



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO
FACULTAD DE CIENCIAS DE LA INGENIERÍA
ESCUELA DE INFORMÁTICA

TESIS DE GRADO
PREVIO A LA OBTENCIÓN DEL TÍTULO
DE INGENIERO EN SISTEMAS

TEMA:

“Construcción de una Guía Operativa basada en software para la Gestión y Control de la Producción y Comercialización de la Asociación de Productores Bananeros de Ventanas APROBAVEN”

Autor:

CÉSAR FABIÁN VARAS BELTRÁN

Director de Tesis

Ing. Ariosto Vicuña Msc.

Quevedo – Ecuador.

Año

2014

AUTORÍA Y CESIÓN DE DERECHOS

Yo, César Fabián Varas Beltrán, declaro que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

La Universidad Técnica Estatal de Quevedo, puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

CÉSAR FABIÁN VARAS BELTRÁN
AUTOR

CERTIFICACIÓN

El suscrito, Ing. Ariosto Vicuña, Docente de la Universidad Técnica Estatal de Quevedo, certifica que el Egresado César Fabián Varas Beltrán, realizó la tesis de grado titulada “**Construcción de una Guía Operativa basada en software para la gestión y control de la producción y comercialización de la Asociación de Productores Bananeros de Ventanas APROBAVEN**”, bajo mi dirección, habiendo cumplido con las disposiciones reglamentarias establecidas para el efecto.

Ing. Msc. Ariosto Vicuña

DIRECTOR DE TESIS



TRIBUNAL DE TESIS

UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO

FACULTAD DE CIENCIAS DE LA INGENIERÍA

CARRERA DE INGENIERÍA EN SISTEMAS

Presentado al Consejo Directivo como requisito previo a la obtención del título de Ingeniero en Sistemas.

Aprobado:

Dr. PhD Amilkar Puris
PRESIDENTE DEL TRIBUNAL DE TESIS

Ing. Washington Chiriboga
MIEMBRO DEL TRIBUNAL DE TESIS

Ing. Carlos Márquez de la Plata
MIEMBRO DEL TRIBUNAL DE TESIS

QUEVEDO – LOS RÍOS – ECUADOR

AÑO 2014

DEDICATORIA Y AGRADECIMIENTO

DEDICATORIA

A Dios quien me dio el don la vida. A mis padres César y Julieta que siempre velaron por bienestar, gracias por apoyarme incondicionalmente en todo, y supieron guiarme por el camino correcto. A mi hermana Gabriela que compartimos juntos toda nuestra vida estudiantil gracias hermana por estar siempre a mi lado y lograr salir siempre adelante juntos. A mi abuelita Noemí que nunca dejo de apoyarme.

A mi esposa Daniela, por su confianza brindada, su paciencia durante la realización de este trabajo y por todo su amor.

A dos angelitos de Dios César y Vicky que con su llegada me hicieron ver el mundo de otra manera, llenándome de alegría y darme una razón más para nunca desfallecer.

César Fabián Varas Beltrán

AGRADECIMIENTO

El autor deja constancia de sus agradecimientos a todas las personas que con su colaboración, hicieron posible la culminación de este trabajo.

Agradezco primeramente a Dios por todas sus bendiciones, por la vida que me ha dado y haberme permitido culminado mi carrera universitaria.

A mis padres César y Julieta, hermana Gaby, la mujer que amo Daniela y a esos dos angelitos Cesar, Victoria y a mis tíos Libia y Rogelio. Gracias a todos Ustedes por su apoyo brindado y por ser las razones que me mantiene con ganas de siempre triunfar en la vida.

Agradecimiento al Ing. Ariosto Vicuña Pino Msc., mi director de tesis, quien compartió conmigo sus valiosos conocimientos, brindándome su apoyo incondicional durante todas las circunstancias en el desarrollo de este trabajo.

A todas las personas que estuvieron siempre presente durante toda mi vida estudiantil, les agradezco sinceramente por todo su apoyo brindado y su confianza.

Al apreciado y estimado docente universitario Ing. Milciades Fernández por su valiosa amistad y apoyo durante mi vida estudiantil.

MATRIZ DE DUBLIN

(DUBLIN CORE) ESQUEMAS DE CODIFICACIÓN			
1.	Título/Title	M	“Construcción de una Guía Operativa basada en software para la Gestión y Control de Producción y Comercialización de la Asociación de Productores Bananeros de Ventanas APROBAVEN”
2.	Creador/Creator	M	Varas Beltrán César Fabián; APROBAVEN
3.	Materia/Subject	M	Ciencias de la Ingeniería; Ingeniería en Sistemas; Aplicación Web; Guía para mejorar el control del manejo agrícola de las fincas productoras de banano
4.	Descripción/Description	M	La presente investigación fue realizada en Ventanas, Provincia de los Ríos. El objetivo principal fue de desarrollar un software para el control efectivo de las operaciones que realizan en APROBAVEN.
5.	Editor/Publisher	M	FCI; Carrera Ingeniería en Sistemas; Varas Beltrán César Fabián, APROBAVEN.
6.	Colaborador/Contributor	O	Ninguno.
7.	Fecha/Date	M	21 de Marzo 2014
8.	Tipo/Type	M	Trabajo de Investigación
9.	Formato/Format	R	Microsoft Office Word 2013 (*.doc) Adobe Reader (*.pdf)

ÍNDICE

AUTORÍA Y CESIÓN DE DERECHOS.....	ii
CERTIFICACIÓN	iii
TRIBUNAL DE TESIS.....	iv
DEDICATORIA Y AGRADECIMIENTO	v
DEDICATORIA	v
AGRADECIMIENTO	vi
MATRIZ DE DUBLIN	vii
RESUMEN	xv
ABSTRACT	xvi
CAPÍTULO I	
MARCO CONTEXTUAL DE LA INVESTIGACIÓN
1.1 INTRODUCCIÓN.....	2
1.2 JUSTIFICACIÓN.....	4
1.3 PROBLEMATIZACIÓN	5
1.4 OBJETIVOS	7
1.4.1 GENERAL	7
1.4.2 ESPECÍFICOS	7
1.5 HIPÓTESIS	8
1.5.1 GENERAL	8
1.5.2 VARIABLES	8
CAPÍTULO II.....	
MARCO TEÓRICO	
2.1 FUNDAMENTACIÓN TEÓRICA.....	11
2.1. INFORMACIÓN BASICA	11

2.1.1 Guía Operativa	11
2.1.2 Asociación De Productores Bananero De Ventanas (Aprobaven)	12
2.1.3 Labores Agrícolas.....	12
2.1.4 Calidad De La Fruta.	13
2.1.5 Cosecha.	13
2.1.6 Manejo Agrícola	13
2.1.7 Hirbernate.....	13
2.1.7.1 Objetos Persistentes	15
2.1.8 Javabeans	16
2.1.10.1 Ventajas y Desventajas del uso Del Patrón.....	19
2.1.11 Java.....	24
2.2 MARCO REFERENCIAL	26
2.2.1 Análisis y diseño e implementación de una guía operativa de cultivo para productos de exportación.	26
CAPÍTULO III	
METODOLOGÍA DE LA INVESTIGACIÓN.....	
3.1 MATERIALES Y MÉTODOS	28
3.1.1 Materiales	28
3.1.1.1 Hardware	28
3.1.1.2 Software	28
3.1.1.3 Suministros.....	29
3.1.1.4 Personal	29
3.2 MÉTODOS	29
3.2.1 Modelo Incremental.	29
3.2.1.1 Fases del Ciclo	30
3.2.2 Método de Desarrollo del Sitio Web	31
3.2.2.1 Elementos del RUP.	32

3.2.2.2 Fases de diseño de metodología RUP.....	32
3.2.2.3 Etapas del RUP.....	33
3.2.2.3.1 Inicio.....	33
3.2.2.3.2 Elaboración	33
3.2.2.3.3 Construcción.	33
3.2.2.3.3.1 ¿Cuál es la finalidad del Modelo Vista Controlador?	34
3.2.3 Metodología de Comprobación de Hipótesis.....	35
3.2.4 Método de Comprobación de Hipótesis.....	36
3.2.4.1 Parametrización de Variables Independientes.	36
3.2.4.1.1 Dimensión Seguimiento de labores	36
3.2.4.1.2 Dimensión Generación de reportes	36
3.2.4.2 Medición de Variables Dependientes.	37
3.2.4.2.1 Dimensión Fiabilidad.	37
CAPÍTULO IV.....	
RESULTADOS Y DISCUSIÓN	
4.1. RESULTADO DE DESARROLLO DEL SITIO WEB	39
4.1.1 Inicio.	39
4.1.2.1 Casos de Usos	41
4.1.2.2 Diagrama de Actividades.....	69
4.1.2.3 Diagramas de Secuencia	83
4.1.2.4 Diagrama de Clase	97
4.1.2.4 Diagrama de Componente	97
4.1.2.5 Diagrama de Despliegue	98
4.1.2 Construcción.....	98
4.1.2.1 Capa 1 Base de Datos.	98
4.1.2.2 Capa 2 Reglas de Negocio.....	101
4.1.2.3 Capa 3 Vista.....	105

4.2. RESULTADO DE COMPROBACIÓN DE HIPÓTESIS	106
4.2.2 Control Continuo del Manejo Agrícola	109
CAPÍTULO V.....	
CONCLUSIONES Y RECOMENDACIONES.....	
5.1 CONCLUSIONES	113
5.2. RECOMENDACIONES.....	114
CAPÍTULO VI.....	
BIBLIOGRAFÍA.....	
6.1 LITERATURA CITADA	116
6.2 LINKOGRAFÍA.....	117
CAPÍTULO VI.....	
ANEXOS.....	118

INDICE DE TABLA

Tabla 1 Estadística de banano a nivel mundial.....	2
Tabla 2 Matriz de conceptualización y operacionalización de variable independiente.....	8
Tabla 3 Matriz de conceptualización y operacionalización de variable dependiente.....	9
Tabla 4 Características del banano	11
Tabla 5 Tabla de métodos utilizados en la aplicación web	103
Tabla 6 Datos de documentos perdidos	106
Tabla 7 Datos para la prueba T - Student.....	107
Tabla 8 Prueba T con función de excel.....	108
Tabla 9 Datos de encuesta a socios	109
Tabla 10 Tabla de frecuencia de h_0	109
Tabla 11 Tabla de frecuencia de h_1	110

ÍNDICE DE ILUSTRACIONES

Ilustración 1 Modelo Vista Controlador	18
Ilustración 2 Diseño Incremental.....	31
Ilustración 3 Diagrama de Casos de Usos.....	40
Ilustración 4 Diagrama de Actividades de Inicio Sesión.....	69
Ilustración 5 Diagrama De Actividades De Registrar Usuario	70
Ilustración 6 Diagrama de Actividades de Registrar Socio	71
Ilustración 7 Diagrama de Actividades de Registrar Hacienda	72
Ilustración 8 Diagrama de Actividades de Asignación de Lotes	73
Ilustración 9 Diagrama de Actividades de Registrar Labores Agrícolas.....	74
Ilustración 10 Diagrama de Actividades de Registrar Característica de Labores	75
Ilustración 11 Diagrama de Actividades de Planificación De Labores	76
Ilustración 12 Diagrama de actividades de control de planificación	77
Ilustración 13 Diagrama de Actividades de Registrar Personal	78
Ilustración 14 Diagrama de Actividades de Registrar Estimación	79
Ilustración 15 Diagrama de Actividades de Registrar Producción	80
Ilustración 16 Diagrama de Actividades de Recuperar Clave	81
Ilustración 17 Diagrama de Actividades de Modificar Socio	82
Ilustración 18 Diagrama de Secuencia de Iniciar Sesión	83
Ilustración 19 Diagrama de Secuencia de Registrar Usuario.....	84
Ilustración 20 Diagrama de Secuencia de Registrar Hacienda.....	85
Ilustración 21 Diagrama de Secuencia de Asignación De Lotes.....	86
Ilustración 22 Diagrama de Secuencia de Registrar Labores Agrícolas	87
Ilustración 23 Diagrama de Secuencia de Registrar Característica De Labor ..	88
Ilustración 24 Diagrama de Secuencia de Registrar Socio	89
Ilustración 25 Diagrama de Secuencia de Registrar Planificación	90
Ilustración 26 Diagrama de Secuencia de Control De Planificación	91
Ilustración 27 Diagrama de Secuencia de Registrar Personal.....	92
Ilustración 28 Diagrama de Secuencia de Registrar Estimación	93
Ilustración 29 Diagrama de Secuencia de Registrar Producción	94
Ilustración 30 Diagrama de Secuencia de Recuperar Clave.....	95

Ilustración 31 Diagrama de Secuencia de Modificar Socio	96
Ilustración 32 Diagrama de Clase	97
Ilustración 33 Diagrama de Componente.....	97
Ilustración 34 Diagrama de Despliegue	98
Ilustración 35 Diseño Lógico de la BD	99
Ilustración 36 Diseño Físico de la BD	100
Ilustración 37 Gráfico de Frecuencia de H_0	110
Ilustración 38 Gráfico de Frecuencia de H_1	110

RESUMEN

La necesidad de contar con un aplicativo web para controlar las labores de cultivo banano para la Asociación de Productores bananeros de Ventanas “APROBAVEN” diseñada con los estándares de la entidad antes mencionada.

El análisis de requerimientos se lo hizo mediante información brindada por la directiva de APROBAVEN, se recopiló información realizando encuestas a personas involucradas en el argot bananero.

Se planteó como objetivo principal desarrollar un software para el control efectivo de las operaciones que se realizan en APROBAVEN, que permite gestionar el manejo de las labores agrícolas, la aplicación admite registrar labores con sus respectivos parámetros, realizar estimación de producción, el acceso al sistema solo lo hacen miembros de la entidad; existen dos tipos de usuarios: usuario administrador es el que tiene acceso completo a la aplicación, usuario simple solo tiene acceso al control de labores agrícolas, estimación, producción, reportes.

Una vez realizada la alimentación de la información a la aplicación, se pudo contar con una guía operativa sumamente confiable que contendrá todas las actividades de manejo agrícola que se realice en cada finca perteneciente ha APROBAVEN.

La aplicación fue desarrollada completamente en entorno JAVA utilizando las herramientas NetBeans IDE 7.3, Mysql 5.0 y Apache TOMCAT 7.0

ABSTRACT

The need for a web application to control the work of banana crop for the Association of Banana Producers Windows "APROBAVEN" designed to the standards of the aforementioned entity.

Requirements analysis is made it through information provided by the directive APROBAVEN, information was compiled by conducting surveys to people involved in the banana slang.

Raised as main objective to develop software for the effective control of the operations performed in APROBAVEN , which manages the handling of agricultural work, the application supports recording work with their respective parameters, making estimation of production, access to the system only do members of the entity , there are two types of users: administrator is the user who has full access to the application , simple user only has access to the control of agricultural work , estimating, production reports.

