideBar para Siembras y Tratamientos aplicados

Fuente: Propia







Lista de siembras registradas

Criterios de Búsqueda

REGISTROS

+ Agregar Nuevo Exportar

Id	Acciones	Fecha de la siembra	Planta	Invernadero	Distancia entre plantas	Plantas sembradas	Comentario
2	 	2019-01-31	tomate	modulo3	10	7500	es la segunda siembra
1	 	2019-01-31	tomate	modulo1	10	1700	es la sexta siembra en este invernadero

2 total

© 2019 FAM.

Figura 39 Registros de siembras

Fuente: Propia

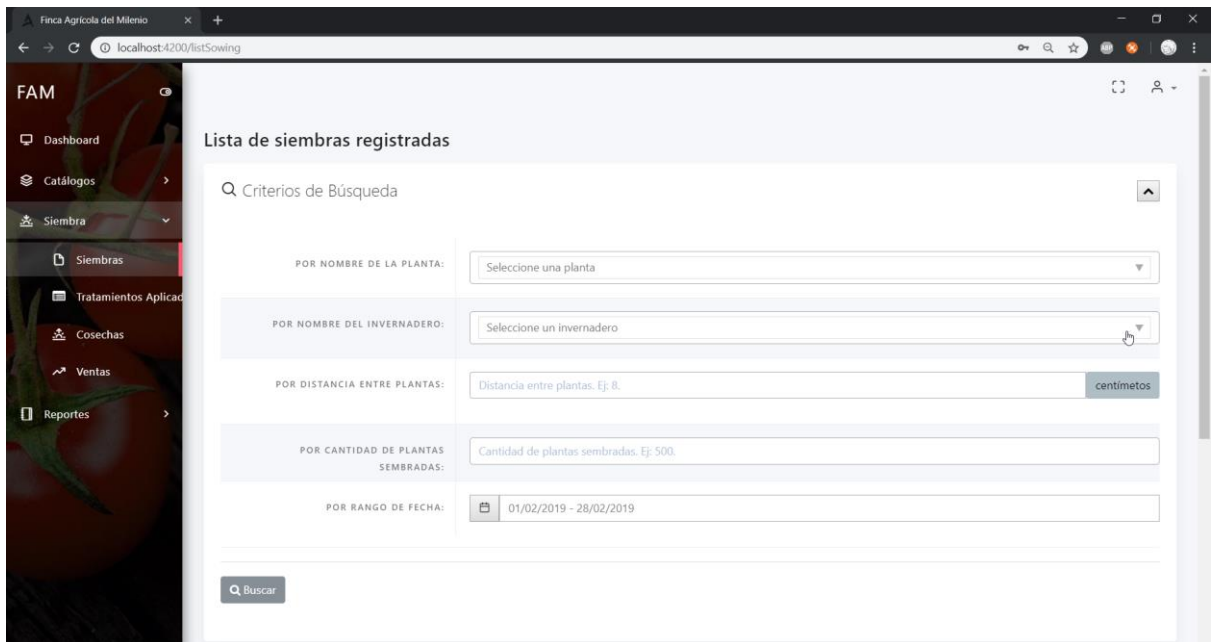


Figura 40 Criterios de Búsqueda, registro de siembras

Fuente: Propia

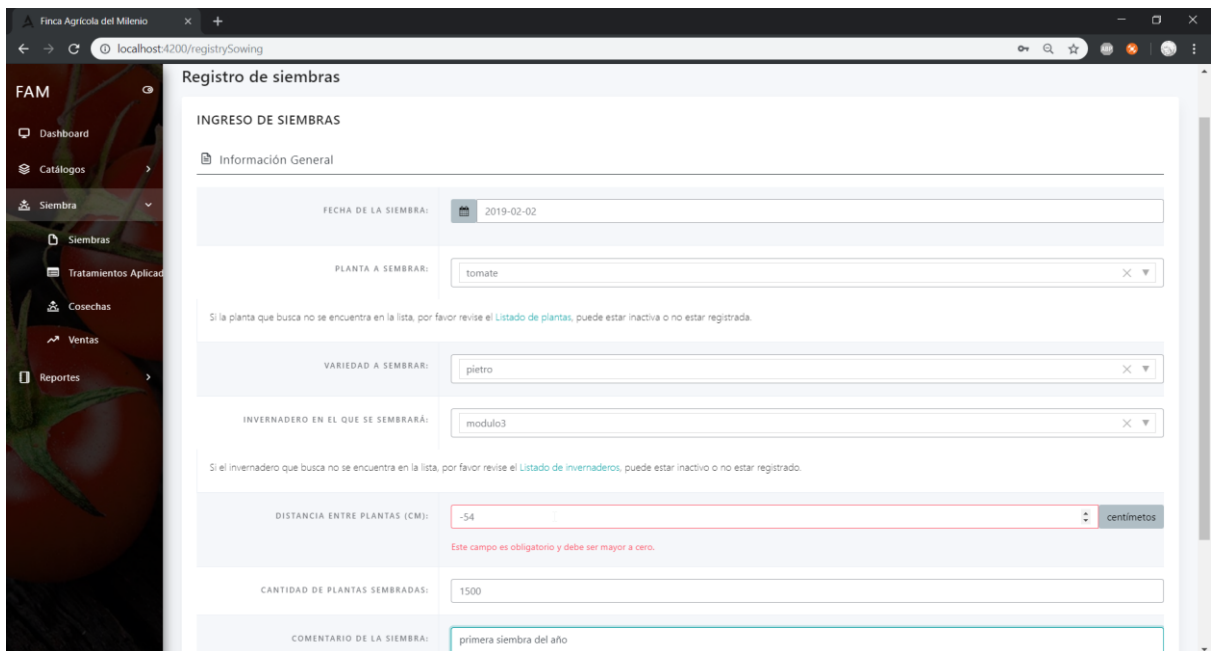


Figura 41 Formulario de registro de siembras

Fuente: Propia

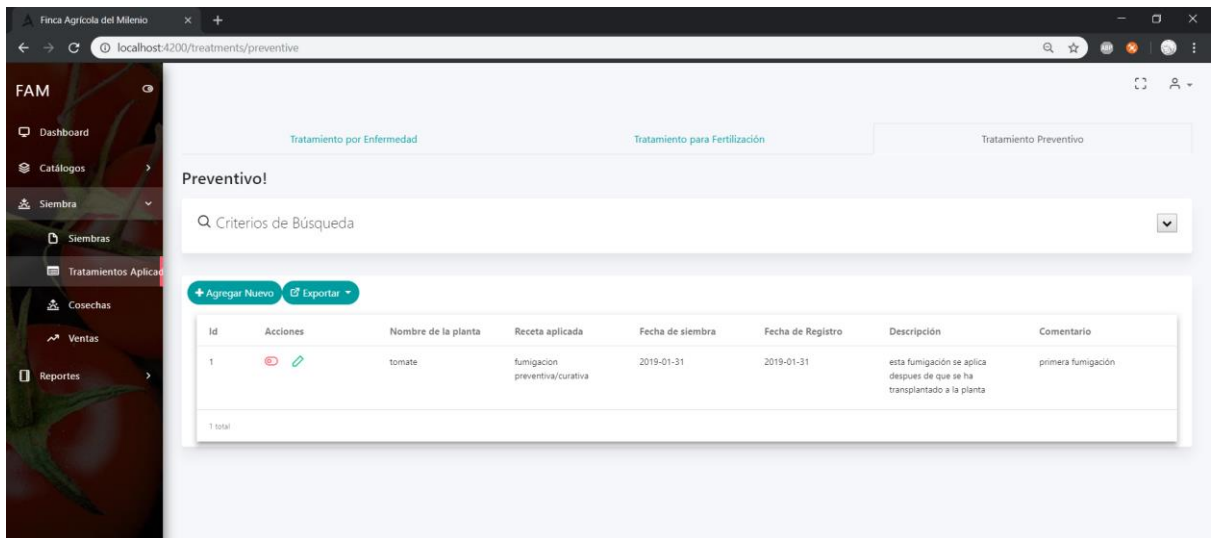


Figura 42 Registros de Tratamientos aplicados

Fuente: Propia

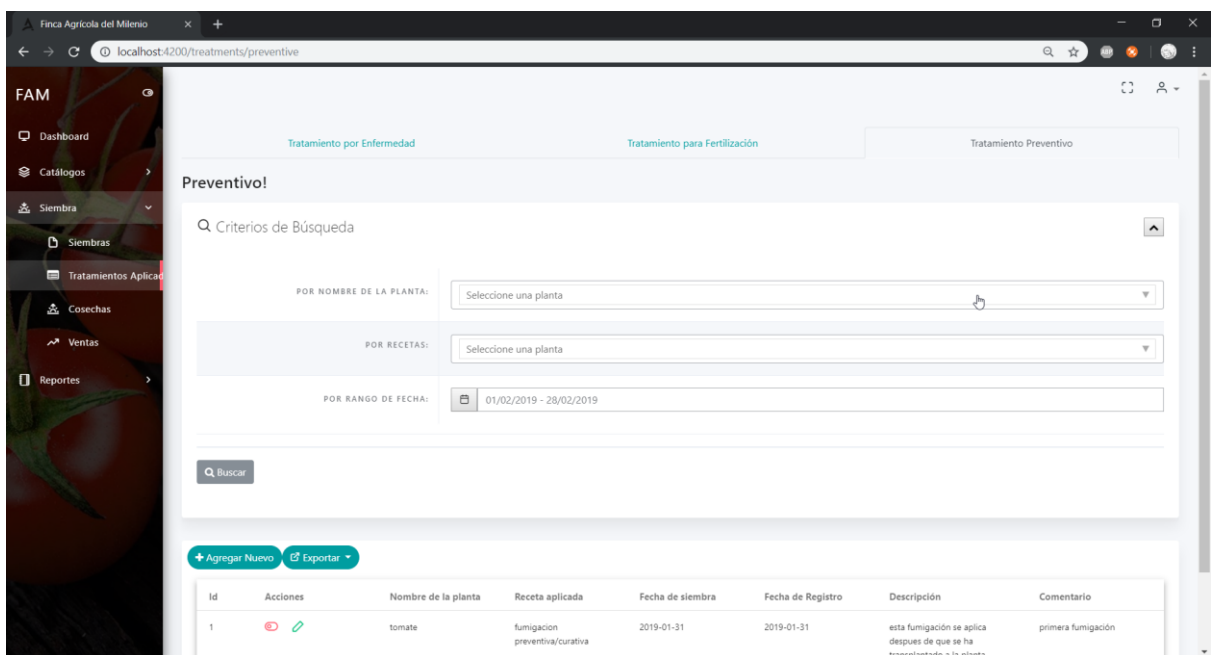


Figura 43 Criterios de Búsqueda, registro de tratamientos aplicados

Fuente: Propia