Once the supply of information to the application , it could have a highly reliable operational guide to contain all farm management activities you perform in each farm belonging has APROBAVEN .

The application was completely developed in JAVA environment using the tools NetBeans IDE 7.3 , MySql 5.0 and Apache TOMCAT 7.0

CAPÍTULO I

MARCO CONTEXTUAL DE LA INVESTIGACIÓN

1.1 INTRODUCCIÓN

En la actualidad Ecuador es el primer exportador a nivel mundial de banano, proveyendo su producto con los estándares de calidad que establecen la Unión Europea, los países de América Del Norte y Los Asiáticos (AEBE, 2012).

Adicionalmente según datos de la Subsecretaria de Agricultura, hay en el país alrededor de 230.000 hectáreas sembrada con banano, de las cuales solamente hay inscrita un 75%; es decir, 170.896,34 hectáreas y un 25% no inscritas. Los mercados para el Banano Ecuatoriano están distribuidos de la siguiente manera:

Tabla 1 Estadística de banano a nivel mundial

Mercados	Porcentaje
EE.UU	22 %
Unión Europea	26 %
Rusia	18 %
Cono Sur	6 %
Otros Mercados	28 %

Nuestros principales competidores son Colombia con un 30% en los Estados Unidos y un 70% en la Unión Europea y Costa Rica con un 50% en los Estados Unidos, 40% en la Unión Europea y 10% en otros.

En base a la situación actual del sector bananero en nuestro país, ha mejorado notoriamente con las gestiones hechas por el Estado Ecuatoriano; es necesario, que la Asociación de productores Bananeros de Ventanas, tenga a su disposición una plataforma informática de primer nivel para estar conectado a nivel mundial con la cadena de producción del primer producto agrícola de exportación del país.

La necesidad de contar con una aplicación informática que ayude a tomar decisiones de una manera eficiente y contar con respaldo disponible en cualquier momento para mejorar el nivel de exportación de la Asociación, es necesario que la asociación cuente con esta tecnología con el objetivo de conocer a la brevedad posible la producción de cada una de las haciendas para, de acuerdo a ello, hacer las negociaciones con las compañías exportadoras y fijar precios que vayan a satisfacer las necesidades de los agricultores agremiados en diferentes épocas del año. Con la Aplicación Informática en funcionamiento la asociación tendrá información individual de cada una de las haciendas perteneciente a dicha entidad y mantendrá nexos de comunicación y comercialización más eficiente con todas las compañías exportadoras e importadoras a nivel mundial de nuestro banano.

1.2 JUSTIFICACIÓN

La LEY ORGÁNICA DE EDUCACIÓN SUPERIOR (LOES) en su Capítulo 2 FINES DE LA EDUCACION SUPERIOR; Artículo 8 Serán Fines de la Educación Superior; Literal C dispone que: Aportar con el cumplimiento de los objetivos del régimen del desarrollo previsto en la Constitución y en el Plan Nacional de Desarrollo.

El Plan Nacional de Desarrollo en su eje de Transformación Económico – Productivo uno de sus objetivos principales es el Desarrollo Rural, con miras a eliminar la brecha urbano – rural y fomentar la soberanía alimentaria: agricultura, ganadería y pesca.

Se propone crear una guía operativa para la gestión y control de la producción y comercialización de Banano en APROBAVEN; la información de producción de banano debe estar siempre disponible y actualizada en base a los datos que se le suministra al sistema. El aplicativo web se lo diseñó para implementar el control de las labores que se realizan en la producción del cultivo de banano, se sugiere que siempre se este suministrando información al software.

Con las Artículos anteriormente citados de la LOES y el Plan Nacional de Desarrollo este trabajo queda plenamente justificado.

1.3 PROBLEMATIZACIÓN

Dentro de las funciones que cumple la Asociación de Productores Bananeros de Ventanas “APROBAVEN” consta: Reunir a todos los pequeños productores de la zona tratando de asociarlos, para de esa manera su producción semanal o mensual sean vendidos o tengan la comercialización externa segura por medio de la asociación. Además, luchar porque se cumpla el precio oficial y que sus socios no sean presa fácil de las compañías exportadoras o de algún intermediario inescrupuloso o “cuperos” como se los llama dentro del argot bananero.

Se mantiene vínculos de amistad y negocio a gran escala con las compañías exportadoras para que sus asociados cuenten con el cupo semanal y por ende obtengan los recursos económicos para satisfacer las necesidades de las fincas y de sus trabajadores. Logra salvaguardar y hacer respetar el precio oficial por medio de contratos durante las 52 semanas del año y que los pequeños productores y en particular los medianos siempre se mantengan asociados, y no solamente en épocas en que los precios están deprimidos busquen la asociación, mientras las épocas de bonanza se mantienen al margen del gremio.

Actualmente APROBAVEN no cuenta con la información en tiempo real de las haciendas perteneciente a dicha entidad; por ende, dificulta los consensos y decisiones para las negociaciones que benefician a los miembros de la agrupación. Existen dificultades a nivel de la Asociación en lo referente al control de todas las labores que se realizan en cada una de las hacienda y por lo tanto la producción de cada una de ellas varía semana a semana y la Asociación lo desconoce. No se cuenta con un patrón general para la realización y control de las siembras, realización de labores agrícolas, cosecha, empaque, transporte y comercialización a nivel de toda la asociación con el fin de mejorar la producción y su infraestructura; pues su gran mayoría carecen de un registro que les permita determinar la rentabilidad y

sostenibilidad de sus fincas en producción. La Entidad establece de las ventas un margen de ganancia a todos los socios productores.

1.4 OBJETIVOS

1.4.1 GENERAL

Desarrollar un software para el control efectivo de las operaciones que se realizan en la Asociación de Productores de Banano del Cantón Ventanas “APROBAVEN”.

1.4.2 ESPECÍFICOS

- Identificar las actividades que se lleva a cabo en la entidad.
- Establecer procesos que permitan realizar dicha actividad de una manera formal, ordenada y eficiente.
- Gestionar los parámetros generales para el manejo y cosecha del cultivo dentro de las fincas pertenecientes a la Asociación.
- Generar reportes de producción que permita la toma de decisiones a nivel gerencial.

1.5 HIPÓTESIS

1.5.1 GENERAL

Una aplicación web especializada en el cultivo de banano aumentará la eficiencia en el manejo de labores de producción de banano.

1.5.2 VARIABLES

1.5.2.1 Variable Independiente

Tabla 2 Matriz de Conceptualización y Operacionalización de Variable Independiente

Matriz de Conceptualización y Operacionalización de Variable			
Variable	Definición	Dimensiones	Indicadores
Aplicación web especializada en cultivo de banano	Contiene elementos que permita una comunicación entre el usuario y la información, apropiada para el cultivo de Banano	Seguimientos de labores	- Eficiencia en resultados
		Generación de reportes	Reporte de planificación. Reportes de producción. Reporte de Estimación

1.5.2.2 Variable Dependiente

Tabla 3 Matriz de Conceptualización y Operacionalización de Variable Dependiente

Matriz de Conceptualización y Operacionalización de Variable			
Variable	Definición	Dimensiones	Indicadores
Eficiencia en el manejo de labores de producción de banano	Es el logro de las metas con la menor cantidad de recursos.	Fiabilidad	Tasa de pérdida de documento Manejo de labores de producción de banano.

CAPÍTULO II

MARCO TEÓRICO

2.1 FUNDAMENTACIÓN TEÓRICA

2.1. INFORMACIÓN BÁSICA

2.1.1 Guía Operativa

Una guía operativa es una herramienta analítica que tiene con fin facilitar las pautas y las directrices para el desarrollo de las acciones para poner en marcha un proyecto.

El banano Cavendish es el de mayor consumo a nivel mundial. Es originario de Vietnam y China, y tiene entre 15 y 25 cm. La piel es verde cuando se vende en los mercados, y luego se vuelve amarilla cuando madura. En el proceso de maduración se producen los azúcares y los aromas características del banano (Castle Food, 2012).

Presentación:

Tabla 4 Características del Banano

Variedad	Banano Cavendish
Clase	"A" Premium
Tamaño de los dedos	20 cm mínimo
Calibre	Min. 39 mm, Max. 46mm
Número de dedos por mano	Min. 5 hasta 12 dedos
Edad de la fruta	Min. 10 semanas Max. 12 semanas
Cajas	22XU
Peso	41 lbs – 43 lbs

2.1.2 Asociación De Productores Bananero De Ventanas (Aprobaven)

Fundada en el 2006, está constituida por 18 pequeños y medianos productores de banano, se encuentra cerca de la ciudad de Ventanas, cantón que está ubicado en el centro de la provincia de Los Ríos.

La industria bananera genera fuentes de trabajo directo e indirecto de 2.5 millones de personas en la región Costa de nuestro país. Por esta razón se debe comercializar directamente con las principales cadenas de: maduradores, supermercados, autoservicios y tiendas de frutas, para lograr que los consumidores obtengan fruta de excelente calidad a un mejor precio.

El aplicativo web brindara información específica de las etapas de producción del cultivo de banano, el software emitira reportes para que los dueños o administradores de la plantación de banano conozcan en que estado se encuentra.

Misión.- Somos una organización de productores de la Provincia de los Ríos dedicados a satisfacer las exigencias de nuestros clientes con alta calidad, bajo el sistema de comercio justo buscando el mejoramiento de las condiciones de los socios y el mejoramiento de nuestro entorno en términos sociales, ambientales y laborales.

Visión.- Hacer de APROBAVEN una estructura colectiva y sólida, con miembros decididos a competir en un mercado que exige calidad y capacidad de producción.

2.1.3 Labores Agrícolas.

Con las labores agrícolas pretendemos el desarrollo óptimo de las plantas, mejorar la calidad y la cantidad de las cosechas. Realizamos acciones mecánicas para mejorar las condiciones del cultivo, respetando el medio ambiente (Macape, 2009). Tomaremos establecer el tipo de labor, que tiene

como objeto arreglar, airear, facilitar la incorporación del agua, alimentar, eliminar las plantas que comprometerían el desarrollo de las cultivadas, favorecer el drenaje.

2.1.4 Calidad De La Fruta.- La calidad de la fruta principalmente es su presentación, establecida en lo que es el grado de corte, el ratio, la merma, numero de manos, longitud de dedos.

2.1.5 Cosecha.- Se denomina cosecha a la labor de recolección de racimos la cual depende de su edad y calibración exigidas por el tipo de embarque. La adecuada programación de cosecha considera los inventarios de fruta por lote y las áreas pendientes por recorrer permitiendo optimizar el aprovechamiento de los racimos en la empacadora (Palacio, 2009).

2.1.6 Manejo Agrícola.- Coordinar y optimizar los recursos para lograr los objetivos de una forma más eficiente.

2.1.7 Hirbernate.

Es una capa de persistencia objeto/relacional y un generador de sentencias sql. Te permite diseñar objetos persistentes que podrán incluir polimorfismo, relaciones, colecciones, y un gran número de tipos de datos. De una manera muy rápida y optimizada podremos generar Base de Datos (BD) en cualquiera de los entornos soportados: Oracle, DB2, MySql, etc. Y lo más importante de todo, es open source, lo que supone, entre otras cosas, que no tenemos que pagar anda por adquirirlo. Uno de los posibles procesos de desarrollo consiste en, una vez tengamos el diseño de datos realizados, mapear esto a ficheros XML (Lenguaje de marcas extensibles XML) siguiendo la DTD (Definición Tipo Documento) de mapeo de Hibernate.

Hibernate se integra en cualquier tipo de aplicación justo por encima del contenedor de datos (Suarez, 2003).

Si el lenguaje es orientado a objetos, es mucho más fácil diseñar todo el modelo de dominio y la aplicación orientando a objetos, sin tener en ningún momento que depender del tipo de modelo de datos a persistir (modelo entidad relación) y del lenguaje de consultas a utilizar (SQL). Aquí parece el concepto de las herramientas de mapeo objeto-relacional (ORM o Object-Relational Mapping), que nos permitirían persistir un modelo de dominio orientada a objetos sin tener la necesidad de utilizar directamente, JDBC, ni SQL; es decir, realizaran una transformación del modelo de dominio orientado a objetos, a un modelo de datos relacional de forma totalmente transparente para el desarrollador. A las aplicaciones de tipo ORM, se las conoce como motores ORM. Hibernate es un motor ORM que cumple también con la especificación de persistencia JPA.




Las principales características de Hibernate son:

- + Es un framework open source, y es el ORM más utilizado en el mercado laboral.
- + Independiza a la aplicación del motor de la base de datos a utilizar: Oracle, MySQL, Derby, DB2, MSSQLServer...etc.
- + Se adapta a cualquier modelo de datos ya existente.
- + Inicialización de objetos bajo demanda (carga perezosa).
- + Es capaz de generar el modelo de datos a partir del modelo de objetos.
- + Posee su propio lenguaje de consultas orientadas a objetos llamado HQL (Hibernate Query Language).
- + Contiene una API para realizar consultas orientadas a objetos llamado Criteria, que será incluido en la especificación JPA 2.0 dentro de Java EE6.
- + Implementa la especificación JPA incluida dentro de EJB3.

2.1.7.1 Objetos Persistentes

La adopción de Hibernate como motor ORM en la gran mayoría de empresas, viene determinada como consecuencia de lo simple que resulta definir nuestro modelo de dominio persistente, solo será necesario indicar que objetos planos (Pojo) queremos persistir, y crear bien un mapeo en documento XML (HBM) o utilizando anotaciones JPA.





Para crear un modelo de dominio persistente tenemos varias opciones:

-  **El modelo de datos objeto relacional ya existe:** en este caso, deberemos crear nuestros objetos a partir de dicho modelo de datos. Independientemente de la nomenclatura del modelo existente, generaremos nuestros objetos y atributos según el diseño técnico, y realizaremos el mapeo de esos objetos y atributos con la nomenclatura existente en el modelo relacional. Podremos generar los objetos persistente de forma manual, o automáticamente utilizando las utilidades de Hibernate (Hibernate Tools).
-  **El modelo de datos objeto relacional no existe:** este caso nos permitirá diseñar nuestra aplicación sin ningún tipo de limitación, es decir, realizaremos el diseño orientado a objetos, y definiremos cual es nuestro modelo de dominio persistente. Hibernate permite generar el modelo de datos relacional a partir de los objetos del modelo de dominio, independientemente del motor de base de datos a utilizar.
-  **El modelo de datos objeto relacional existe en parte pero hay que ampliarlo:** En este caso tendremos que generar nuestro modelo de dominio persistente, primero teniendo en cuenta la base de datos relacional y posteriormente podremos utilizar Hibernate para generar el modelo de datos relacional, completo, o parcial, a partir de nuestro modelo de dominio persistente.

2.1.7.2 Mapeo De Objetos

Una vez que tenemos definido el modelo de dominio persistente, tan solo tendremos que crear los ficheros de mapeo en XML de Hibernate HBM (Hibernate Mapping File).

Dentro de los ficheros de mapeo debemos indicar:

-  La relación de la clase con la tabla del modelo entidad-relacion.
-  El identificador del objeto que corresponderá con la clave de la tabla (primary key).
-  El mapeo de atributos del objeto con los campos de la tabla.
-  El mapeo de los objetos relacionados existente dentro del propio objeto (grafo de objetos).

2.1.8 Javabeans

La tecnología EJB, al igual que las tecnologías JSP y Servlet, se integran en la plataforma Java Enterprise Edition (Java EE). Sun Microsystems, mediante los componentes EJB apuesta por la incorporación de Java en el lado del Servidor. La tecnología EJB 3.0 está incluida dentro de las especificaciones JSR 220.

2.1.9 Servlets y Jsp.

Los servlets y las paginas JSP son aplicaciones que se ejecutan en un servidor Web. Estas aplicaciones se quedan esperando peticiones de los clientes. Cuando un cliente, como por ejemplo un Applet, realiza una petición a un Servlets, este genera el código oportuno y retorna la respuesta de la petición al Applet.

Los Servlets y las páginas JSP nos permitirán generar respuesta dinámicas, es decir; con estas dos tecnologías podremos generar contenido dinámico en nuestras aplicaciones Web.

2.1.9.1 Característica de los Servlets.

Las principales características de los servlets:

- ✚ Debido a que son 100 por ciento puro Java, los independizan totalmente de la plataforma.
- ✚ Descargan el servidor Web en gran medida, ya que en la primera petición, el servidor Web crea un proceso para el Servlets y, en posteriores peticiones, crea hilos (threads) de ejecución de este proceso.
- ✚ Son muchos más rápidos que los CGI, ya que están precompilados y, debido al segundo punto, no se generan procesos independientes cada vez que se solicita un URL (como pasa en CGI).
- ✚ Son más seguros, ya que utilizan el API de seguridad de Java.
- ✚ La comunicación con otros Servlets es muy sencilla.
- ✚ Mediante el control de excepciones pueden tratarse de una forma bastante sencilla todos los errores que puedan ocurrir durante la ejecución del Servlets.
- ✚ Etc.

Hay que tener en cuenta una cosa: los Applets se ejecutan en el lado del cliente, más bien en el navegador del cliente; sin embargo, los Servlets se ejecutan en el lado del servidor, con lo cual un cliente no tiene por qué tener soporte para Java (el navegador no tiene que ser compatible con Java), ya que se puede generar; por ejemplo, una página HTML como respuesta a una petición de un cliente (Lima, 2010).

2.1.10 Patrón Modelo Vista Controlador

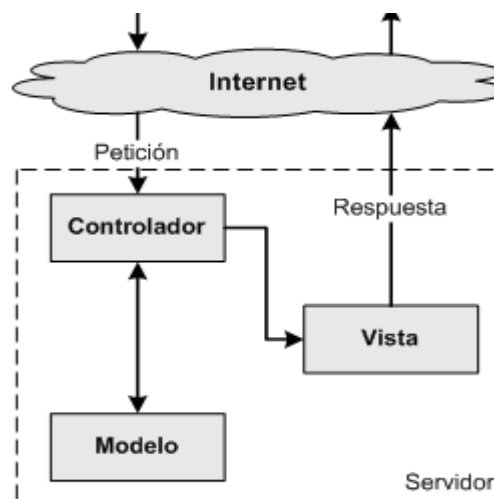
El patrón Modelo – Vista – Controlador fue inventado en el contexto de Smalltalk para realizar una separación entre la interfaz gráfica y el código del funcionamiento de una aplicación. Esta idea teórica afectó, de forma

importante, a gran parte del código de Smalltalk y fue posteriormente aplicada a los lenguajes que están basados en objetos.

En el paradigma MVC, las entradas del usuario, los modelos del mundo exterior y la retroalimentación visual son explícitamente separados y manejados por tres tipos de objetos, cada uno especializado para un conjunto de tareas específicas.

El objetivo primordial del patrón es dar soporte a los modelos funcionales y mapas mentales de la información relevante para los usuarios, permitiendo un modelo que facilite la consulta y manejo de los mismos. La única manera de construir artefactos manejables es ayudar al usuario a construir modelos del sistema. Pero esto es imposible si el modelo mental no ha sido diseñado dentro del artefacto desde el principio. Intentar adicionar los modelos mentales del usuario cuando ya se ha avanzado en el desarrollo puede ser imposible. A continuación un gráfico que resume el patrón.

Ilustración 1 Modelo Vista Controlador



2.1.10.1 Ventajas y Desventajas del uso Del Patrón

Se tienen muchas ventajas como:

- ✚ La implementación se realiza de forma modular.
- ✚ Sus vistas muestran información actualizada siempre. El programador no debe preocuparse de solicitar que las vistas se actualicen, ya que este proceso es realizado automáticamente por el modelo de la aplicación.
- ✚ Cualquier modificación que afecte al dominio, como aumentar métodos o datos contenidos, implica una modificación sólo en el modelo y las interfaces del mismo con las vistas, no todo el mecanismo de comunicación y de actualización entre modelos.
- ✚ Las modificaciones a las vistas no afectan al modelo de dominio, simplemente se modifica la representación de la información, no su tratamiento.
- ✚ MVC está demostrando ser un patrón de diseño bien elaborado pues las aplicaciones que lo implementan presentan una extensibilidad y una mantenibilidad únicas comparadas con otras aplicaciones basadas en otros patrones.

Como desventajas tenemos:

- ✚ Para desarrollar una aplicación bajo el patrón de diseño MVC es necesario una mayor dedicación en los tiempos iniciales del desarrollo. Normalmente el patrón exige al programador desarrollar un mayor número de clases que, en otros entornos de desarrollo, no son necesarias. Sin embargo, esta desventaja es muy relativa ya que posteriormente, en la etapa de mantenimiento de la aplicación, una aplicación MVC es mucho más mantenible, extensible y modificable que una aplicación que no lo implementa.
- ✚ MVC requiere la existencia de una arquitectura inicial sobre la que se deben construir clases e interfaces para modificar y comunicar los

módulos de una aplicación. Esta arquitectura inicial debe incluir, por lo menos, un mecanismo de eventos para poder proporcionar las notificaciones que genera el modelo de aplicación; una clase Modelo, otra clase Vista y una clase Controlador genéricas que realicen todas las tareas de comunicación, notificación y actualización que serán luego transparentes para el desarrollo de la aplicación.

- ✚ MVC es un patrón de diseño orientado a objetos por lo que su implementación es sumamente costosa y difícil en lenguajes que no siguen este paradigma.

2.1.10.2 Modelo

El modelo es un conjunto de clases que representan la información del mundo real que el sistema debe reflejar. Es la parte encargada de representar la lógica de negocio de una aplicación. Así, por ejemplo, un sistema de administración de datos geográficos tendrá un modelo que representara la altura, coordenadas de posición, distancia, etc. sin tomar en cuenta ni la forma en la que esa información va a ser mostrada ni los mecanismos que hacen que esos datos estén dentro del modelo, es decir, sin tener relación con ninguna otra entidad dentro de la aplicación.

El modelo, a nivel teórico, no debe tener conocimiento acerca de la existencia de las vistas y del controlador. Esta situación es interesante, pero de difícil aplicación práctica, pues deben existir interfaces que permitan a los módulos comunicarse entre sí, por lo que SmallTalk sugiere que el modelo en realidad esté formado por dos submódulos: El modelo del dominio y el modelo de la aplicación.

2.1.10.2.1 Modelo de Dominio

Se entiende por modelo de dominio al conjunto de clases definidas a través del análisis de la situación real del problema que queremos solucionar. Si seguimos el ejemplo anterior, sobre datos geográficos, pertenecerían a este

dominio las clases; Río, Montaña, Mar, etc...El modelo del dominio no debería tener relación con nada externo a la información que contiene.

2.1.10.2.2 Modelo de la Aplicación

El modelo de la aplicación es un conjunto de clases que sirven de puente en la relación de las vistas con el modelo de dominio. Tienen conocimiento de las vistas e implementan los mecanismos necesarios para notificar a éstas los cambios que se pudieren dar en el modelo del dominio. El modelo de la aplicación es llamado también coordinador de la aplicación.

2.1.10.3 Vista

Las vistas son las encargadas de la representación de los datos, contenidos en el modelo, al usuario. La relación entre las vistas y el modelo son de muchas a uno, es decir cada vista se asocia a un modelo, pero pueden existir muchas vistas asociadas al mismo modelo. De esta manera, por ejemplo, se puede tener una vista mostrando la hora del sistema como un reloj analógico y otra vista mostrando la misma información como un reloj digital.

La vista solo necesita la información requerida del modelo para realizar un despliegue. Cada vez que se realiza una actuación, que implica una modificación del modelo de dominio, la vista cambia a través de notificaciones generadas por el modelo de la aplicación. Sencillamente, es la representación visual del modelo que redibuja las partes necesarias cuando se produce una modificación del mismo.



2.1.10.4 Controlador

El controlador es el encargado de interpretar y dar sentido a las instrucciones que realiza el usuario, realizando actuaciones sobre el modelo. Si se realiza algún cambio, comienza a actuar, tanto si la modificación se produce en una

vista o en el modelo. Interactúa con el Modelo a través de una referencia al propio Modelo.

2.1.10.5 El Patrón MVC y el Lenguaje Java

El lenguaje de programación Java proporciona soporte para la arquitectura MVC. Provee de dos clases que son las encargadas de realizar las notificaciones de cambios en los estados de los objetos. Se definen a continuación los objetos:

-  **Observer:** Es cualquier objeto que desee ser notificado cuando el estado de otro objeto sea alterado.
-  **Observable:** Es cualquier objeto cuyo estado puede representar interés y sobre el cual otro objeto ha demostrado ese interés.

Estas dos clases no sólo se utilizan en la aplicación del patrón MVC, tienen una utilidad mayor dentro del lenguaje. Serán útiles en cualquier sistema en el que se necesite que algunos objetos sean notificados cuando ocurran cambios en otros objetos.

El Modelo es un subtipo de Observable y la Vista es un subtipo de Observer. Estas dos clases manejan adecuadamente la función de notificación de cambios que necesita la arquitectura MVC. Proporcionan el mecanismo por el cual las Vistas pueden ser notificadas automáticamente de los cambios producidos en el Modelo. Referencias al objeto Modelo tanto en el Controlador como en la Vista permiten acceder a los datos de ese objeto Modelo

2.1.10.6 ¿Cómo Funciona una Aplicación MVC?

2.1.10.6.1 Captura de la Petición en el Controlador.

La aplicación recibe peticiones que son centralizadas en el Controlador. Éste es el encargado de interpretar, a partir de la URL de la solicitud, el tipo de

operación que hay que realizar. Normalmente, esto se hace analizando el valor de algún parámetro que se envía anexando a la URL de la petición y que se utiliza con esta finalidad.

2.1.10.6.2 Procesamiento de la Petición

Una vez que el Controlador determine la operación a realizar, procede a ejecutar las acciones pertinentes, invocando para ello a los diferentes métodos expuestos por el Modelo.

Dependiendo de las acciones a realizar (por ejemplo, un alta de un usuario en el sistema), el Modelo necesitará manejar los datos enviados por el cliente en la petición, datos que le serán proporcionados por el controlador. De la misma manera, los resultados generados por el Modelo (por ejemplo la información resultante de una búsqueda serán entregados directamente al controlador).

Para facilitar este intercambio de datos entre el Controlador y Modelo y, posteriormente, entre Controlador y Vista, las aplicaciones MVC suelen hacer uso de JavaBeans. Un JavaBean no es más que una clase que encapsula un conjunto de datos con métodos de tipo set/get para proporcionar un acceso a los mismos desde el exterior.

2.1.10.6.3 Generación de Respuestas

Los resultados devueltos por el Modelo al Controlador son depositados por éste en una variable de petición, sesión o aplicación, según el alcance que deban tener. A continuación, el Controlador invoca a la página JSP que debe encargarse de generar la vista correspondiente, esta página accederá a la variable de ámbito donde estén depositados los resultados y los utilizará para generar dinámicamente la respuesta XHTML que será enviada al cliente (Andalucía, 2013).

2.1.11 Java

Java es un lenguaje de programación orientado a objetos que fue creado por la compañía Sun Microsystems en 1995. El origen del nombre Java no está claro, aunque se atribuye a un tipo de café que estaba disponible a una cafetería cercana al lugar de trabajo del equipo que lo desarrolló. El nombre original de Java era Oak, pero que tuvo que ser cambiado por que el nombre ya estaba registrado por otra compañía. La plataforma se puede descargar de forma gratuita y hoy en día Java se ejecuta en cientos de millones de computadoras personales de todo el mundo y en miles de millones de dispositivos, como lo son los dispositivos móviles, aparatos de televisión, aparatos especializados y consolas de juegos.

Cuando descargas el software de Java obtienes varios componentes: el entorno de tiempo de ejecución (JRE -enlace en inglés-), que consiste de la máquina virtual de Java (JVM -enlace en inglés-), las clases centrales de la plataforma Java y bibliotecas de apoyo de la plataforma Java. El JRE es la parte relacionada con la ejecución del software de Java y es lo único necesario para ejecutar Java en un navegador Web, permitiéndote ejecutar los applets (componentes de una aplicación que se ejecutan dentro el contexto de otro programa) escritos en este lenguaje de programación.

El complemento Java es un componente del entorno de ejecución de Java (JRE). El complemento Java no es un programa independiente de la plataforma y no se puede instalar por separado.

Java Platform, Micro Edition (Java ME) ofrece un entorno para aplicaciones de dispositivos como son: teléfonos móviles, TDT, reproductores Blu-ray, dispositivos multimedia digitales, módulos M2M e impresoras, por mencionar algunos.

La tecnología Java ME se creó originalmente para atenuar las limitaciones asociadas a la creación de aplicaciones para dispositivos pequeños (Castro, 2014).

2.2 MARCO REFERENCIAL

2.2.1 Análisis y diseño e implementación de una guía operativa de cultivo para productos de exportación.

El objetivo de este trabajo es realizar el análisis para una agroindustria productora de plátano “Barraganete” de exportación, que permita la creación de una aplicación informática que garantice la gestión efectiva de los distintos procesos y subprocesos asegurando la rentabilidad empresarial (Bozada, 2010).

Este trabajo fue desarrollado para la Escuela Politécnica del Ejército por Armando Wilson Bozada Cabrera.