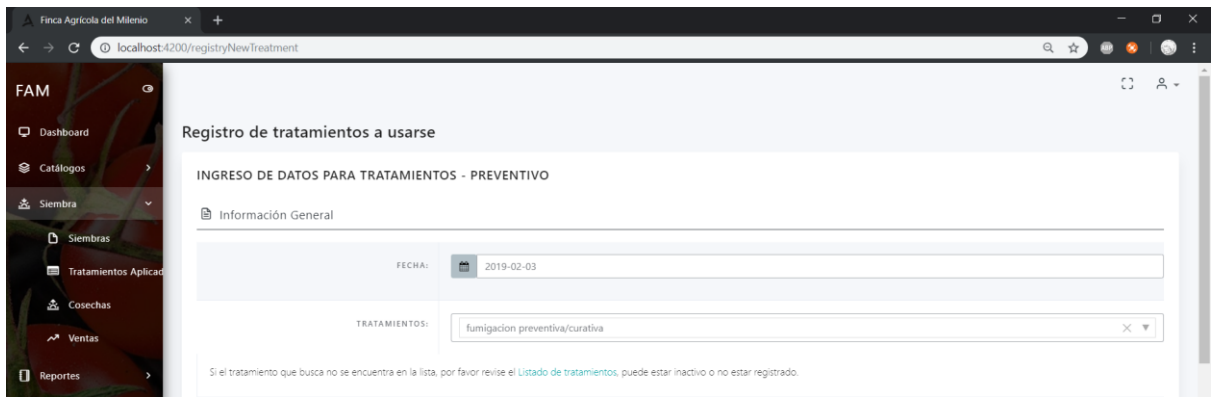


Figura 44 Formulario de registro, tratamientos

Fuente: Propia

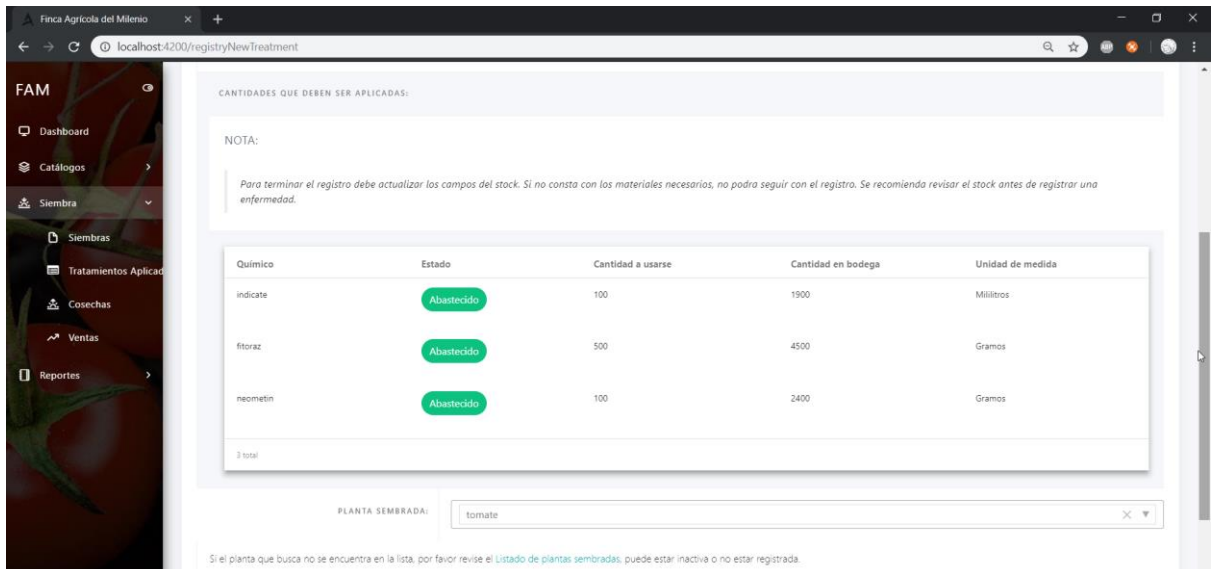


Figura 45 Formulario de registro, tratamientos

Fuente: Propia

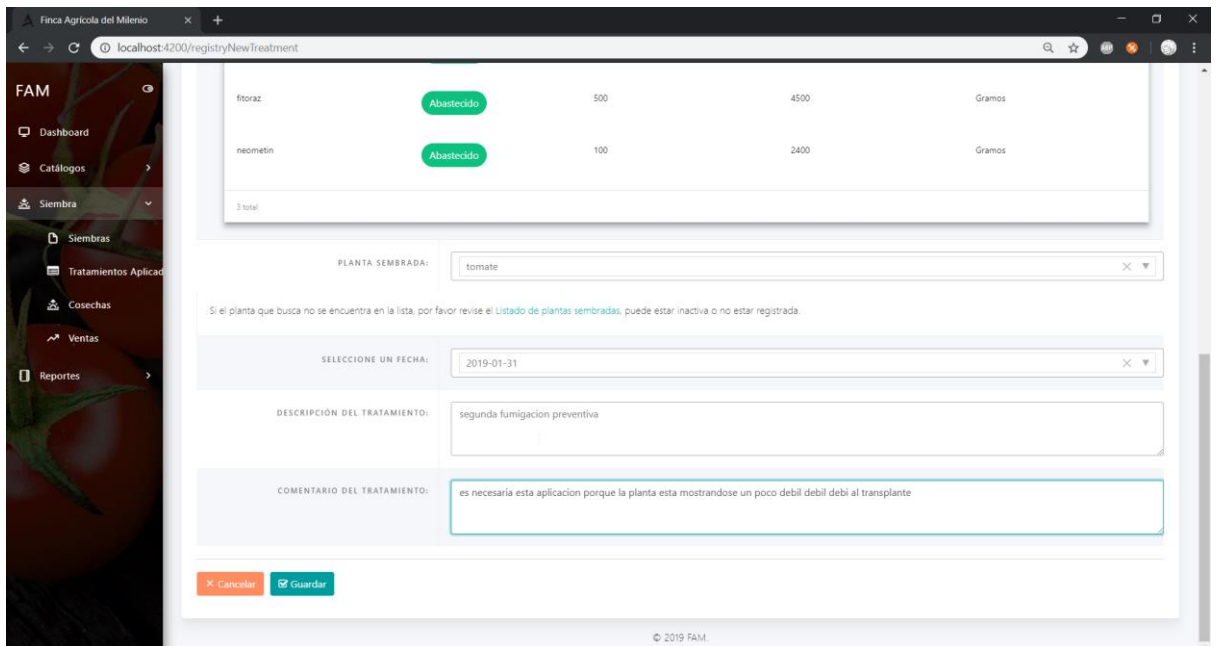


Figura 46 Formulario de registro, tratamientos

Fuente: Propia

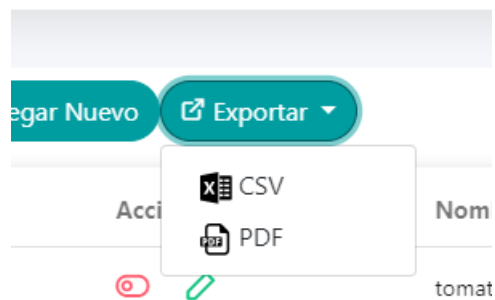


Figura 47 Exportación de archivos, siembras y tratamientos aplicados

Fuente: Propia

c. Revisión retrospectiva

La comunicación del equipo de desarrollo con el usuario final es bilateral, debido a esto se pueden expresar mejor las ideas y contemplar mejor los escenarios de desarrollo.

Se produjo un problema al momento de registrar los tratamientos aplicados, ya que en este proceso se debe manipular el stock de los químicos.

Es necesario trabajar conjuntamente con el usuario final ya que con su ayuda se agiliza ejecución del sprint, aportando ideas para el desarrollo.