Este proyecto es una guía colaborativa para este trabajo de investigación ya que el propósito de las dos tesis es la creación y desarrollo de una guía operativa con la diferencia que en la aplicación que se realizará para la Asociación de Productores Bananeros de Ventanas “APROBAVEN” es exclusivamente para el manejo agrícola de la producción de banano.

CAPÍTULO III

METODOLOGÍA DE LA INVESTIGACIÓN

3.1 MATERIALES Y MÉTODOS

3.1.1 Materiales

3.1.1.1 Hardware

Laptop HP ProBook 4530s

- Disco duro de 500 Gb
- Procesador Intel CORE™ I3 2310M 2.10 GHz
- Memoria RAM 4 Gb
- Impresora Canon PX
- Pendrive 10 Gb

3.1.1.2 Software

Sistema Operativo

- Windows 7 Home premium

Lenguaje de Programación

- Netbean Ide for Java
- Apache Web Server

Software Multimedia y Diseño

- Dreamweaver CS6
- Adobe Photoshop CS6

Digitalización

- Microsoft Office 2010

Base de Datos

- MYSQL Database

Diagramación

- StarUml

Utilitarios

- Microsoft Project 2010

3.1.1.3 Suministros

- Resma de Papel A4
- Cartuchos de Tinta
- Cds
- Varios

3.1.1.4 Personal

El presente trabajo es desarrollado por César Fabián Varas Beltrán, el mismo que se encargó de la recopilación de información, documentación y cumplir con las fases de análisis, diseño, implementación y pruebas.

Desarrollador: César Fabián Varas Beltrán

Tutor: Msc. Ariosto Vicuña

3.2 MÉTODOS

3.2.1 Modelo Incremental.

Es un modelo de tipo evolutivo que está fundamentado en varios ciclos cascada realimentados utilizados repetitivamente, con una filosofía interactiva (Tecnológico, 2011).

Los principios básicos son:

- Una serie de mini-cascada se llevan a cabo, donde todas las fases de la cascada modelo desarrollo se han completado para una pequeña parte de los sistemas, antes de proceder a la próxima incremental, o
- Se definen los requisitos antes de proceder con lo evolutivo, se realiza un mini-cascada de desarrollo de cada uno de los incrementos del sistema, o
- El concepto inicial de software, análisis de las necesidades, y el diseño de la arquitectura y colectiva básicas se definen utilizando el enfoque de cascada, seguida por lo iterativo de prototipos, que culmina con la instalación del prototipo final.

3.2.1.1 Fases del Ciclo

El ciclo de vida consiste en una serie de fases que en su conjunto conducen al sistema final. Cada etapa está compuesta por un número de interacciones que generan versiones del sistema. Las fases que lo componen son:





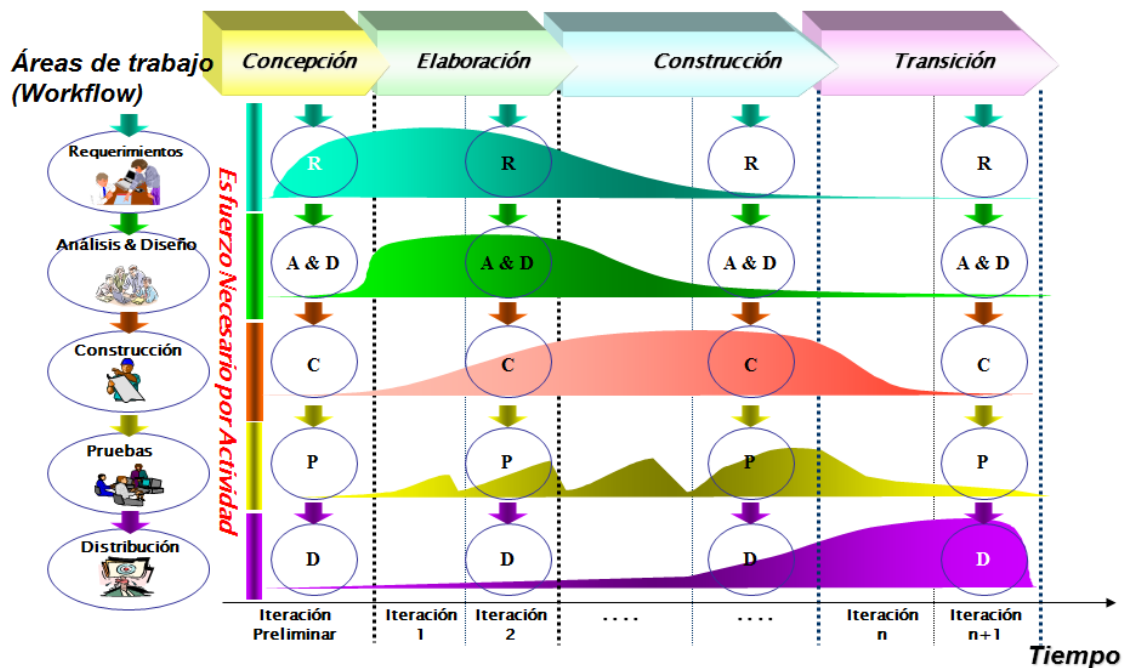
-  Concepción, Inicio o Estudio de Oportunidad.
-  Elaboración.
-  Construcción.
-  Transición (Gonzales).

Ilustración 2 Diseño Incremental







En la etapa de requerimientos se identificaron las actividades que lleva a cabo la entidad para realizar las actividades de una manera formal, ordenada y eficiente.

Se estandarizó que un racimo enfundado será estimado como una caja de producción de banano.

3.2.2 Método de Desarrollo del Sitio Web

Para el desarrollo de este trabajo se empleó la Metodología RUP (Rational Unified Procces) es una secuencia de pasos necesarios para el desarrollo y/o mantenimiento de gran cantidad de sistemas, en diferentes áreas de aplicación diferentes organizaciones, diferentes medios de competencia y en proyectos de tamaños variables (desde el más básico al más complejo). Actualmente es propiedad de International Business Machines (IBM) y está basado en un enfoque disciplinado de asignación de tareas y responsabilidades dentro de una organización de desarrollo con la finalidad de asegurar la obtención de un software de alta calidad que satisfagan la necesidad de los usuarios finales dentro de un calendario y tiempo predecible.

3.2.2.1 Elementos del RUP.

-  Disciplinas: son los contenedores empleados para organizar todas las actividades durante el ciclo de vida de sistema.
-  Artefactos: son los elementos de entrada y salida de las actividades. Es un elemento que el proyecto produce y utiliza para componer el producto final.
-  Flujos de trabajo: constituye la secuencia de actividades que producen resultados visible por medio de la integración de roles y las actividades, artefacto y disciplinas.
-  Roles: son las personas o entes que están involucrados en cada proceso (Equipo, 2012).

3.2.2.2 Fases de diseño de metodología RUP.

Consististe en determinar la solución técnica y necesaria del proyecto a ejecutar. Así como durante la fase de inicio se determinó el qué, ahora es necesario el cómo. Es esta fase durante la cual elaboramos los requisitos al nivel del diseño y por tanto, nos pone en posición de saber si el proyecto es técnicamente viable así como conocer la tecnología que vamos a utilizar durante la construcción. En esta fase es donde se procede al desarrollo del prototipo del sistema, por lo tanto es necesario la participación activa de los usuarios involucrados en el proyecto.

En esta fase del diseño es importante tomar en cuenta los niveles de requerimientos de información, específicamente: el nivel Operativo, el nivel Administrativo, el nivel Gerencial y el nivel Legal (Rodríguez, 2012).

3.2.2.3 Etapas del RUP.

3.2.2.3.1 Inicio.

La especificación de los requerimientos de Salidas, que consiste en el prototipo de los reportes, pantallas de consulta, reportes, salidas a archivos, etc. Comprende un modelado lo más preciso posible a lo que se implementará.

La especificación de los requerimientos de Entradas, que consiste en el prototipo de las interfaces o pantallas de actualización de datos. Se debe contemplar aspectos propios de la implementación, como ser: ubicación de menú, ubicación de botones de opciones, barra de acceso rápido, diseño de íconos, uso de colores corporativos, tipo de letra, tamaño de letra, etc.

3.2.2.3.2 Elaboración

Aquí en esta fase de desarrollo, es en donde se construye los diagramas de caso de uso, actividades, secuencia, clases.

3.2.2.3.3 Construcción.

En esta fase es donde pasamos a moldear el producto que vamos a mostrar al usuario con todos sus requerimientos, además de que en esta etapa se realiza la implantación y las pruebas al mismo.

En esta etapa usamos la arquitectura MVC Modelo Vista Controlador. Es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página, el modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio y el controlador es el responsable de recibir los eventos de entrada desde la vista.

3.2.2.3.1 ¿Cuál es la finalidad del Modelo Vista Controlador?

La finalidad del modelo es mejorar la reusabilidad por medio del desacople entre la vista y el modelo. Los elementos del patrón son los siguientes:

El modelo es el responsable de:

- Acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento.
- Define las reglas de negocio (la funcionalidad del sistema). Un ejemplo de regla puede ser: “Si la mercancía pedida no está en el almacén, consultar el tiempo de entrega estándar del proveedor”.
- Lleva un registro de las vistas y controladores del sistema.
- Si estamos ante un modelo activo, notificará a las vistas los cambios que en los datos pueda producir un agente externo (por ejemplo, un fichero bath que actualiza los datos, un temporizador que desencadena una inserción, etc).

El controlador es el responsable de:

- Recibe los eventos de entrada (un clic, un cambio en un campo de texto, etc.).
- Contiene reglas de gestión de eventos, del tipo “Si Evento Z, entonces Acción W”. Estas acciones pueden suponer peticiones al modelo o a las vistas. Una de estas peticiones a las vistas puede ser una llamada al método “Actualizar()”. Una petición al modelo puede ser “Obtener_tiempo_de_entrega(nueva_orden_de_venta)”.

Las vistas son responsables de:

- Recibir datos del modelo y los muestra al usuario.
- Tienen un registro de su controlador asociado (normalmente porque además lo instancia).
- Pueden dar el servicio de “Actualización()”, para que sea invocado por el controlador o por el modelo (cuando es un modelo activo que informa de los cambios en los datos producidos por otros agentes).

3.2.2.3.3.2 ¿Qué Ventajas trae utilizar el MVC?

- ✚ Es posible tener diferentes vistas para un mismo modelo (eg. representación de un conjunto de datos como una tabla o como un diagrama de barras).
- ✚ Es posible construir nuevas vistas sin necesidad de modificar el modelo subyacente.
- ✚ Proporciona un mecanismo de configuración a componentes complejos muchos más tratable que el puramente basado en eventos (el modelo puede verse como una representación estructurada del estado de la interacción) (Sebastián, 2010).

3.2.2.3.4 Transición.

Durante esta fase de transición busca garantizar que se tiene un producto preparado para su entrega al usuario.

3.2.3 Metodología de Comprobación de Hipótesis

Después de terminar el desarrollo de la aplicación web, se procede a demostrar la hipótesis de la investigación de la siguiente forma:

Para comprobar cuál de las 2 hipótesis planteadas se ejecuta, se diseñará un experimento en el cual se tomaran datos reales de una determinada hacienda perteneciente a APROBAVEN; se realizará 2 obtenciones de datos O_1 y O_2 .

O_1 estos datos serán tomados antes de la implementación de la aplicación web. O_2 La segunda obtención de datos se realizara después de la implementación de la aplicación web.

Esquema del Diseño:

Grupo Experimental

O_1 X O_2

Una vez que se definió el diseño de esta investigación, los métodos de obtención de datos que serán utilizados se realizarán de acuerdo a los indicadores ya establecidos con anterioridad.

El método que se utilizará para la verificación de la hipótesis será. La prueba t de Student y el nivel de significancia será $\alpha = 0,05$. Para el análisis de los indicadores se utilizarán las técnicas de observación, encuesta y entrevista.

3.2.4 Método de Comprobación de Hipótesis

3.2.4.1 Parametrización de Variables Independientes.

3.2.4.1.1 Dimensión Seguimiento de labores

Una vez planificado las labores agrícolas en la aplicación web el usuario podrá saber si se cumplió o no la determinada labor. La aplicación web deberá entonces cumplir con el indicador:

- ✚ Eficiencia en resultados: Se agregará un registro sobre el plan de labores agrícolas ingresada en la aplicación.

3.2.4.1.2 Dimensión Generación de reportes

Para la generación de reportes se utilizarán los datos registrados en la aplicación. Lo cual significa que la aplicación cumplirá con los indicadores:



- ✚ Reporte de planificación. Se generan de la planificación asignada al manejo agrícola.
- ✚ Reporte de Estimación. Se genera en base a la posible producción que tendrá una determinada hacienda.
- ✚ Reporte de Producción. Se genera en base a la producción que genera una determinada hacienda.

3.2.4.2 Medición de Variables Dependientes.

Para verificar las dimensiones de la investigación se utilizarán las pruebas anteriores, con lo cual se podrá exponer el grado de eficiencia de la aplicación.

3.2.4.2.1 Dimensión Fiabilidad.

Se comprobará si la documentación si se cumple la planificación previamente establecido. Para que la planificación se cumpla con éxito la aplicación deberá realizar con los indicadores:

-  Tasa de pérdida de documentos: se determina la reducción de pérdida de documentos, por medio de los registros generados en el ingreso de datos a la aplicación. Se aplica una encuesta a los administradores de haciendas.
-  Manejo de labores de producción de Banano. Este indicador mostrará si se cumple o no la planificación de una determinada hacienda; para evaluar se aplica encuesta a dueños de haciendas










CAPÍTULO IV

RESULTADOS Y DISCUSIÓN

4.1. RESULTADO DE DESARROLLO DEL SITIO WEB

4.1.1 Inicio.

Los requerimientos para el desarrollo de la aplicación se han definido de acuerdo al manejo agrícola de la producción de banano y a las necesidades de los socios productores de APROBAVEN que fueron establecidas con los siguientes requerimientos:

-  Ingresos a la aplicación web mediante un nombre de usuario y clave.
-  Gestionar Usuarios.
-  Registros de socios.
-  Registros de Hacienda.
-  Asignación de lotes.
-  Definición de procesos.
-  Registro de estimación.
-  Registro de producción.
-  Módulo de Reportes.

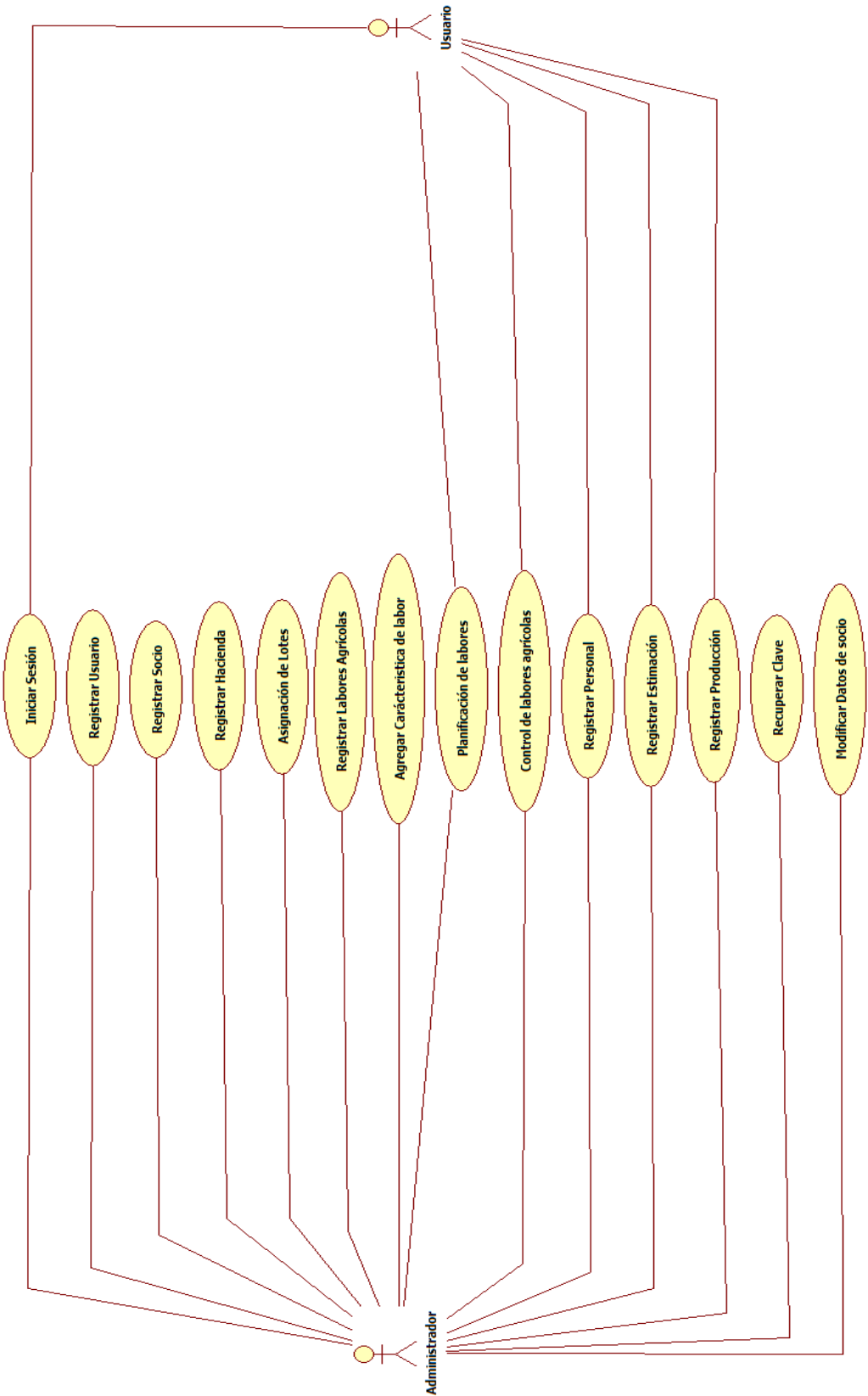
En el requerimiento denominado como definición de procesos este debe contener toda la información pertinente en forma detallada para su aplicación.

Como se detalló en el análisis de requerimientos el cultivo de banano está sujeto a procesos que no varían su estructura y mantiene las fases durante toda su producción. Por esta razón el sistema debe permitir crear los procesos indispensables con sus respectivas características para una correcta forma del cultivo de banano.

4.1.2 Elaboración.

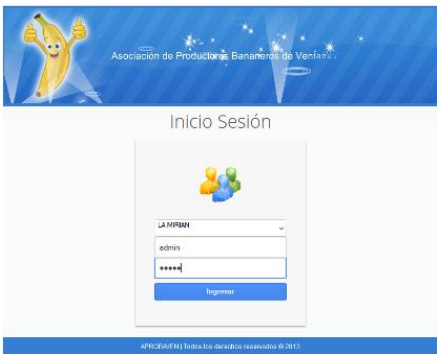
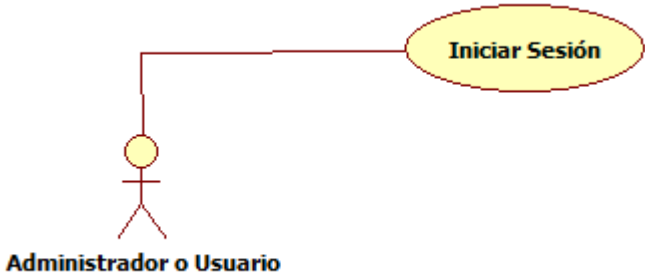
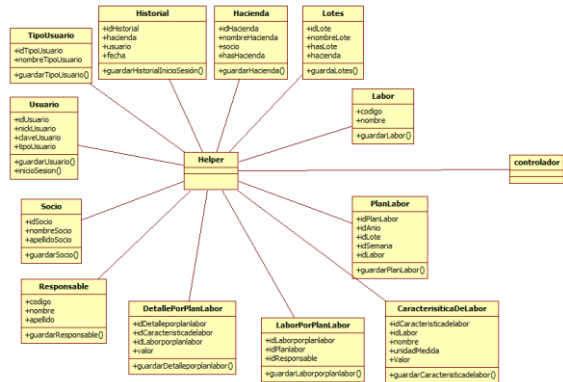
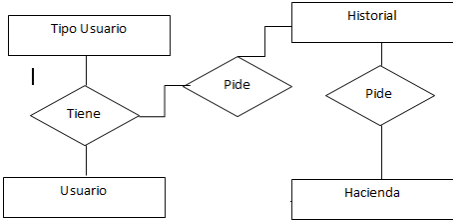
Se detallan el diagrama de casos de usos y los casos de usos


Ilustración 3 Diagrama de casos de usos




4.1.2.1 Casos de Usos

Caso de Uso	Iniciar Sesión		
Código	CU0001		
Paquete	Controlador		
Actores	Administrador o Usuario (Iniciador)		
Propósito	Permitir acceder a los servicios de sistemas tales como registrar haciendas, control del manejo agrícola		
Resumen	El Usuario o Administrador inicia sesión con su nombre de Usuario y Clave respectivamente, el sistema verifica que los datos ingresados sean correctos. Una vez iniciada la sesión: el sistema visualiza el menú de opciones disponibles.		
Prioridad	Alta	Implementación	Esencial
Tipo	Primario		
Precondiciones	El Usuario o Administrador se encuentren en la página de Inicio de Sesión. La cuenta de Usuario exista		
Flujo Normal Del evento	Acción del Actor	Respuesta del Sistema	
	1. El Usuario o Administrador escribe el URL para acceder a la aplicación. 3. El Usuario o Administrador escriben los datos solicitados. 4. Da clic sobre el botón Ingresar.	2. Muestra una página web donde el Usuario o Administrador puedan ingresar su Nombre de Usuario y Clave. 5. Verifica que los datos sean correctos. 6. El sistema le permite acceso al Usuario o Administrador y muestra una página con el menú de opciones.	
Flujo Alterno	<ul style="list-style-type: none">• Paso 2: El servidor no esté disponible y no se pueda hacer uso de la aplicación.• Paso 5: EL nombre de Usuario y clave no sean correctos se retorna al paso 3.		
Post Condiciones	EL Usuario o Administrador puede hacer uso del sistema.		


Página	
Operaciones	Se almacena datos de Inicio de Sesión
Validaciones	<ul style="list-style-type: none"> • Verificar que existen datos en la casilla de nombre de usuario y clave. • Verificar que el nombre de usuario y la clave existen. • Verificar nivel de acceso.
Diagrama de de Usos relacionados	
Clases Relacionadas	
Diagrama de datos relacionado	

Caso de Uso	Registrar Usuario		
Código	CU0002		
Paquete	Controlador		
Actores	Administrador (Iniciador)		
Propósito	Crear una cuenta para acceder a los recursos del sistema		
Resumen	Se ingresa el sistema y se escoge la opción Registrarme y se procede a registrar los datos solicitados para ingresar un nuevo usuario.		
Prioridad	Alta	Implementación	Esencial
Tipo	Primario		
Precondiciones	<ul style="list-style-type: none">El administrador haya escrito la URL en el navegador web.Utilizar el CU0001 Iniciar Sesión		
Flujo Normal Del evento	<p>Acción del Actor</p> <p>1. Administrador accede al sitio Web usando el CU0001 Iniciar Sesión.</p> <p>2. Escoger la opción Registrarme.</p> <p>4. Ingresa los datos solicitados. A continuación presionar sobre el botón Guardar.</p>	<p>Respuesta del Sistema</p> <p>3. Solicita llenar la información (Id, Nick, Clave, Nombre, Tipo de Usuario).</p> <p>5. El sistema guarda los datos ingresados.</p>	
Flujo Alterno	<ul style="list-style-type: none">Paso 3: El servidor no esté disponible y no se pueda hacer uso de la aplicación.Paso 5: Si el Nick ha sido registrado anteriormente muestra un mensaje que el nick ya existe y retorna al paso 3.		
Post Condiciones	Los datos son guardados en el sistema y puede continuar usando la aplicación.		
Página	<div></div>		


Operaciones	Se almacenan datos de Usuarios registrados.
Validaciones	Verificar que existan datos en todas las casillas requeridas.
Diagrama de de Usos relacionados	<pre> graph LR Admin[Administrador] --- Registrar([Registrar Usuario]) Registrar -.-> <<include>> Iniciar([Iniciar Sesión]) </pre>
Clases Relacionadas	<pre> classDiagram class TipoUsuario { +idTipoUsuario +nombreTipoUsuario +guardarTipoUsuario() } class Usuario { +idUsuario +emailUsuario +claveUsuario +guardarUsuario() +inicioSesion() } class Socio { +idSocio +nombreSocio +apellidoSocio +guardarSocio() } class Responsable { +codigo +nombre +apellido +guardarResponsable() } class Historial { +idHistorial +idHacienda +usuario +fecha +guardarHistorialInicioSesion() } class Hacienda { +idHacienda +nombreHacienda +precio +idHacienda +guardarHacienda() } class Lotes { +idLote +nombreLote +idLote +idHacienda +guardarLotes() } class Labor { +codigo +nombre +guardarLabor() } class PlantLabor { +idPlantLabor +idLote +idLote +idSemana +idLabor +guardarPlantLabor() } class DetallePorPlantLabor { +idDetallePorPlantLabor +idCaracteristicaDelLabor +idLaborPorPlantLabor +idLabor +guardarDetallePorPlantLabor() } class LaborPorPlantLabor { +idLaborPorPlantLabor +idPlantLabor +idResponsable +guardarLaborPorPlantLabor() } class CaracteristicaDelLabor { +idCaracteristicaDelLabor +idLabor +nombre +unidadMedida +idLabor +guardarCaracteristicaDelLabor() } class controlador Helper TipoUsuario --> Helper Usuario --> Helper Socio --> Helper Responsable --> Helper Historial --> Helper Hacienda --> Helper Lotes --> Helper Labor --> Helper PlantLabor --> Helper DetallePorPlantLabor --> Helper LaborPorPlantLabor --> Helper CaracteristicaDelLabor --> Helper controlador --> Helper </pre>
Diagrama de datos relacionado	<pre> graph TD TU[Tipo Usuario] -- Tiene --> U[Usuario] </pre>

Caso de Uso	Registrar Socio		
Código	CU0003		
Paquete	Controlador		
Actores	Administrador (Iniciador)		
Propósito	Crear una lista de Socios pertenecientes a la Asociación		
Resumen	Se ingresa el sistema y se escoge la opción Socio y se procede a registrar los datos solicitados para ingresar un nuevo socio.		
Prioridad	Alta	Implementación	Esencial
Tipo	Primario		
Precondiciones	<ul style="list-style-type: none">El administrador haya escrito la URL en el navegador web.Utilizar el CU0001 Iniciar Sesión.		
Flujo Normal Del evento	<p>Acción del Actor</p> <p>1. Administrador accede al sitio Web usando el CU0001 Iniciar Sesión.</p> <p>2. Escoger la opción Socio.</p> <p>4. Ingresar los datos solicitados. A continuación presionar sobre el botón Guardar.</p>	<p>Respuesta del Sistema</p> <p>3. Solicita llenar la información (N° de cédula, Nombres, Apellidos, Dirección, Teléfono, Celular y Correo).</p> <p>5. El sistema guarda los datos ingresados.</p>	
Flujo Alterno	<ul style="list-style-type: none">Paso 3: El servidor no está disponible y no se pueda hacer uso de la aplicación.Paso 5: Si el socio ha sido registrado anteriormente muestra un mensaje que el socio ya existe y retorna al paso 3.		
Post Condiciones	Los datos son guardados en el sistema y puede continuar usando la aplicación.		
Página	<div></div>		


Operaciones	Se almacenan datos de un nuevo socio
Validaciones	Verificar que existan datos en todas las casillas requeridas.
Diagrama de de Usos relacionados	<pre> graph TD Admin[Administrador] --- RegistrarSocio((Registrar Socio)) RegistrarSocio -.-> <<include>> IniciarSesion((Iniciar Sesión)) </pre>
Clases Relacionadas	<pre> classDiagram class TipoUsuario { +idTipoUsuario +nombreTipoUsuario +guardarTipoUsuario() } class Usuario { +idUsuario +emailUsuario +claveUsuario +guardarUsuario() +inicioSesion() } class Historial { +idHistorial +idHacienda +usuario +fecha +guardarHistorialInicioSesion() } class Hacienda { +idHacienda +nombreHacienda +socio +guardarHacienda() } class Lotes { +idLote +nombreLote +idHacienda +guardarLotes() } class Labor { +codigo +nombre +guardarLabor() } class PlantLabor { +idPlantLabor +idLote +idLabor +guardarPlantLabor() } class Responsable { +codigo +nombre +apellido +guardarResponsable() } class Socio { +idSocio +nombreSocio +apellidoSocio +guardarSocio() } class DetallePorPlantLabor { +idDetalleporplantlabor +idCaracteristicadelabor +idLaborporplantlabor +email +guardarDetalleporplantlabor() } class LaborPorPlantLabor { +idLaborporplantlabor +idPlantlabor +idResponsable +guardarLaborporplantlabor() } class CaracteristicadelLabor { +idCaracteristicadelabor +idLabor +nombre +unidadMedida +valor +guardarCaracteristicadelabor() } class controlador Helper --- TipoUsuario Helper --- Usuario Helper --- Historial Helper --- Hacienda Helper --- Lotes Helper --- Labor Helper --- PlantLabor Helper --- Responsable Helper --- Socio Helper --- DetallePorPlantLabor Helper --- LaborPorPlantLabor Helper --- CaracteristicadelLabor Helper --- controlador </pre>
Diagrama de datos relacionado	<pre> classDiagram class Socio </pre>