2.3.6 Sprint 5 Gestión de Cosechas y Ventas

En este sprint se procede a desarrollar las funcionalidades del módulo de Cosechas y Ventas.

a. Planificación del sprint

Se la realizo con el Product Owner, Scrum Master y el equipo de desarrollo la fecha de 13/08/2018.

Tabla 16 Planificación del sprint 5

ID	Sprint 5
NOMBRE DE ITERACIÓN	Gestión de Cosechas y Ventas
FECHA INICIAL	13/08/2018
FECHA FINAL	07/09-/2018

Fuente: Propia

De la reunión se determinó el sprint backlog, que se conforman por las historias de usuario que se desarrollarán en el Sprint 5

Tabla 17 Sprint backlog del sprint 5

Sprint	S5 Gestión de Cosechas y Ventas	Inicio	13/08/2018
Responsable	Carlos Proaño	Fin	07/09/2018
Estado	TERMINADO	Duración	160
HISTORIA DE USUARIO	TAREAS	TIPO	
Registro de cosechas	Creación del modelo "Picking" de la tabla "tblpicking" en el model de springBoot	Desarrollo	
	Creación del repositorio para el modelo "Sowing"	Desarrollo	
	Creación del servicio web rest "save" para el registro de cosechas	Desarrollo	
	Instalación de los servicios web	Desarrollo	
	Creación de la clase "picking.service" y agregar el método de "guardar" para que capture el servicio web	Desarrollo	
	Creación del módulo de Registro de cosechas para el front-end	Desarrollo	
	Creación y agregamiento de las clases module.ts y rounting.ts al módulo de registro de cosechas del front-end	Desarrollo	
	Creación del routing para la navegación dentro del sistema web	Desarrollo	
	Creación del formulario para el registro de cosechas en el front-end	Desarrollo	
	Validación de los campos del registro	Desarrollo	
	Creación del modelo "picking.model" en el front-end	Desarrollo	
	Asignación del modelo "picking.model" a cada campo del formulario	Desarrollo	
	Asignación de la clase "picking.service" al module.ts del registro de cosechas	Desarrollo	
	Creación del método guardar	Desarrollo	

Actualización de cosechas	Creación del servicio web rest para la actualización del registro de cosechas	Desarrollo
	Instalación de los servicios web	Desarrollo
	Agregar el método de "editar" en la clase "picking.service" para que capture el servicio web	Desarrollo
	Creación del método para que los datos a editar se muestren en el registro de cosechas en el front-end	Desarrollo
	Asignación del modelo "picking.model" a cada campo del formulario	Desarrollo
	Creación del método editar	Desarrollo
Visualización de los registros de cosechas	Creación del servicio web rest para la visualización del listado cosechas	Desarrollo
	Instalación de los servicios web	Desarrollo
	Agregar el método de "findAll" en la clase "picking.service" para que capture el servicio web	Desarrollo
	Creación del módulo de Listado de cosechas para el front-end	Desarrollo
	Creación y agregamiento de las clases module.ts y routing.ts al módulo de listado de cosechas del front-end	Desarrollo
	Creación del routing para la navegación dentro del sistema web	Desarrollo
	Creación del formulario para la visualización de los registros de cosechas en el front-end	Desarrollo
	Asignación de la clase "picking.service" al module.ts del registro de listado de cosechas	Desarrollo
	Creación del método "getpickings"	Desarrollo
	Importación del componente "ngx-data-table"	Desarrollo
	Mostrar los datos dentro del componente	Desarrollo
Filtros de búsqueda en el registros de cosechas	Creación del servicio web rest para la visualización del listado cosechas	Desarrollo
	Instalación de los servicios web	Desarrollo
	Agregar el método de "getPickingByCriteria" en la clase "picking.service" para que capture el servicio web	Desarrollo
	Creación de los campos de búsqueda en el módulo de listado de cosechas	Desarrollo
	Creación del método para realizarla búsqueda en el front-end	Desarrollo
Registro de ventas	Creación del modelo "sales" de la tabla "tblsales" en el model de springBoot	Desarrollo
	Creación del repositorio para el modelo "sales"	Desarrollo
	Creación del servicio web rest "save" para el registro de ventas	Desarrollo
	Instalación de los servicios web	Desarrollo
	Creación de la clase "sale.service" y agregar el método de "guardar" para que capture el servicio web	Desarrollo
	Creación del módulo de Registro de ventas para el front-end	Desarrollo
	Creación y agregamiento de las clases module.ts y routing.ts al módulo de registro de ventas del front-end	Desarrollo
	Creación del routing para la navegación dentro del sistema web	Desarrollo
	Creación del formulario para el registro de ventas en el front-end	Desarrollo
	Validación de los campos del registro	Desarrollo
	Creación del modelo "sale.model" en el front-end	Desarrollo
	Asignación del modelo "sale.model" a cada campo del formulario	Desarrollo
	Asignación de la clase "sale.service" al module.ts del registro de ventas	Desarrollo
Creación del método guardar	Desarrollo	

Actualización de ventas	Creación del servicio web rest para la actualización del registro de ventas	Desarrollo
	Instalación de los servicios web	Desarrollo
	Agregar el método de "editar" en la clase "sale.service" para que capture el servicio web	Desarrollo
	Creación del método para que los datos a editar se muestren en el registro de ventas en el front-end	Desarrollo
	Asignación del modelo "sale.model" a cada campo del formulario	Desarrollo
	Creación del método editar	Desarrollo
Visualización de los registros de ventas	Creación del servicio web rest para la visualización del listado ventas	Desarrollo
	Instalación de los servicios web	Desarrollo
	Agregar el método de "findAll" en la clase "sale.service" para que capture el servicio web	Desarrollo
	Creación del módulo de Listado de ventas para el front-end	Desarrollo
	Creación y agregamiento de las clases module.ts y routing.ts al módulo de listado de ventas del front-end	Desarrollo
	Creación del routing para la navegación dentro del sistema web	Desarrollo
	Creación del formulario para la visualización de los registros de ventas en el front-end	Desarrollo
	Asignación de la clase "sale.service" al module.ts del registro de listado de ventas	Desarrollo
	Creación del método "getsale"	Desarrollo
	Importación del componente "ngx-data-table"	Desarrollo
	Mostrar los datos dentro del componente	Desarrollo
Filtros de búsqueda en el registros de ventas	Creación del servicio web rest para la visualización del listado ventas	Desarrollo
	Instalación de los servicios web	Desarrollo
	Agregar el método de "getsaleByCriteria" en la clase "sale.service" para que capture el servicio web	Desarrollo
	Creación de los campos de búsqueda en el módulo de listado de ventas	Desarrollo
	Creación del método para realizarla búsqueda en el front-end	Desarrollo
Exportación de datos	Instalación de componentes para la exportación de datos	Desarrollo
	Creación de los métodos	Desarrollo

Fuente: propia

b. Revisión del sprint

Ya realizadas las tareas del sprint y ejecutado las pruebas pertinentes, se entrega al módulo como terminado, a continuación, se muestran las pantallas del módulo finalizado.

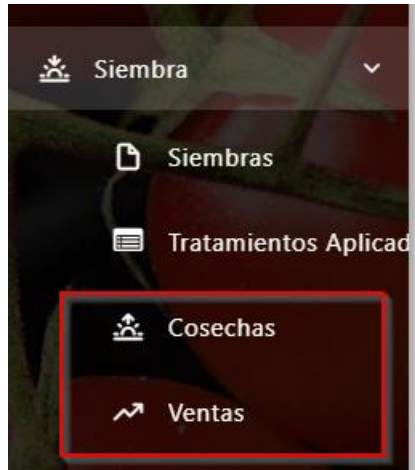
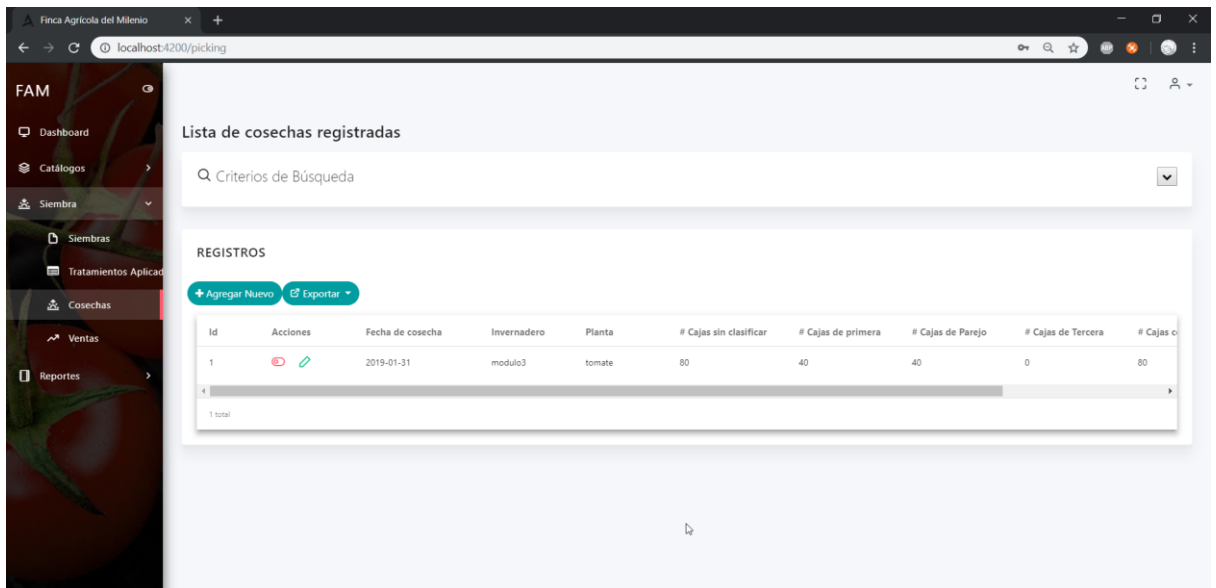