Caso de Uso	Registrar Hacienda		
Código	CU0004		
Paquete	Controlador		
Actores	Administrador		
Propósito	Permitir que el administrador pueda agregar las haciendas al sistema, que posteriormente permitirán que los usuarios registrados puedan llevar un control del manejo agrícola.		
Resumen	El administrador ingresa los datos de la hacienda, lo registra y queda listo para los usuarios registrados puedan hacer uso del sitio.		
Prioridad	Alta	Implementación	Esencial
Tipo	Primario		
Precondiciones	<ul style="list-style-type: none">El Administrador ha escrito la URL en el navegador web.Utilizar el CU0001 Iniciar Sesión.		
Flujo Normal Del evento	<p>Acción del Actor</p> <p>1. Administrador accede al sitio Web usando el CU0001 Iniciar Sesión.</p> <p>2. Escoger la opción Hacienda.</p> <p>4. Ingresa los datos solicitados. A continuación presionar sobre el botón Guardar.</p>	<p>Respuesta del Sistema</p> <p>3. Solicita llenar la información (Código, Nombres, Dirección, Hectáreas y Socio).</p> <p>5. El sistema guarda los datos ingresados.</p>	
Flujo Alterno	<ul style="list-style-type: none">Paso 3: El servidor no está disponible y no se puede hacer uso de la aplicación.Paso 5: Si la Hacienda ha sido registrada anteriormente muestra un mensaje que la hacienda ya existe y retorna al paso 3.		
Post Condiciones	Los datos son guardados en el sistema y puede continuar usando la aplicación.		
Página	<div></div>		


Operaciones	Se almacena datos de una nueva hacienda.
Validaciones	Verificar que existan datos en todas las casillas requeridas.
Diagrama de de Usos relacionados	<pre> graph TD Admin[Administrador] --- Registrar([Registrar Hacienda]) Registrar -.-> <<include>> Iniciar([Iniciar Sesión]) </pre>
Clases Relacionadas	<pre> classDiagram class TipUsuario { +idTipUsuario +nombreTipUsuario +guardarTipUsuario() } class Usuario { +idUsuario +emailUsuario +dniUsuario +guardarUsuario() +iniciarSesion() } class Socio { +idSocio +nombreSocio +apellidoSocio +guardarSocio() } class Responsable { +codigo +nombre +apellido +guardarResponsable() } class Historial { +idHistorial +idHacienda +usuario +fecha +guardarHistorialIniciarSesion() } class Hacienda { +idHacienda +nombreHacienda +socio +idPlantLabor +guardarHacienda() } class Lotes { +idLote +nombreLote +idHacienda +guardarLotes() } class Labor { +codigo +nombre +guardarLabor() } class PlantLabor { +idPlantLabor +idLote +idLabor +guardarPlantLabor() } class DetallePorPlantLabor { +idDetallePorPlantLabor +idCaracteristicaDeLabor +idLaborPorPlantLabor +valor +guardarDetallePorPlantLabor() } class LaborPorPlantLabor { +idLaborPorPlantLabor +idPlantLabor +idResponsable +guardarLaborPorPlantLabor() } class CaracteristicaDeLabor { +idCaracteristicaDeLabor +idLabor +nombre +unidadMedida +valor +guardarCaracteristicaDeLabor() } class controlador Helper --- TipUsuario Helper --- Usuario Helper --- Socio Helper --- Responsable Helper --- Historial Helper --- Hacienda Helper --- Lotes Helper --- Labor Helper --- PlantLabor Helper --- DetallePorPlantLabor Helper --- LaborPorPlantLabor Helper --- CaracteristicaDeLabor controlador --- Labor controlador --- PlantLabor </pre>
Diagrama de datos relacionado	<pre> classDiagram class Hacienda { +idHacienda } Hacienda --> Hacienda </pre>

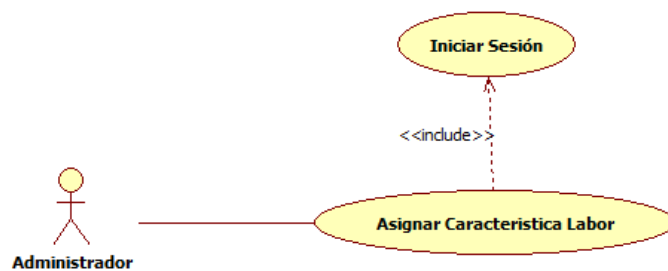
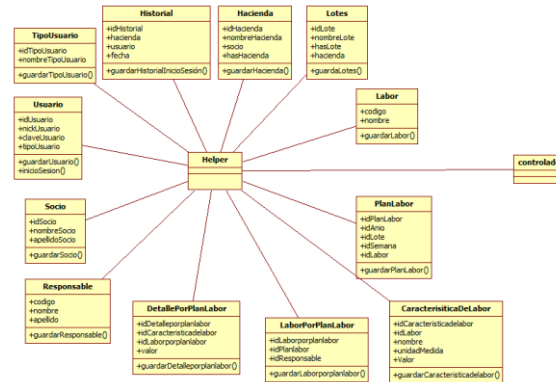
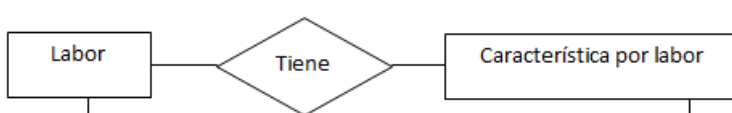
Caso de Uso	Asignación de Lotes		
Código	CU0005		
Paquete	Controlador		
Actores	Administrador (Iniciador)		
Propósito	El administrador divide la hacienda por lote.		
Resumen	El administrador asigna los lotes necesarios según sus hectáreas.		
Prioridad	Alta	Implementación	Esencial
Tipo	Primario.		
Precondiciones	<ul style="list-style-type: none">El Administrador ha escrito la URL en el navegador web.Utilizar el CU0001 Iniciar Sesión.		
Flujo Normal Del evento	<p>Acción del Actor</p> <p>1. Administrador accede al sitio Web usando el CU0001 Iniciar Sesión.</p> <p>2. Escoger la opción Lotes.</p> <p>4. Ingresa los datos solicitados. A continuación presionar sobre el botón Guardar.</p>	<p>Respuesta del Sistema</p> <p>3. Muestra los lotes Registrado de la Hacienda y solicita llenar la información (Nombres y Hectáreas).</p> <p>5. El sistema guarda los datos ingresados.</p>	
Flujo Alterno	<ul style="list-style-type: none">Paso 3: El servidor no está disponible y no se puede hacer uso de la aplicación.Paso 5: Si la Hectáreas asignadas son mayor a las Hectáreas registradas retorna al paso 3.		
Post Condiciones	Los datos son guardados en el sistema y puede continuar usando la aplicación.		
Página	<div></div>		


Operaciones	Se almacena los lotes de la Hacienda.
Validaciones	Verificar que existan datos en las casillas requeridas.
Diagrama de de Usos relacionados	<pre> graph LR Admin[Administrador] --- Asignacion([Asignación de Lotes]) Asignacion -.-> <<include>> IniciarSesion([Iniciar Sesión]) </pre>
Clases Relacionadas	<pre> classDiagram class TipoUsuario { +idTipoUsuario +nombreTipoUsuario +guardarTipoUsuario() } class Usuario { +idUsuario +emailUsuario +claveUsuario +tipoUsuario +guardarUsuario() +inicioSesion() } class Socio { +idSocio +nombreSocio +apellidoSocio +guardarSocio() } class Responsable { +codigo +nombre +apellido +guardarResponsable() } class Historial { +idHistorial +hacienda +usuario +fecha +guardarHistorialInicioSesion() } class Hacienda { +idHacienda +nombreHacienda +especie +guardarHacienda() } class Lotes { +idLote +nombreLote +estadoLote +guardarLote() } class Labor { +codigo +nombre +guardarLabor() } class PlantLabor { +idPlantLabor +idLote +idHacienda +idLabor +guardarPlantLabor() } class DetallePorPlantLabor { +idDetallePorPlantLabor +idCaracteristicaDelLabor +idLaborPorPlantLabor +email +guardarDetallePorPlantLabor() } class LaborPorPlantLabor { +idLaborPorPlantLabor +idPlantLabor +idResponsable +guardarLaborPorPlantLabor() } class CaracteristicasDelLabor { +idCaracteristicaDelLabor +idLabor +nombre +cantidadPiedra +valor +guardarCaracteristicaDelLabor() } class controlador Helper --- TipoUsuario Helper --- Usuario Helper --- Socio Helper --- Responsable Helper --- Historial Helper --- Hacienda Helper --- Lotes Helper --- Labor Helper --- PlantLabor Helper --- DetallePorPlantLabor Helper --- LaborPorPlantLabor Helper --- CaracteristicasDelLabor Helper --- controlador </pre>
Diagrama de datos relacionado	<pre> graph LR Hacienda[Hacienda] --- Tiene{Tiene} Tiene --- Lotes[Lotes] </pre>

Caso de Uso	Registrar Labores Agrícolas		
Código	CU0006		
Paquete	Controlador		
Actores	Administrador (Iniciador)		
Propósito	Permitir que el administrador ingrese las labores		
Resumen	El Administrador ingresa todas las labores que se realizan dentro de las plantaciones.		
Prioridad	Alta	Implementación	Esencial
Tipo	Primario		
Precondiciones	<ul style="list-style-type: none">El Administrador ha escrito la URL en el navegador web.Utilizar el CU0001 Iniciar Sesión.		
Flujo Normal Del evento	<p>Acción del Actor</p> <p>1. Administrador accede al sitio Web usando el CU0001 Iniciar Sesión.</p> <p>2. Escoger la opción Administración.</p> <p>4. Escogemos la opción Agregar Labor.</p> <p>6. Ingresa los datos solicitados. A continuación presionar sobre el botón Guardar.</p>	<p>Respuesta del Sistema</p> <p>3. Muestra un menú con opciones</p> <p>5. Muestra la página de Agregar Labor y nos solicita llenar la información (Código de labor, Nombre de labor).</p> <p>7. El Sistema guarda los datos ingresados.</p>	
Flujo Alterno	<ul style="list-style-type: none">Paso 3: El servidor no está disponible y no se puede hacer uso de la aplicación.		
Post Condiciones	Los datos son guardados en el sistema y puede continuar usando la aplicación.		
Página	<div></div>		

Operaciones	Se registran las labores
Validaciones	Verificar que existan datos en las casillas requeridas
Diagrama de de Usos relacionados	<pre> graph LR Admin[Administrador] --- Registrar([Registrar Labores Agrícolas]) Registrar -.-> <<include>> Iniciar([Iniciar Sesión]) </pre>
Clases Relacionadas	<pre> classDiagram class TipoUsuario { +idTipoUsuario +nombreTipoUsuario +guardarTipoUsuario() } class Usuario { +idUsuario +nombreUsuario +emailUsuario +guardarUsuario() +eliminarUsuario() } class Socio { +idSocio +nombreSocio +apellidoSocio +guardarSocio() } class Responsable { +codigo +nombre +apellido +guardarResponsable() } class Historial { +idHistorial +hacienda +usuario +fecha +guardarHistorialInicioSesion() } class Hacenda { +idHacienda +nombreHacienda +modo +guardarHacienda() } class Lotes { +idLote +nombreLote +hacienda +guardarLotes() } class Labor { +codigo +nombre +guardarLabor() } class PlantLabor { +idPlantLabor +idHacienda +idLote +idSemana +idLabor +guardarPlantLabor() } class CaracteristicaLabor { +idCaracteristicaLabor +idLabor +nombre +unidadMedida +valor +guardarCaracteristicaLabor() } class LaborPorPlantLabor { +idLaborPorPlantLabor +idPlantLabor +idResponsable +guardarLaborPorPlantLabor() } class DetallePorPlantLabor { +idDetallePorPlantLabor +idCaracteristicaLabor +idLaborPorPlantLabor +idLaborPorPlantLabor +validar +guardarDetallePorPlantLabor() } class controlador Helper --- TipoUsuario Helper --- Usuario Helper --- Socio Helper --- Responsable Helper --- Historial Helper --- Hacenda Helper --- Lotes Helper --- Labor Helper --- PlantLabor Helper --- CaracteristicaLabor Helper --- LaborPorPlantLabor Helper --- DetallePorPlantLabor controlador --- Labor </pre>
Diagrama de datos relacionado	<pre> graph TD Labor[Labor] </pre>


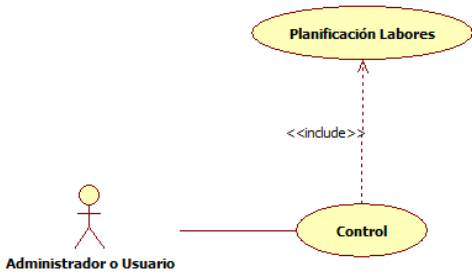
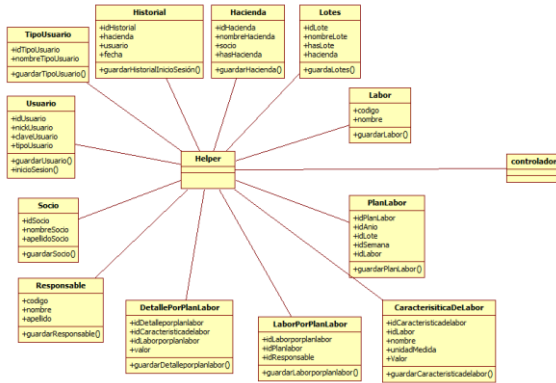
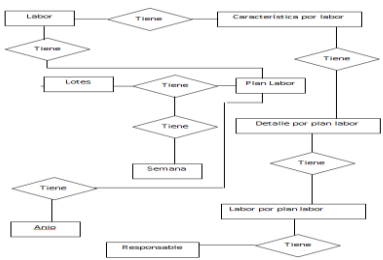
Caso de Uso	Agregar Características de Labor		
Código	CU0007		
Paquete	Controlador		
Actores	Administrador (Iniciador)		
Propósito	Permitir que el Administrador pueda tener las características de cada labor.		
Resumen	Registrar las características de cada labor ingresada al sitio web.		
Prioridad	Alta	Implementación	Esencial
Tipo	Primario		
Precondiciones	<ul style="list-style-type: none">El Administrador o Usuario ha escrito la URL en el navegador web.Utilizar el CU0001 Iniciar Sesión.Se haya ingresado la labor en el Sitio Web.		
Flujo Normal Del evento	<p>Acción del Actor</p> <p>1. Administrador o Usuario accede al sitio Web usando el CU0001 Iniciar Sesión.</p> <p>2. Escoger la opción Administración.</p> <p>4. Escogemos la opción Característica labor.</p> <p>6. Ingresamos los datos solicitados y damos clic en el botón guardar.</p>	<p>Respuesta del Sistema</p> <p>3. Muestra un menú con opciones.</p> <p>5. Muestra la página de Característica labor y solicita llenar la información (Nombre de la característica y unidad de medida de la característica).</p> <p>7. EL sistema guarda los datos ingresados.</p>	
Flujo Alterno	<ul style="list-style-type: none">Paso 3: El servidor no está disponible y no se puede hacer uso de la aplicación.		
Post Condiciones	Los datos son guardados en el sistema y puede continuar usando la aplicación.		
Página			