Figura 48 SideBar para Cosechas y Siembras

Fuente: Propia

A screenshot of a web browser displaying a page titled 'Lista de cosechas registradas'. The browser address bar shows 'localhost:4200/picking'. The page has a sidebar on the left with the 'FAM' logo and navigation items: 'Dashboard', 'Catálogos', 'Siembra', 'Siembras', 'Tratamientos Aplicados', 'Cosechas' (highlighted), 'Ventas', and 'Reportes'. The main content area features a search bar labeled 'Criterios de Búsqueda', a 'REGISTROS' section with 'Agregar Nuevo' and 'Exportar' buttons, and a table with the following data:



Id	Acciones	Fecha de cosecha	Invernadero	Planta	# Cajas sin clasificar	# Cajas de primera	# Cajas de Parejo	# Cajas de Tercera	# Cajas de Cuarta
1	 	2019-01-31	modulo3	tomate	80	40	40	0	80

Figura 49 Registros de cosechas

Fuente: Propia

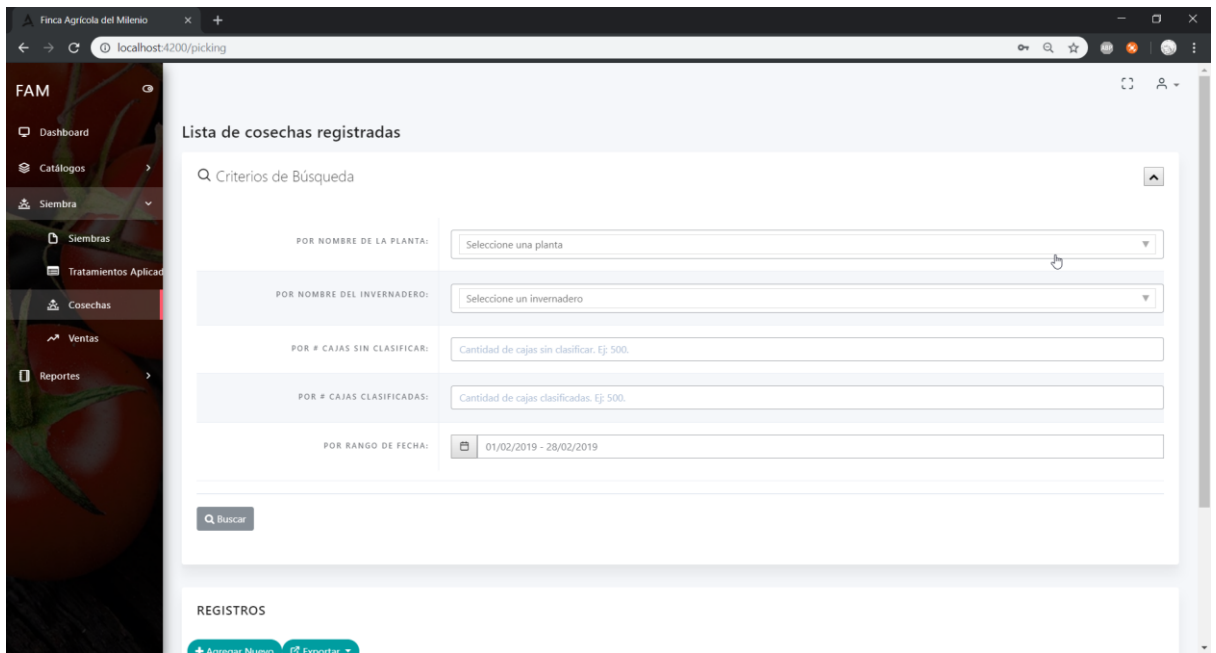


Figura 50 Criterios de Búsqueda, registro de cosechas

Fuente: Propia

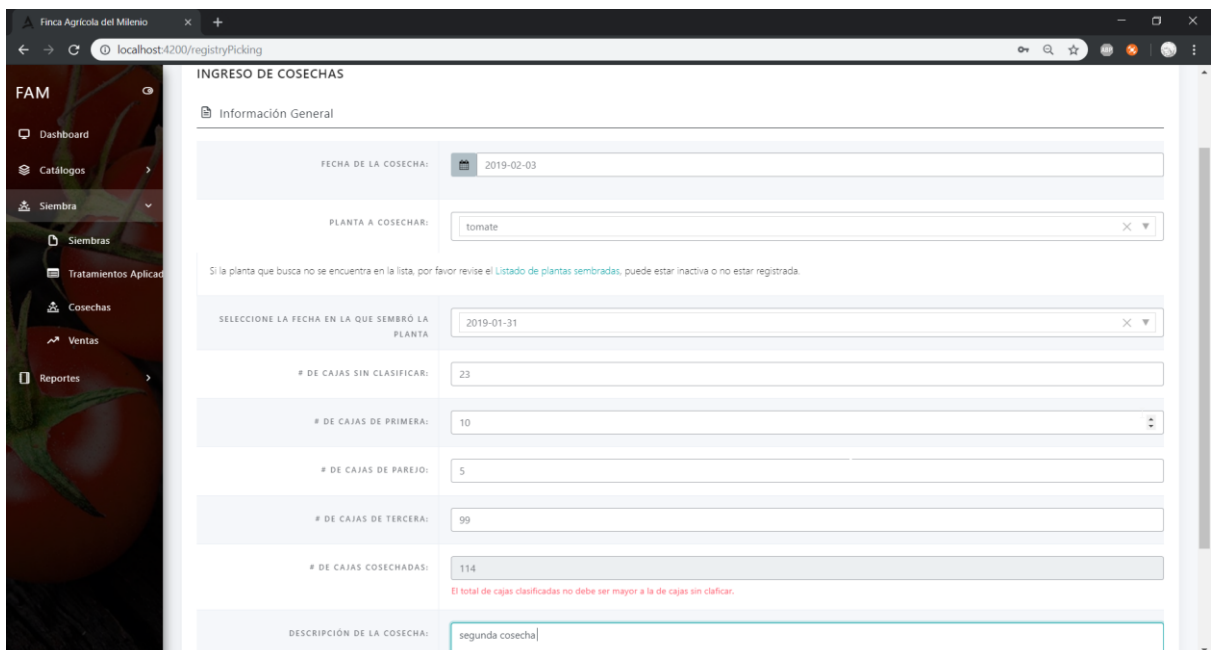


Figura 51 Formulario de registro de cosechas

Fuente: Propia

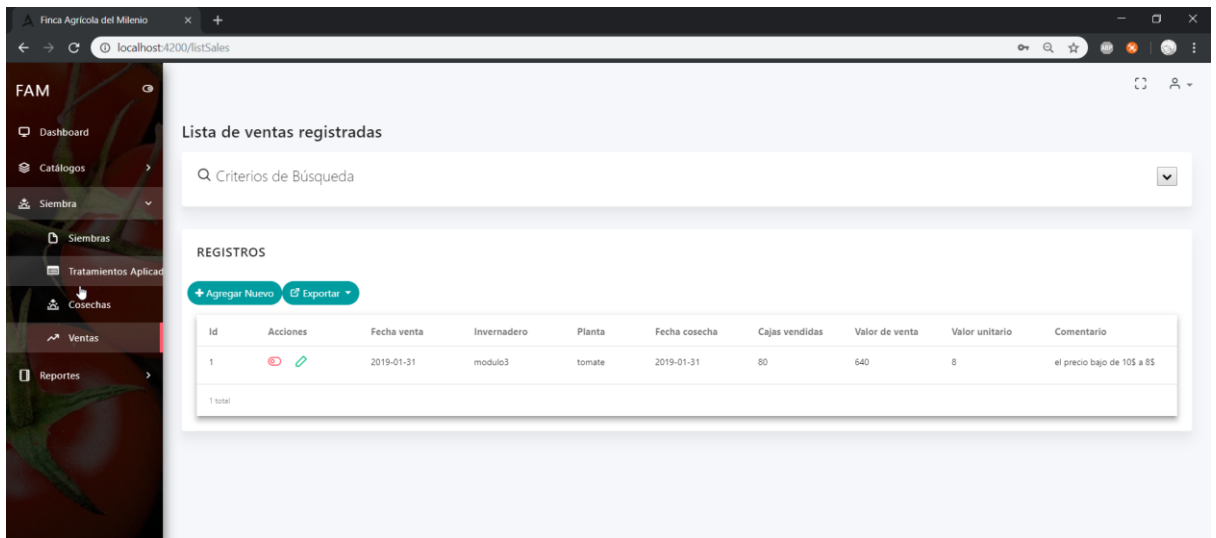


Figura 52 Registros de ventas

Fuente: Propia

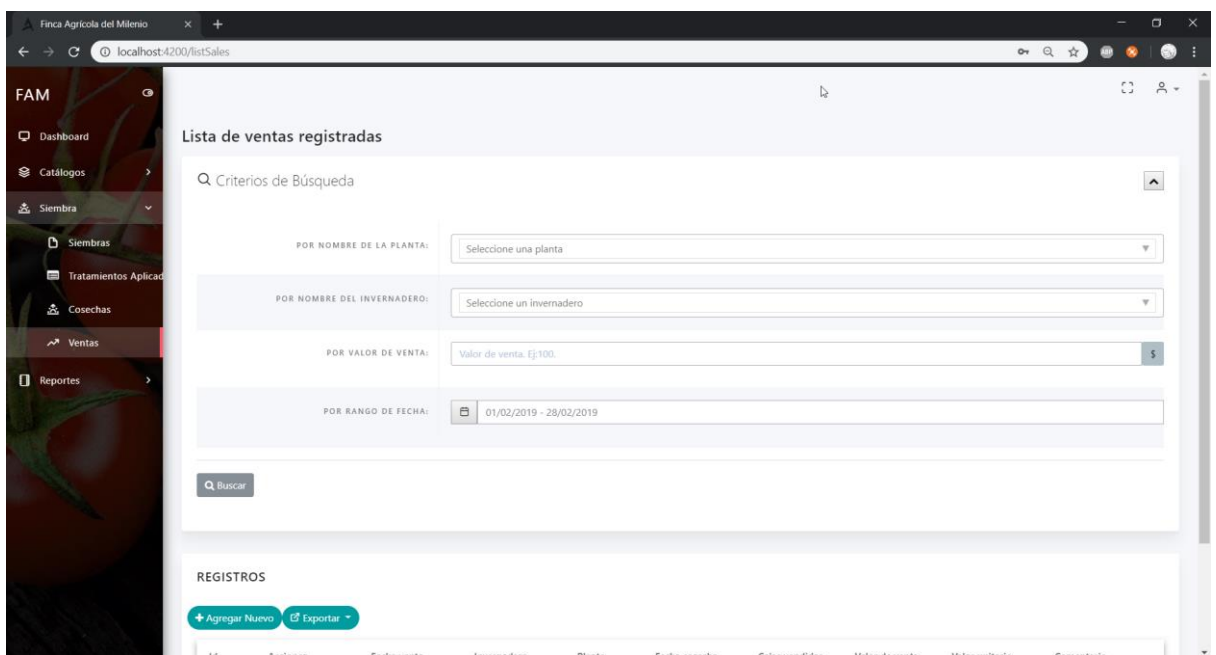


Figura 53 Criterios de Búsqueda, registro de ventas

Fuente: Propia

The image shows a web browser window displaying a form for recording sales. The browser address bar shows 'localhost:4200/registrySales'. The form is titled 'Registro ventas' and 'INGRESO DE VENTAS REALIZADAS'. It includes a sidebar with navigation options like 'Dashboard', 'Catálogos', 'Siembra', 'Siembras', 'Tratamientos Aplicados', 'Cosechas', 'Ventas', and 'Reportes'. The main form area has a section for 'Información General' with the following fields: 'FECHA DE VENTA' (calendar icon, value: 2019-02-03), 'PLANTA COSECHADA' (dropdown menu, value: tomate), 'LOTE A VENDER' (calendar icon, value: 2019-01-31), and 'VALOR DE VENTA' (text input, value: 900, currency symbol: \$). Below these is a 'COMENTARIO DE LA VENTA' text area with a red note 'Este campo es obligatorio.' and two buttons: 'Cancelar' and 'Guardar'.

Figura 54 Formulario de registro de ventas

Fuente: Propia

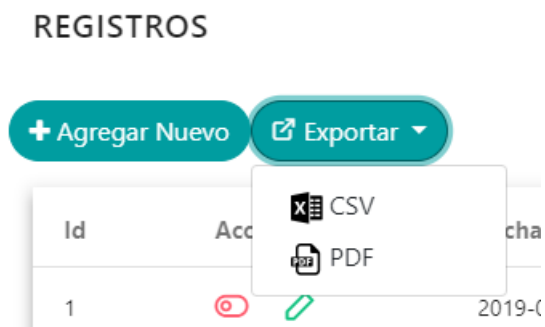


Figura 55 Exportación de archivos, cosechas y ventas

Fuente: Propia

c. Revisión retrospectiva

Debido al correcto análisis de los procesos de cosechas y ventas el desarrollo no tuvo mayores inconvenientes.

Para realizar este sprint fue necesario que el equipo de desarrollo se introduzca a fondo en este proceso pues se debe seguir ciertos criterios para guardar la información correcta, por lo cual el desarrollo se vio paralizado hasta abstraer la información necesaria.

Se deben hacer peticiones al dueño del proceso, con un tiempo de antelación, para que pueda brindar el acceso a los diferentes lugares en donde se realizan los procesos de manera manual.

2.3.7 Sprint 6 Gestión del Dashboard

En este sprint se procede a desarrollar las funcionalidades para el desarrollo de dashboard.

a. Planificación del sprint

Se la realizo con el Product Owner, Scrum Master y el equipo de desarrollo la fecha de 17/09/2018.

Tabla 18 Planificación del sprint 6

ID	Sprint 6
NOMBRE DE ITERACIÓN	Gestión del Dashboard
FECHA INICIAL	17/09/2018
FECHA FINAL	12/10/2018

Fuente: propia

De la reunión se determinó el sprint backlog, que se conforman por las historias de usuario que se desarrollarán en el Sprint 6.

Tabla 19 Sprint backlog del sprint 6

Sprint	S6 Gestión del Dashboard	Inicio	17/09/2018
Responsable	Carlos Proaño	Fin	12/10/2018
Estado	TERMINADO	Duración	160
HISTORIA DE USUARIO	TAREAS	TIPO	
Creación del Dashboard	Creación de los servicios web	Desarrollo	
	Instalación de los servicios web	Desarrollo	
	Instalación de los componentes calendar y Chart	Desarrollo	
	Creación del módulo del Dashboard	Desarrollo	
	Creación y agregamiento de las clases module.ts y rounting.ts al módulo de registro de cosechas del front-end	Desarrollo	
	Implementación de los servicios web	Desarrollo	

Fuente: propia

b. Revisión del sprint

Ya realizadas las tareas del sprint y ejecutado las pruebas pertinentes, se entrega al módulo como terminado, a continuación, se muestran las pantallas del módulo finalizado.