Operaciones	Se registra cada característica de cada labor.
Validaciones	Verificar que existan datos en todas las casillas requeridas.
Diagrama de de Usos relacionados	 <pre> graph TD Admin[Administrador] --- Asignar(Asignar Característica Labor) Asignar -.-> <<include>> Iniciar(Iniciar Sesión) </pre>
Clases Relacionadas	 <pre> classDiagram class TipoUsuario { +idTipoUsuario +nombreTipoUsuario +guardarTipoUsuario() } class Usuario { +idUsuario +emailUsuario +claveUsuario +tipoUsuario +guardarUsuario() +iniciarSesion() } class Historial { +idHistorial +idHacienda +usuario +fecha +guardarHistorialInicioSesion() } class Hacienda { +idHacienda +nombreHacienda +socio +hashHacienda +guardarHacienda() } class Lotes { +idLote +nombreLote +hashLote +idHacienda +guardarLotes() } class Labor { +codigo +nombre +guardarLabor() } class PlantLabor { +idPlantLabor +idLote +idHacienda +idSemana +idLabor +guardarPlantLabor() } class Responsable { +codigo +nombre +apellido +guardarResponsable() } class DetallePorPlantLabor { +idDetallePorPlantLabor +idCaracteristicaDeLabor +idLaborPorPlantLabor +valor +guardarDetallePorPlantLabor() } class LaborPorPlantLabor { +idLaborPorPlantLabor +idPlantLabor +idResponsable +guardarLaborPorPlantLabor() } class CaracteristicaDeLabor { +idCaracteristicaDeLabor +idLabor +nombre +unidadMedida +valor +guardarCaracteristicaDeLabor() } class controlador Helper --- TipoUsuario Helper --- Usuario Helper --- Historial Helper --- Hacienda Helper --- Lotes Helper --- Labor Helper --- PlantLabor Helper --- Responsable Helper --- DetallePorPlantLabor Helper --- LaborPorPlantLabor Helper --- CaracteristicaDeLabor Helper --- controlador </pre>
Diagrama de datos relacionado	 <pre> graph LR Labor[Labor] --- Tiene{Tiene} Tiene --- Caracteristica[Característica por labor] </pre>

Caso de Uso	Planificación de labores		
Código	CU0008		
Paquete	Controlador		
Actores	Administrador o Usuario (Iniciador)		
Propósito	Planificar por semana las labores por cada lote de la plantación.		
Resumen	El Administrador o Usuario planifica por cada lote semanalmente todas las labores que se deben realizar.		
Prioridad	Alta	Implementación	Esencial
Tipo	Primario		
Precondiciones	<ul style="list-style-type: none">El Administrador o Usuario ha escrito la URL en el navegador web.Utilizar el CU0001 Iniciar Sesión.Que los lotes conste en el sistema.		
Flujo Normal Del evento	<p>Acción del Actor</p> <p>1. Administrador o Usuario accede al sitio Web usando el CU0001 Iniciar Sesión.</p> <p>2. Escoger la opción Planificación.</p> <p>4. Se ingresa los datos requeridos y a continuación damos clic en el botón Guardar.</p>	<p>Respuesta del Sistema</p> <p>3. Muestra la página de Planificación y solicita llenar la información (Lote, Año, Semana, Labor).</p> <p>5. El sistema guarda los datos ingresados.</p>	
Flujo Alterno	<ul style="list-style-type: none">Paso 3: El servidor no está disponible y no se puede hacer uso de la aplicación.		
Post Condiciones	Los datos son guardados en el sistema y puede continuar usando la aplicación.		
Página	<div></div>		


Operaciones	Se almacenan la planificación en el sistema
Validaciones	Verificar que existan datos en todas las casillas requeridas.
Diagrama de Usos relacionados	<pre> graph LR Actor[Administrador o Usuario] --- UC1((Planificación Labores)) UC1 -.-> <<include>> UC2((Iniciar Sesión)) </pre>
Clases Relacionadas	<pre> classDiagram class TipoUsuario { +idTipoUsuario +nombreTipoUsuario +guardarTipoUsuario() } class Usuario { +idUsuario +nombreUsuario +emailUsuario +guardarUsuario() +eliminarUsuario() } class Socio { +idSocio +nombreSocio +apellidoSocio +guardarSocio() } class Responsable { +codigo +nombre +apellido +guardarResponsable() } class Historial { +idHistorial +idHacienda +idUsuario +fecha +guardarHistorialInicioSesion() } class Hacienda { +idHacienda +nombreHacienda +especie +guardarHacienda() } class Lotes { +idLote +nombreLote +idHacienda +guardarLotes() } class Labor { +codigo +nombre +guardarLabor() } class PlantLabor { +idPlantLabor +idLote +idLabor +idSemana +guardarPlantLabor() } class CaracteristicaDelLabor { +idCaracteristicaDelLabor +idLabor +nombre +unidadMedida +valor +guardarCaracteristicaDelLabor() } class LaborPorPlantLabor { +idLaborPorPlantLabor +idPlantLabor +idResponsable +guardarLaborPorPlantLabor() } class DetallePorPlantLabor { +idDetallePorPlantLabor +idCaracteristicaDelLabor +idLaborPorPlantLabor +valor +guardarDetallePorPlantLabor() } class Controlador Helper --- TipoUsuario Helper --- Usuario Helper --- Socio Helper --- Responsable Helper --- Historial Helper --- Hacienda Helper --- Lotes Helper --- Labor Helper --- PlantLabor Helper --- CaracteristicaDelLabor Helper --- LaborPorPlantLabor Helper --- DetallePorPlantLabor Helper --- Controlador </pre>
Diagrama de datos relacionado	<pre> graph TD Labor -- Tiene --> Hacienda Labor -- Tiene --> Lotes Hacienda -- Tiene --> Lotes Lotes -- Tiene --> PlanLabor[Plan Labor] PlanLabor -- Tiene --> Semana Semana -- Tiene --> Año </pre>

Caso de Uso	Control de labores agrícolas		
Código	CU0009		
Paquete	Controlador		
Actores	Administrador o Usuario (Iniciador)		
Propósito	Controlar lo planificado dentro de la plantación		
Resumen	El Administrador o Usuario escoge el Lote, Año y la Semana para saber que se planificó para dicha semana, e ingresa los respectivos datos.		
Prioridad	Alta	Implementación	Esencial
Tipo	Primario		
Precondiciones	<ul style="list-style-type: none">El Administrador o Usuario ha escrito la URL en el navegador web.Que el plan de labores se encuentre registrado.		
Flujo Normal Del evento	Acción del Actor	Respuesta del Sistema	
	1. Administrador o Usuario accede al sitio Web usando el CU0001 Iniciar Sesión.		
	2. Selecciona la Opción Control.	3. Muestra la página de Control.	
	4. Selecciona el Lote, Año y Semana. A continuación da clic en el botón ejecutar.	5. Muestra las labores planificadas para dicha semana.	
	6. Selecciona una labor y da clic en el botón agregar.	7. Muestra una página con las características de la labor seleccionada.	
	8. Ingresa los datos solicitados y da clic en el botón guardar.	9. El sistema guarda los datos ingresados.	
Flujo Alterno	<ul style="list-style-type: none">Paso 3: El servidor no está disponible y no se puede hacer uso de la aplicación.Pasó 5: No exista datos para dicho lote, semana y Año.		

Post Condiciones	Los datos son guardados en el sistema y puede continuar usando la aplicación.
Página	
Operaciones	Controla que se cumpla lo planificado.
Validaciones	Verificar que existan datos en las casillas requeridas.
Diagrama de de Usos relacionados	
Clases Relacionadas	
Diagrama de datos relacionado	

Caso de Uso	Registrar personal		
Código	CU0010		
Paquete	Controlador		
Actores	Administrador o Usuario (Iniciador)		
Propósito	Tener una nómina del personal.		
Resumen	Registrar los datos del personal que efectúa las labores dentro de la hacienda.		
Prioridad	Alta	Implementación	Real
Tipo	Primario		
Precondiciones	<ul style="list-style-type: none">El Administrador o Usuario ha escrito la URL en el navegador web.El Administrador o Usuario ha iniciado sesión.		
Flujo Normal Del evento	<p>Acción del Actor</p> <p>1. Administrador o Usuario accede al sitio Web usando el CU0001 Iniciar Sesión.</p> <p>2. Escoger la opción Personal.</p> <p>4. Se ingresa los datos requeridos y a continuación damos clic en el botón Guardar.</p>	<p>Respuesta del Sistema</p> <p>3. Muestra la página de Personal.</p> <p>5. El sistema guarda los datos ingresados.</p>	
Flujo Alterno	<ul style="list-style-type: none">Paso 3: El servidor no está disponible y no se puede hacer uso de la aplicación.		
Post Condiciones	Los datos son guardados en el sistema y puede continuar usando la aplicación.		
Página	<div></div>		
Operaciones	Guardar datos de personal.		

Validaciones	Verificar que existan datos en las casillas requeridas.
Diagrama de de Usos relacionados	<pre> graph LR Actor[Administrador o Usuario] --- UC1((Registrar Personal)) UC1 -.-> <<include>> UC2((Iniciar Sesión)) </pre>
Clases Relacionadas	<pre> classDiagram class TipoUsuario { +idTipoUsuario +nombreTipoUsuario +guardarTipoUsuario() } class Usuario { +idUsuario +emailUsuario +claveUsuario +ipUsuario +guardarUsuario() +iniciarSesion() } class Socio { +idSocio +nombreSocio +apellidoSocio +guardarSocio() } class Responsable { +idResponsable +nombre +apellido +guardarResponsable() } class Historial { +idHistorial +idHacienda +idUsuario +fecha +guardarHistorialIniciacion() } class Hacienda { +idHacienda +nombreHacienda +apodo +idHacienda +guardarHacienda() } class Lotes { +idLote +nombreLote +idLote +idHacienda +guardarLotes() } class Labor { +idLabor +nombre +guardarLabor() } class PlantLabor { +idPlantLabor +idLabor +idLote +idSemana +idLabor +guardarPlantLabor() } class DetallePorPlantLabor { +idDetallePorPlantLabor +idCaracteristicaLabor +idLaborPorPlantLabor +idLabor +guardarDetallePorPlantLabor() } class LaborPorPlantLabor { +idLaborPorPlantLabor +idPlantLabor +idResponsable +guardarLaborPorPlantLabor() } class CaracteristicaDelLabor { +idCaracteristicaDelLabor +idLabor +nombre +unidadMedida +idLabor +guardarCaracteristicaDelLabor() } class controlador Helper --- TipoUsuario Helper --- Usuario Helper --- Socio Helper --- Responsable Helper --- Historial Helper --- Hacienda Helper --- Lotes Helper --- Labor Helper --- PlantLabor Helper --- DetallePorPlantLabor Helper --- LaborPorPlantLabor Helper --- CaracteristicaDelLabor Helper --- controlador </pre>
Diagrama de datos relacionado	

Caso de Uso	Registrar Estimación		
Código	CU0011		
Paquete	Controlador		
Actores	Administrador o Usuario (Iniciador)		
Propósito	Tener datos de la posible producción.		
Resumen	Registrar datos con la posible producción de la finca-		
Prioridad	Alta	Implementación	Real
Tipo	Primario		
Precondiciones	<ul style="list-style-type: none">El Administrador o Usuario ha escrito la URL en el navegador web.El Administrador o Usuario ha iniciado sesión.		
Flujo Normal Del evento	<p>Acción del Actor</p> <p>1. Administrador o Usuario accede al sitio Web usando el CU0001 Iniciar Sesión.</p> <p>2. Escoger la opción Estimación.</p> <p>4. Se ingresa los datos requeridos y a continuación damos clic en el botón Guardar.</p>	<p>Respuesta del Sistema</p> <p>3. Muestra la página de Estimación con sus respectivas opciones.</p> <p>5. El sistema guarda los datos ingresados.</p>	
Flujo Alterno	<ul style="list-style-type: none">Paso 3: El servidor no está disponible y no se puede hacer uso de la aplicación.		
Post Condiciones	Los datos son guardados en el sistema y puede continuar usando la aplicación.		
Página			
Operaciones	Guardar datos de Estimación.		