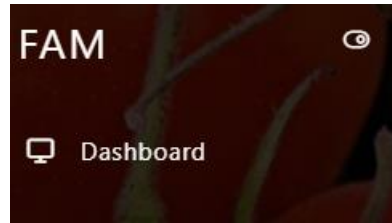


Figura 56 SideBar para el Dashboard

Fuente: Propia

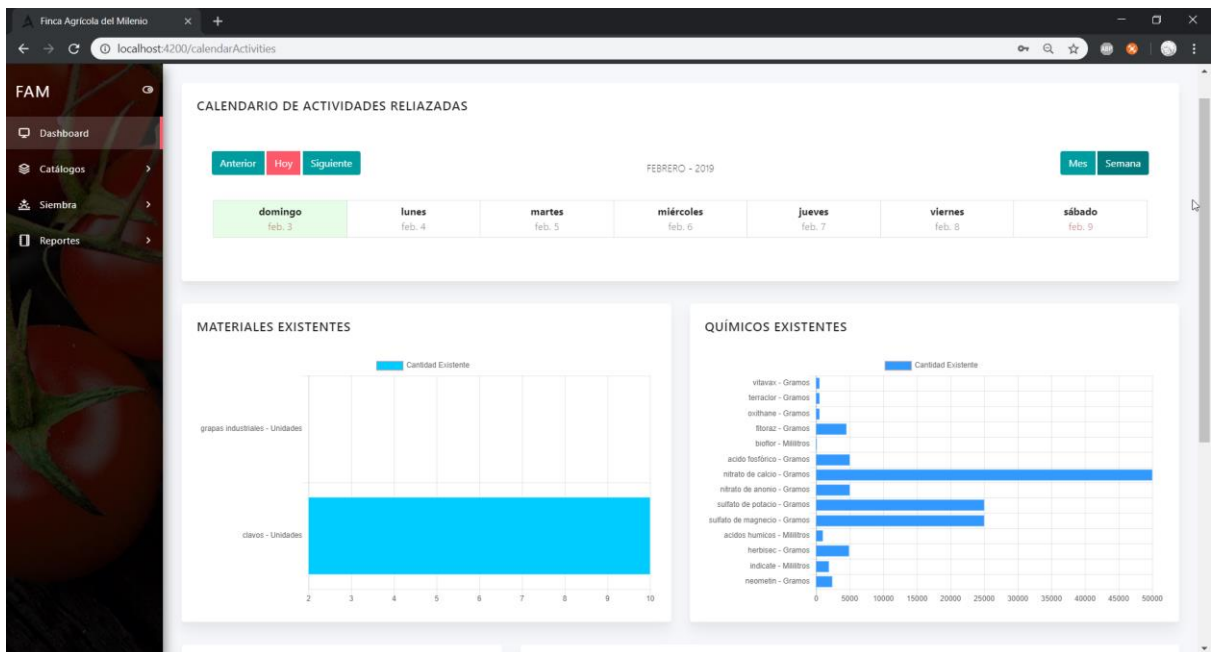


Figura 57 Dashboard, calendario y stock

Fuente: Propia

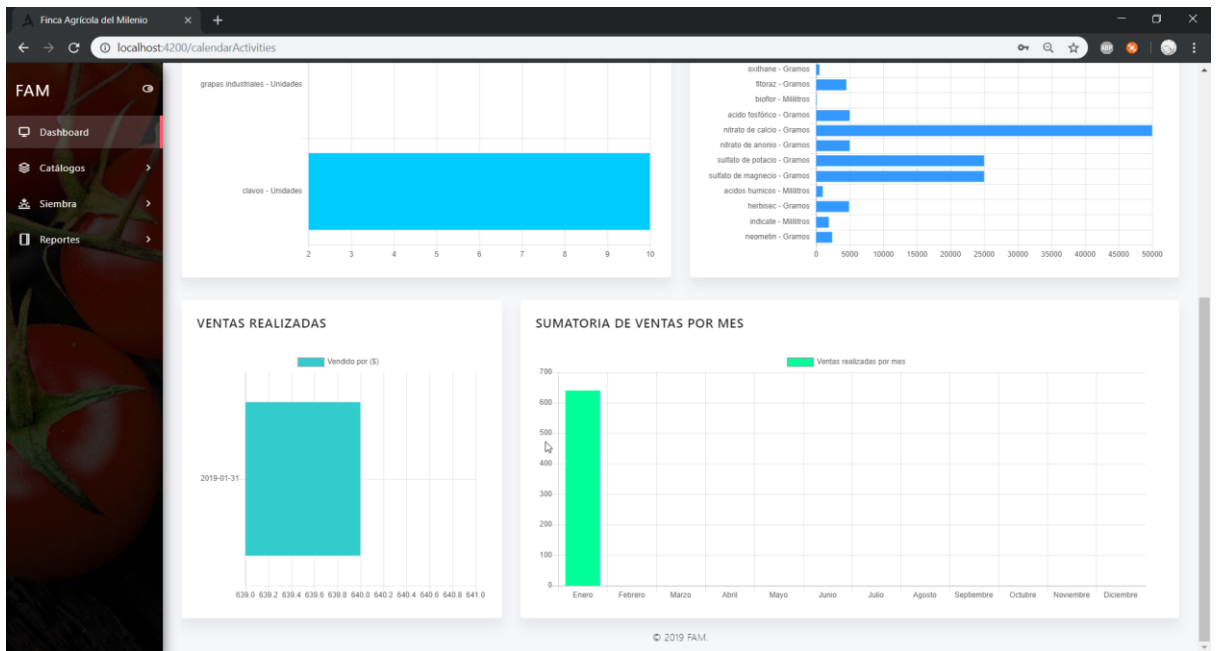


Figura 58 Dashboard, ventas

Fuente: Propia

c. Revisión retrospectiva

El dueño del proceso prestó su ayuda para la captación de datos críticos, los mismos que serán usados para el desarrollo del sprint.

Se produjo un problema al momento de seleccionar el componente que muestra los datos como gráficos, pues no leían correctamente la información y la pantalla de mostraba un error de ejecución.

Es recomendable buscar ayuda en los foros que el mismo componente nos facilita, ya que existen otros desarrolladores que han pasado por los mismos problemas y con su experiencia, se facilita la depuración de los errores.

2.3.8 Sprint 7 Gestión de Acceso

En este sprint se procede a desarrollar las funcionalidades el ingreso y salida del sistema FAM.

a. Planificación del sprint

Se la realizo con el Product Owner, Scrum Master y el equipo de desarrollo la fecha de 05/11/2018.

Tabla 20 Planificación del sprint 7

ID	Sprint 7
NOMBRE DE ITERACIÓN	Gestión de Acceso
FECHA INICIAL	05/11/2018
FECHA FINAL	30/11/2018

Fuente: Propia

De la reunión se determinó el sprint backlog, que se conforman por las historias de usuario que se desarrollarán en el Sprint 6.

Tabla 21 Sprint backlog del sprint 7

Sprint	S7 Gestión de Acceso	Inicio	05/11/2018
Responsable	Carlos Proaño	Fin	30/11/2018
Estado	TERMINADO	Duración	160
HISTORIA DE USUARIO	TAREAS	TIPO	
Inicio de Sesión	Creación del modelo "User" de la tabla "user" en el model de springBoot	Desarrollo	
	Creación del repositorio para el modelo "User"	Desarrollo	
	Creación del servicio web rest "login" para el ingreso al sistema	Desarrollo	
	Instalación de los servicios web	Desarrollo	
	Creación de la clase "user.service" y agregar el método de "login" para que capture el servicio web	Desarrollo	
	Creación y agregamiento de las clases module.ts y routing.ts al módulo de usuarios del front-end	Desarrollo	
	Creación del routing para la navegación dentro del sistema web	Desarrollo	
	Creación de la pantalla para el ingreso al sistema	Desarrollo	
	Creación del modelo "user.model" en el front-end	Desarrollo	
	Creación del método ingresar	Desarrollo	
Cerrar Sesión	Creación del servicio web rest "login" para el ingreso al sistema	Desarrollo	
	Instalación de los servicios web	Desarrollo	
	Creación de la opción "Salir" en el front-end	Desarrollo	
	Creación del método "logout"	Desarrollo	

Fuente: Propia

b. Revisión del sprint

Ya realizadas las tareas del sprint y ejecutado las pruebas pertinentes, se entrega al módulo como terminado, a continuación, se muestran las pantallas del módulo finalizado.

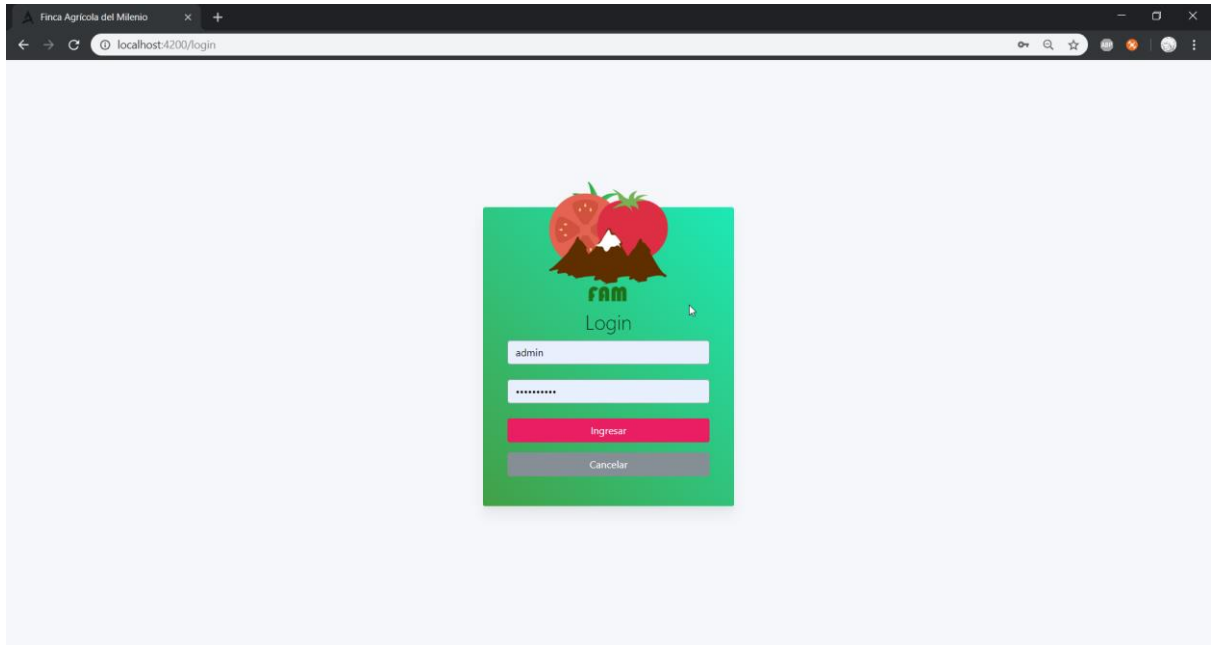


Figura 59 Iniciar Sesión

Fuente: Propia

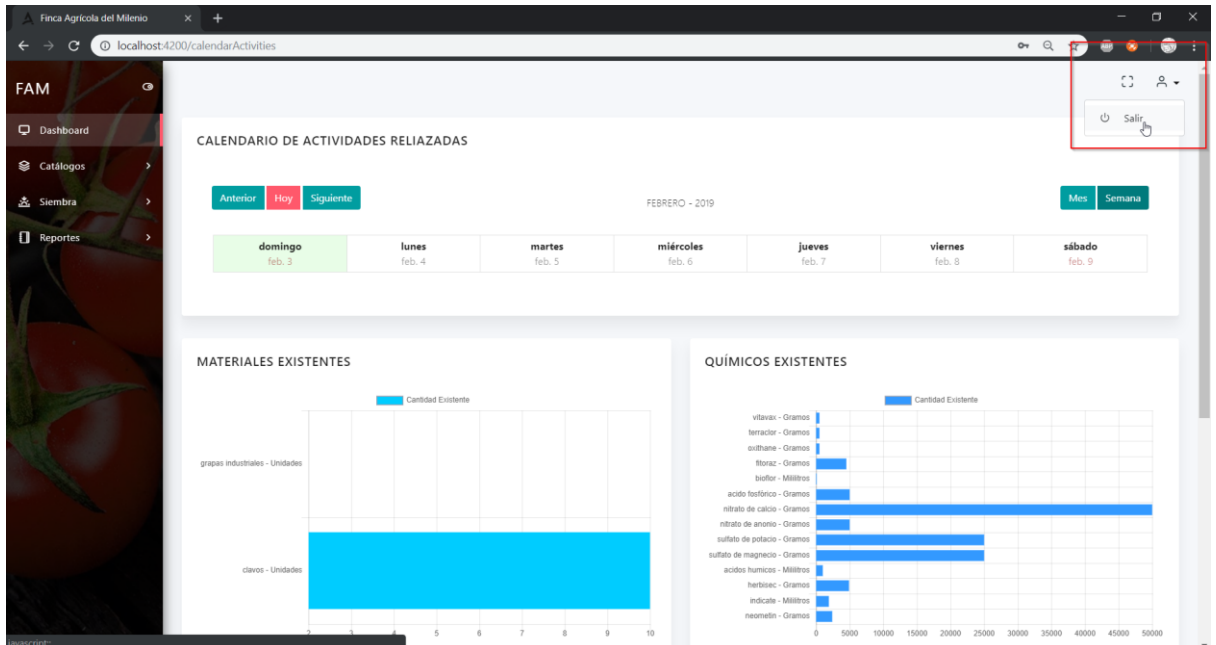


Figura 60 Cerrar Sesión

Fuente: Propia

c. Revisión retrospectiva

No se presentó ningún problema por parte de las reuniones, dado que el product owner se ha acostumbrado a este proceso, y puede expresar mejor sus ideas al equipo de desarrollo.

Se presentó un problema en el desarrollo debido a que uno de los componentes a importar no se encontraba actualizado, por lo cual no permitía ejecutar funciones propias de angular.

Es recomendable realizar un análisis de versiones para estar al tanto de la documentación de los componentes, ya que esta cambia después de cada actualización, y nos muestra cómo solucionar los errores antes mencionados.

CAPÍTULO 3

3. RESULTADOS

3.1 PRUEBAS DE ACEPTACIÓN

Ya culminados con los sprints, con sus respectivas tareas, revisiones y pruebas de funcionalidad, se realiza la entrega del aplicativo resultante. Es necesario realizar estas pruebas de aceptación ya que de esta manera el usuario final puede comprobar que los requerimientos han sido cumplidos en su totalidad.

Tabla 22 Pruebas de aceptación

Sprints		Tareas	Resultado Obtenido
ID	Nombre		
Sprint 0	Análisis y planificación del proyecto	Diseño de la base de datos para el sistema FAM	Éxito
		Configuración de herramientas	Éxito
Sprint 1	Gestión de plantas e invernaderos	Codificación y diseño del módulo de plantas	Éxito
		Codificación y diseño del módulo de invernaderos	Éxito
Sprint 2	Gestión de insumos	Codificación y diseño del módulo de químicos	Éxito
		Codificación y diseño del módulo de materiales	Éxito
Sprint 3	Gestión de Tratamientos y Mantenimientos	Codificación y diseño del módulo de tratamientos	Éxito
		Codificación y diseño del módulo de mantenimientos	Éxito
Sprint 4	Gestión de Siembras y Tratamientos Aplicados	Codificación y diseño del módulo de siembras	Éxito
		Codificación y diseño del módulo de tratamientos aplicados	Éxito
Sprint 5	Gestión de Cosechas y Ventas	Codificación y diseño del módulo de cosechas	Éxito
		Codificación y diseño del módulo de ventas	Éxito
Sprint 6	Gestión del Dashboard	Codificación y diseño del dashboard	Éxito
Sprint 7	Gestión de Acceso	Codificación y diseño para el ingreso y salida del sistema	Éxito

Fuente: Propia

Con ayuda del Scrum Master, el desarrollo de todos los módulos propuestos han sido concluidos con éxito, cumpliendo a cabalidad las pruebas de funcionalidad, con lo cual el proyecto se entrega como culminado a FAM.

3.2 BENEFICIOS DE LA IMPLEMENTACIÓN

- Facilita un registro histórico y ordenado de todos los procesos almacenados en el sistema.
- Para un realizar un registro, el sistema muestra al usuario datos consistentes al registro, con el fin de que se almacenen datos congruentes.
- Evita perdida de información, ya que maneja estados de activo e inactivo, con lo cual los registros siempre estarán en el sistema.
- Evita el daño de los registros, ya que estos no pueden ser comprometidos por la humedad, incendios y demás imprevistos.
- Muestra al usuario las cantidades existentes de los materiales y químicos que se guardan en bodega.
- Permite visualizar un calendario con las actividades realizadas en la finca.
- Agiliza el proceso de visualización de ganancia de ventas por meses.
- Permite manipular el control de inventarios, conforme se los va utilizando, mediante los registros de tratamientos y mantenimientos.
- Es capaz de generar archivos PDF y Excel con información verídica y seleccionada por el mismo usuario.
- Presenta una interfaz amigable para el usuario, permitiendo que la ejecución de operaciones sea eficiente.
- Disminuye significativamente el uso de papel, debido a que no es necesario mantener respaldos físicos.

3.2.1 Indicadores de beneficios

Con el fin de medir los beneficios que ha proporcionado el sistema web a FAM, se hacen uso de indicadores de beneficios (Gestión & Gestión, 2019). Considerando que un indicador es un conjunto de variables, que nos muestran mediciones que están relacionadas con los servicios o productos generados por la institución, y su resultante es un valor comparativo haciendo énfasis a una meta en específico.

Para la elaboración de los indicadores es necesario plantearse las siguientes preguntas:

- ¿Qué se hace?
- ¿Qué se desea medir?
- ¿Quién utilizará la información?
- ¿Con qué o quién se compara?

Considerando que un indicador es un conjunto de variables, que nos muestran mediciones que están relacionadas con los servicios o productos generados por la institución, y su resultante es un valor comparativo haciendo énfasis a una meta en específico (Carlos & Mejía, 2019).

Por medio de una reunión del scrum master, product owner y el equipo de desarrollo el análisis de beneficios se enfocará en:

- Eficiencia

Es la capacidad de producir un máximo de resultados con un mínimo esfuerzo.

Tabla 23 Tabla de eficiencia

Tipo	Indicador	Manual (minutos)	Sistema (minutos)	Mejora (%)	Promedio (%)
Eficiencia	Listado de plantas	6	1	83,33	86,50
	Listado de invernaderos	6	1	83,33	
	Listado siembras	6	1	83,33	
	Listado de cosechas	6	1	83,33	
	Listado de ventas	6	1	83,33	
	Listado de Mantenimientos	6	1	83,33	
	Listado de Tratamientos Aplicados	6	1	83,33	
	Ventas sumariadas por mes	20	1	95,00	
	Stock de materiales	15	1	93,33	
	Stock de químicos	15	1	93,33	

Fuente: Propia

La Tabla 22 es el resultante de la medición en tiempo real, de los procesos manuales y los procesos que lleva acabo el sistema de FAM.

- Eficacia

Nos muestra el grado en el que los objetivos y metas han sido alcanzados.

Tabla 24 Tabla de eficacia

Tipo	Indicador	Funcionalidades disponibles	Funcionalidades propuestas	Grado Alcanzado (%)
Eficacia	Listado de plantas	5	5	100
	Listado de invernaderos	4	4	100
	Listado siembras	4	4	100
	Listado de cosechas	4	4	100
	Listado de ventas	4	4	100
	Listado de Mantenimientos	7	7	100
	Listado de Tratamientos Aplicados	7	7	100
	Ventas sumarizadas por mes	5	5	100
	Stock de materiales	4	4	100
	Stock de químicos	4	4	100

Fuente: Propia

3.3 ANÁLISIS DE IMPACTOS

El análisis de impactos tiene como objetivo analizar las repercusiones que ha tenido una organización después de que se ha implementado un sistema (Posso, 2013).

Tabla 25 Análisis de impactos

Valor	Impacto
-3	Impacto alto negativo
-2	Impacto medio negativo
-1	Impacto bajo negativo
0	No hay impacto
1	Impacto bajo positivo
2	Impacto medio positivo
3	Impacto alto positivo

Fuente: Propia

3.3.1 Impacto socio cultural

Tabla 26 Impacto socio cultural

Indicador	Nivel de impacto							Total
	-3	-2	-1	0	1	2	3	
Transferencias Tecnológicas							x	3
Formación						x		2
Bienestar							x	3
Total						2	6	8

Fuente: Propia

$$\text{Total de impacto: } \frac{\sum \text{Nivel de impacto}}{\text{número de indicadores}}$$

$$\text{Total de impacto} = \frac{8}{3}$$

$$\text{Total de impacto} = 2.67$$

El resultado indica que en un periodo mediano de tiempo la implantación del sistema beneficia a FAM.

3.3.2 Impacto económico

Tabla 27 Impacto económico

Indicador	Nivel de impacto							Total
	-3	-2	-1	0	1	2	3	
Incremento de productividad							x	3
Generación de empleo			x					-1
Total			-1				3	2

Fuente: Propia

$$\text{Total de impacto: } \frac{\sum \text{Nivel de impacto}}{\text{número de indicadores}}$$

$$\text{Total de impacto} = \frac{2}{2}$$

$$\text{Total de impacto} = 1$$

El resultado nos indica que tendrá un impacto bajo positivo debido a que disminuirá la necesidad de contratar a personas debido a que se automatiza varios procesos, y debido a eso se incrementara la productividad en FAM.

3.3.3 Impacto ambiental

Tabla 28 Impacto ambiental

Indicador	Nivel de impacto							Total
	-3	-2	-1	0	1	2	3	
Disminución de materiales de oficina							x	3
Total							3	3

Fuente: Propia

Total de impacto: $\frac{\sum \text{Nivel de impacto}}{\text{número de indicadores}}$

$$\text{Total de impacto} = \frac{3}{1}$$

$$\text{Total de impacto} = 3$$

El sistema de FAM, será muy beneficio para el ambiente debido a que disminuirá frásticamente el uso de papel, tinta, etc.

3.4 CONCLUSIONES Y RECOMENDACIONES

- **Conclusiones**

Se constató que los procesos efectuados en la FAM, previo a la implantación del sistema web, requieren una gran cantidad de esfuerzo y tiempo, por parte de los responsables de la ejecución de los distintos procesos. Las operaciones ejecutadas cotidianamente y de forma manual consistían en llevar un registro a mano de las plantas, invernaderos, siembras, fumigaciones, mantenimientos, cosechas y ventas, lo que conlleva a la inversión de mucho tiempo sobre todo cuando surgía la necesidad de realizar un análisis del stock existente o un análisis de ventas.

Con la metodología de desarrollo SCRUM, realizar entregables del sistema en periodos cortos de tiempo es posible debido a su forma de interactuar con el usuario pues posibilita que el proceso de desarrollo sea rápido y conciso, de igual forma, es posible documentar dicho proceso de manera fácil, sencilla y estructurada, la participación del usuario en el desarrollo garantiza la entrega de un sistema o módulo de calidad que cumpla con las expectativas del usuario final.

Gracias a la implantación de todos los módulos propuestos para FAM, se impulsó y mejoró la eficiencia en un 86.50% ya que la manera en la que se llevan los registros, evita realizar registros manuales, agilizando el proceso de ingreso y más adelante la lectura de datos registrados, y así mismo permitiendo reduce gastos en el uso de papel.

Con los resultados de los indicadores de beneficios podemos concluir que el sistema es de gran ayuda para FAM, debido a su rápida respuesta antes las diferentes peticiones del usuario.

- **Recomendaciones**

Es recomendable que el desarrollador se involucre en el entorno, para conocer y comprender todos los procesos que se realizan diariamente, además de analizar el mecanismo de la ejecución de las distintas actividades, con el fin de realizar un correcto levantamiento de requerimientos, y por ende integrar de la mejor manera las funcionalidades que la entidad requiere en la solución tecnológica que va a ser elaborada.

Se recomienda hacer uso de la metodología SCRUM para la elaboración de sistemas que requieran un desarrollo rápido y constante presentación de resultados.

Es necesario involucrar al usuario final en el proceso de implantación del sistema a través de etapas de pruebas, con el fin de facilitar la introducción del nuevo proceso de registro y así aprovechar al máximo los beneficios que brinda.

Es recomendable hacer uso del framework Angular 4 para el desarrollo de sistemas de gestión, debido a su gran adaptabilidad a entornos de desarrollo, con lo cual reduce el tiempo de programación. Y conjuntamente con SCRUM hacen posible el desarrollo de aplicaciones muy versátiles.

REFERENCIAS

- Alegsa.com.ar. (2019). Ventajas y desventajas de las aplicaciones web. Retrieved from http://www.alegsa.com.ar/Respuesta/ventajas_y_desventajas_de_las_aplicaciones_web.htm
- API_MARKET, B. (2017). Gestión de Proyectos Ágiles y SCRUM una selección de plataformas útiles. Retrieved from <https://bbvaopen4u.com/es/actualidad/gestion-de-proyectos-agile-y-scrum-una-seleccion-de-plataformas-utiles>
- Aranda, S. (2017). Historia de SCRUM. Retrieved from <https://proyectosagiles.org/historia-de-scrum/>
- Barcelona, U. de. (2002). JIT, 89–97. Retrieved from http://www.ub.edu/gidea/recursos/casseat/JIT_concepte_carac.pdf
- Barzanallana, R. (2019). Servicios en internet Historia del desarrollo de aplicaciones Web. Retrieved from <https://www.um.es/docencia/barzana/DIVULGACION/INFORMATICA/Historia-desarrollo-aplicaciones-web.html>
- CampusMVP. (2019). ¿Es Angular 2, Angular 4 o simplemente Angular? Retrieved from <https://www.campusmvp.es/recursos/post/es-angular-2-angular-4-o-simplemente-angular.aspx>
- Carlos, P., & Mejía, A. (2019). INDICADORES DE EFECTIVIDAD Y EFICACIA, (76).
- Díaz, R. A. (2016). ¿Por qué Angular 2? Retrieved from <https://www.adictosaltrabajo.com/tutoriales/por-que-angular-2/>
- Eclipse. (2019). Eclipse. Retrieved from <https://help.eclipse.org/2018-12/index.jsp>
- Editores Noriega. (2015). La filosofía del justo a tiempo. *En Busca de La Excelencia Industrial. JIT*, 9–33. Retrieved from http://www.mercadeo.com/16_jitsales.htm#¿Qué es Justo a Tiempo - JIT?
- Garzas, J. (2012). Encuesta sobre el estado de lo ágil. Retrieved from <http://www.javiergarzas.com/2012/02/encuesta-agil-2011.html>
- Gestión, I. D. E., & Gestión, D. D. E. I. D. E. (2019). LORINO, Philippe, Óp. Cit, p. 194 67, (6).
- Hora, D. la. (2013, May 22). Producción agrícola en Ecuador, p. 1. Retrieved from <https://www.lahora.com.ec/noticia/1101510042/produccion-agricola-en-ecuador>

- iebschool. (2019). Las metodologías ágiles más utilizadas y sus ventajas dentro de la empresa. Retrieved from <https://www.iebschool.com/blog/que-son-metodologias-agiles-agile-scrum/>
- Manuel Trigas Gallego. (2019). Metodologia SCRUM mtrigasTFC0612memoria. Retrieved from <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/17885/1/mtrigasTFC0612memoria.pdf>
- MELÉNDEZ, E. (2019). Introducción a Spring Boot. Retrieved from <http://blog.eddumelendez.me/2016/09/introduccion-a-spring-boot/>
- Oracle. (2019). MySQL. Retrieved from <https://www.oracle.com/technetwork/database/mysql/index.html>
- Oriol, E. (2016). Introducción a Angular 2, Modulo, Componente, Template y Metadatos. Retrieved from <http://blog.enriqueoriol.com/2016/06/introduccion-a-angular-2-parte-i-componente.html>
- Oriol, E. (2019a). Introducción a Angular 2 (parte I) – Modulo, Componente, Template y Metadatos. Retrieved from <http://blog.enriqueoriol.com/2016/06/introduccion-a-angular-2-parte-i-componente.html>
- Oriol, E. (2019b). Introducción a Angular 2 (parte II) – Data Binding y Directivas. Retrieved from <http://blog.enriqueoriol.com/2016/07/introduccion-angular-2-parte-ii-binding-directivas.html>
- Oriol, E. (2019c). Introducción a Angular 2 (parte III) – Servicios. Retrieved from <http://blog.enriqueoriol.com/2016/07/introduccion-angular-2-parte-iii-servicios.html>
- Oriol, E. (2019d). Introducción a Angular 2 (parte IV) – Dependency Injection. Retrieved from <http://blog.enriqueoriol.com/2016/07/introduccion-angular-2-parte-iv-dependency-injection.html>
- Palacio, A., Menzinsky, G., & López, J. (2016). Scrum Manager, 94. Retrieved from http://www.scrummanager.net/files/sm_proyecto.pdf
- Posso, M. (2013). Proyectos, tesis y marco lógico: Planes e informes de investigación, 334.
- Proyectos ágiles. (2019). Qué es SCRUM. Retrieved from <https://proyectosagiles.org/que-es-scrum/>
- Softeng. (2019). Proceso y Roles de Scrum. Retrieved from <https://www.softeng.es/es-es/empresa/metodologias-de-trabajo/metodologia-scrum/proceso-roles-de-scrum.html>

Spring. (2019). SPRING INITIALIZR bootstrap your application now. Retrieved from <https://start.spring.io/>

Sutherland, K., & Schwaber, J. (2013). La guía de SCRUM, 21. Retrieved from <http://www.scrumguides.org/docs/scrumguide/v1/scrum-guide-es.pdf>

Telégrafo, D. el. (2017, April 2). El agro concentra el 29,3% de empleos en el país. Retrieved from <https://www.eltelegrafo.com.ec/noticias/economia/8/el-agro-concentra-el-29-3-de-empleos-en-el-pais>

Uniwebsidad. (2019). Principios S.O.L.I.D. Retrieved from <https://uniwebsidad.com/libros/tdd/capitulo-7/principios-solid?from=librosweb>