Validaciones	Verificar que existan datos en las casillas requeridas.
Diagrama de de Usos relacionados	<pre> graph LR Actor[Administrador o Usuario] --- UC1((Registrar Estimacion)) UC1 -.-> <<include>> UC2((Iniciar Sesión)) </pre>
Clases Relacionadas	<pre> classDiagram class TipoUsuario { +id +nombre +get() +set() } class Estimacion { +id +fecha +cantidad +valor +get() +set() } class Historial { +id +fecha +valor +get() +set() } class Hacienda { +id +nombre +direccion +telefono +get() +set() } class Loter { +id +nombre +direccion +telefono +get() +set() } class Responsable { +id +nombre +direccion +telefono +get() +set() } class CaracteristicLabor { +id +nombre +direccion +telefono +get() +set() } class Usuario { +id +nombre +apellido +telefono +get() +set() } class Socio { +id +nombre +apellido +telefono +get() +set() } class PlanLabor { +id +nombre +direccion +telefono +get() +set() } class Semana { +id +nombre +get() +set() } class Ases { +id +nombre +get() +set() } class Laborporplanlabor { +id +nombre +direccion +telefono +get() +set() } class Detallelaborporplanlabor { +id +nombre +direccion +telefono +get() +set() } class Labor { +id +nombre +get() +set() } class Produccion { +id +nombre +direccion +telefono +get() +set() } class Controlador(Servlet) { +id +nombre +get() +set() } TipoUsuario --> Estimacion TipoUsuario --> Historial TipoUsuario --> Hacienda TipoUsuario --> Loter TipoUsuario --> Responsable TipoUsuario --> CaracteristicLabor Usuario --> Estimacion Usuario --> Historial Usuario --> Hacienda Usuario --> Loter Usuario --> Responsable Usuario --> CaracteristicLabor Socio --> Estimacion Socio --> Historial Socio --> Hacienda Socio --> Loter Socio --> Responsable Socio --> CaracteristicLabor PlanLabor --> Estimacion PlanLabor --> Historial PlanLabor --> Hacienda PlanLabor --> Loter PlanLabor --> Responsable PlanLabor --> CaracteristicLabor Semana --> Estimacion Semana --> Historial Semana --> Hacienda Semana --> Loter Semana --> Responsable Semana --> CaracteristicLabor Ases --> Estimacion Ases --> Historial Ases --> Hacienda Ases --> Loter Ases --> Responsable Ases --> CaracteristicLabor Laborporplanlabor --> Estimacion Laborporplanlabor --> Historial Laborporplanlabor --> Hacienda Laborporplanlabor --> Loter Laborporplanlabor --> Responsable Laborporplanlabor --> CaracteristicLabor Detallelaborporplanlabor --> Estimacion Detallelaborporplanlabor --> Historial Detallelaborporplanlabor --> Hacienda Detallelaborporplanlabor --> Loter Detallelaborporplanlabor --> Responsable Detallelaborporplanlabor --> CaracteristicLabor Labor --> Estimacion Labor --> Historial Labor --> Hacienda Labor --> Loter Labor --> Responsable Labor --> CaracteristicLabor Produccion --> Estimacion Produccion --> Historial Produccion --> Hacienda Produccion --> Loter Produccion --> Responsable Produccion --> CaracteristicLabor Controlador(Servlet) --> Estimacion Controlador(Servlet) --> Historial Controlador(Servlet) --> Hacienda Controlador(Servlet) --> Loter Controlador(Servlet) --> Responsable Controlador(Servlet) --> CaracteristicLabor </pre>
Diagrama de datos relacionado	<pre> graph TD Hacienda[Hacienda] -- Tiene --> Estimacion[Estimación] </pre>


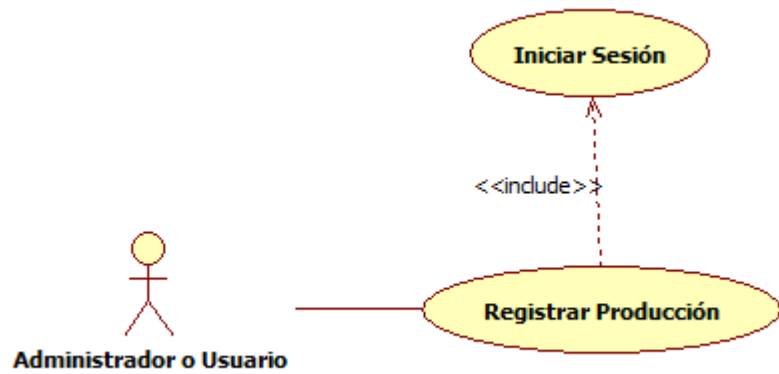
Caso de Uso	Registrar Producción		
Código	CU0012		
Paquete	Controlador		
Actores	Administrador o Usuario (Iniciador)		
Propósito	Tener datos de la producción.		
Resumen	Registrar los datos de la producción real de la finca.		
Prioridad	Alta	Implementación	Real
Tipo	Primario		
Precondiciones	<ul style="list-style-type: none">El Administrador o Usuario ha escrito la URL en el navegador web.El Administrador o Usuario ha iniciado sesión.		
Flujo Normal Del evento	<p>Acción del Actor</p> <p>1. Administrador o Usuario accede al sitio Web usando el CU0001 Iniciar Sesión.</p> <p>2. Escoger la opción Producción.</p> <p>4. Se ingresa los datos requeridos y a continuación damos clic en el botón Guardar.</p>	<p>Respuesta del Sistema</p> <p>3. Muestra la página de Producción con sus respectivas opciones.</p> <p>5. El sistema guarda los datos ingresados.</p>	
Flujo Alterno	<ul style="list-style-type: none">Paso 3: El servidor no está disponible y no se puede hacer uso de la aplicación.		
Post Condiciones	Los datos son guardados en el sistema y puede continuar usando la aplicación.		
Página			
Operaciones	Guardar datos de producción.		
Validaciones	Verificar que existan datos en las casillas requeridas.		

Diagrama de de Usos relacionados



Clases Relacionadas

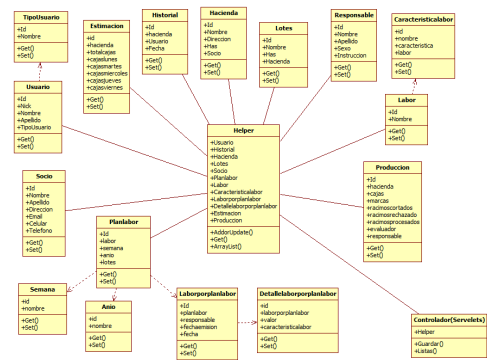
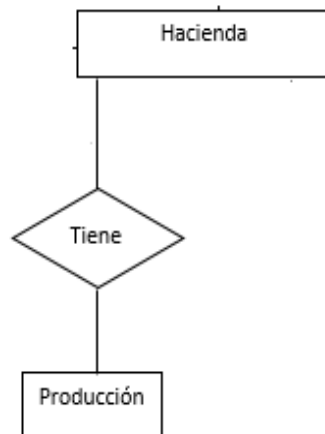




Diagrama de datos relacionado



Caso de Uso	Recuperar clave		
Código	CU0013		
Paquete	Controlador		
Actores	Administrador (Iniciador)		
Propósito	Cambiar la clave para no perder acceso al sistema.		
Resumen	Se ingresa el sistema y se escoge la opción Registrarme y y se da clic en modificar y se procede a registrar los datos solicitados para cambiar clave de usuario.		
Prioridad	Alta	Implementación	Esencial
Tipo	Primario		
Precondiciones	<ul style="list-style-type: none">El administrador haya escrito la URL en el navegador web.Utilizar el CU001 Iniciar Sesión		
Flujo Normal Del evento	Acción del Actor	Respuesta del Sistema	
	1. Administrador accede al sitio Web usando el CU0001 Iniciar Sesión.		
	2. Escoger la opción Registrarme.		
	3. Dar clic en el botón modificar	4. Muestra una página y solicita escoger nombre de usuario.	
	5. Se selecciona el nombre de usuario y da clic en buscar.	6. Muestra los datos del usuario seleccionado.	
	7. Ingresa la nueva contraseña y selecciona el tipo de usuario. A continuación presionar sobre el botón Guardar.	8. El sistema guarda los datos ingresados.	
Flujo Alterno	<ul style="list-style-type: none">Paso 3: El servidor no esté disponible y no se pueda hacer uso de la aplicación.		
Post Condiciones	Los datos son guardados en el sistema y puede continuar usando la aplicación.		
Página	<div><div>APROBAVEN</div><div>Guía Operativa</div></div> <div><div>Usuario: ADMINISTRADOR</div><div>Buscar</div><div><div><div>Id: 0</div><div>Nick: ADMIN</div><div>Nombre: ADMINISTRADOR</div><div>Apellido: SISTEMA</div><div>Contraseña</div><div>Tipo Usuario: </div></div><div>Guardar</div></div></div>		

Operaciones	Se almacenan datos de Usuarios registrados.
Validaciones	Verificar que existan datos en todas las casillas requeridas.
Diagrama de de Usos relacionados	<pre> graph LR Admin[Administrador] --- IniciarSesion([Iniciar Sesión]) Admin --- RecuperarClave([Recuperar clave]) RecuperarClave -.-> <<include>> IniciarSesion </pre>
Clases Relacionadas	<pre> classDiagram class TipoUsuario { +id +nombre +get() +set() } class Usuario { +id +nombre +apellido +telefono +get() +set() } class Socio { +id +nombre +apellido +direccion +telefono +get() +set() } class Semana { +id +nombre +get() +set() } class Asocio { +id +nombre +get() +set() } class PlanLabor { +id +labor +responsable +fecha +get() +set() } class Labor { +id +nombre +get() +set() } class Responsable { +id +nombre +apellido +telefono +get() +set() } class CaracteristicLabor { +id +nombre +caracteristica +labor +get() +set() } class Produccion { +id +nombre +codigo +descripcion +acompañamiento +material +responsable +get() +set() } class ControladorServicios { +id +get() +set() } class Helper { +usuario +password +codigo +id +caracteristicLabor +labor +acompañamiento +material +responsable +get() +set() } TipoUsuario --> Usuario Usuario --> Socio Socio --> Semana Semana --> Asocio Asocio --> PlanLabor PlanLabor --> Labor Labor --> Responsable Responsable --> CaracteristicLabor CaracteristicLabor --> Produccion Produccion --> ControladorServicios ControladorServicios --> Helper Helper --> TipoUsuario Helper --> Usuario Helper --> Socio Helper --> Semana Helper --> Asocio Helper --> PlanLabor Helper --> Labor Helper --> Responsable Helper --> CaracteristicLabor Helper --> Produccion Helper --> ControladorServicios </pre>
Diagrama de datos relacionado	<pre> graph TD TU[Tipo Usuario] -- Tiene --> U[Usuario] </pre>

Caso de Uso	Modificar Socio		
Código	CU0014		
Paquete	Controlador		
Actores	Administrador (Iniciador)		
Propósito	Modificar datos de socios pertenecientes a la Asociación		
Resumen	Se ingresa el sistema y se escoge la opción Socio y se da clic en el botón modificar y se procede a ingresar los datos solicitados.		
Prioridad	Alta	Implementación	Esencial
Tipo	Primario		
Precondiciones	<ul style="list-style-type: none">El administrador haya escrito la URL en el navegador web.Utilizar el CU0001 Iniciar Sesión.		
Flujo Normal Del evento	<p>Acción del Actor</p> <p>1. Administrador accede al sitio Web usando el CU0001 Iniciar Sesión.</p> <p>2. Escoger la opción Socio.</p> <p>3. Dar clic en el botón modificar</p> <p>5. Selecciona un nombre y dar clic en buscar.</p> <p>7. Ingrese datos de nueva dirección, teléfono, celular y correo. A continuación presionar sobre el botón Guardar.</p>	<p>Respuesta del Sistema</p> <p>4. Muestra una pagina y solicita seleccionar un nombre</p> <p>6. Muestra los datos del socio seleccionado.</p> <p>8. El sistema guarda los datos ingresados.</p>	
Flujo Alterno	<ul style="list-style-type: none">Paso 3: El servidor no está disponible y no se pueda hacer uso de la aplicación.		
Post Condiciones	Los datos son guardados en el sistema y puede continuar usando la aplicación.		
Página	<div><div><div>Socio:</div><div>Admin Admin</div><div>Buscar</div><div>Id:</div><div>1</div><div>Nombre:</div><div>Admin</div><div>Apellido:</div><div>Admin</div><div>Dirección:</div><div>DirAdmin</div><div>Telefono:</div><div>0000</div><div>Celular:</div><div>0000</div><div>E-mail:</div><div>aprobaven@gmail.com</div><div>Guardar</div></div></div>		

4.1.2.2 Diagrama de Actividades

Ilustración 4 Diagrama de Actividades de Inicio Sesión

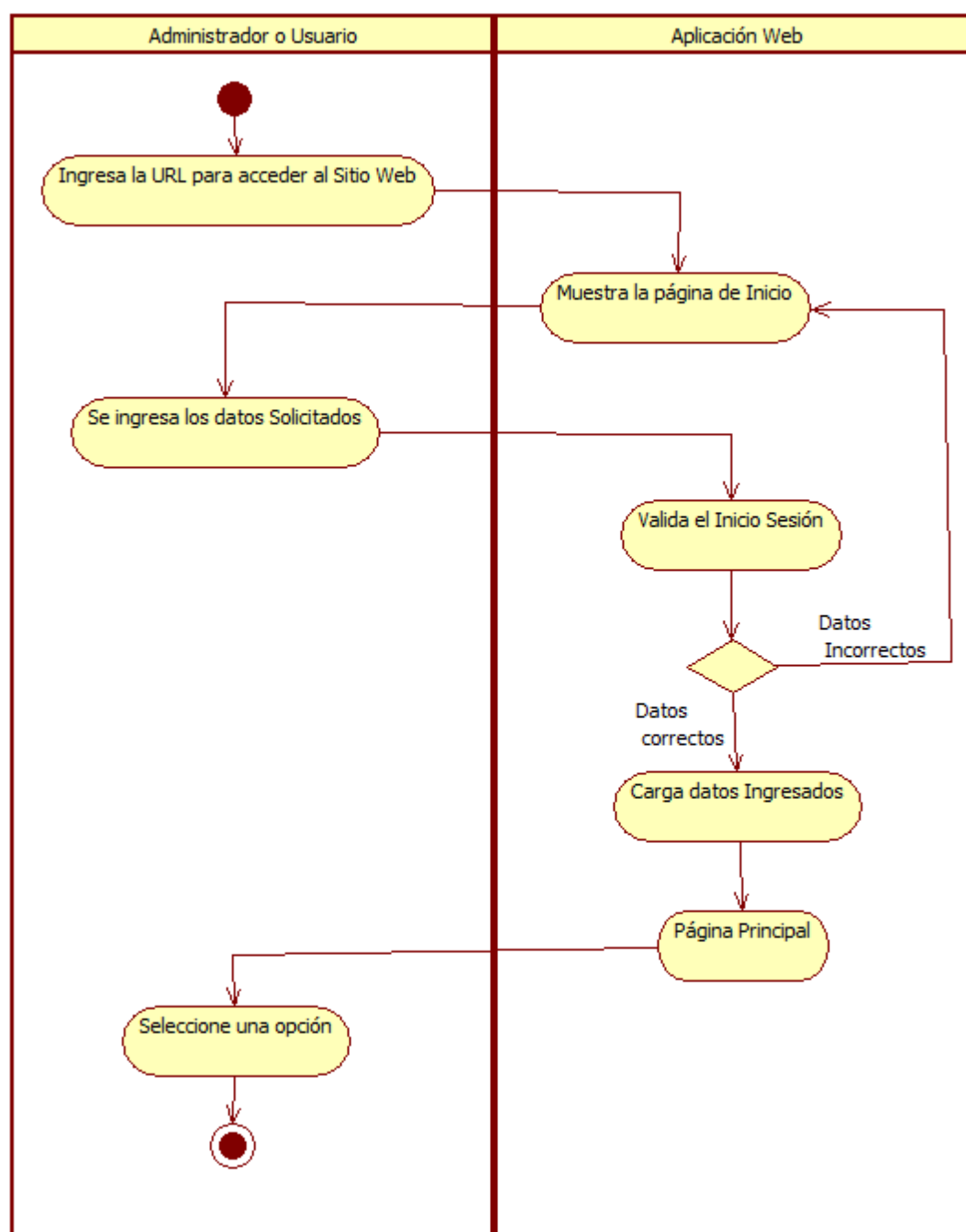


Ilustración 5 Diagrama de Actividades de Registrar Usuario

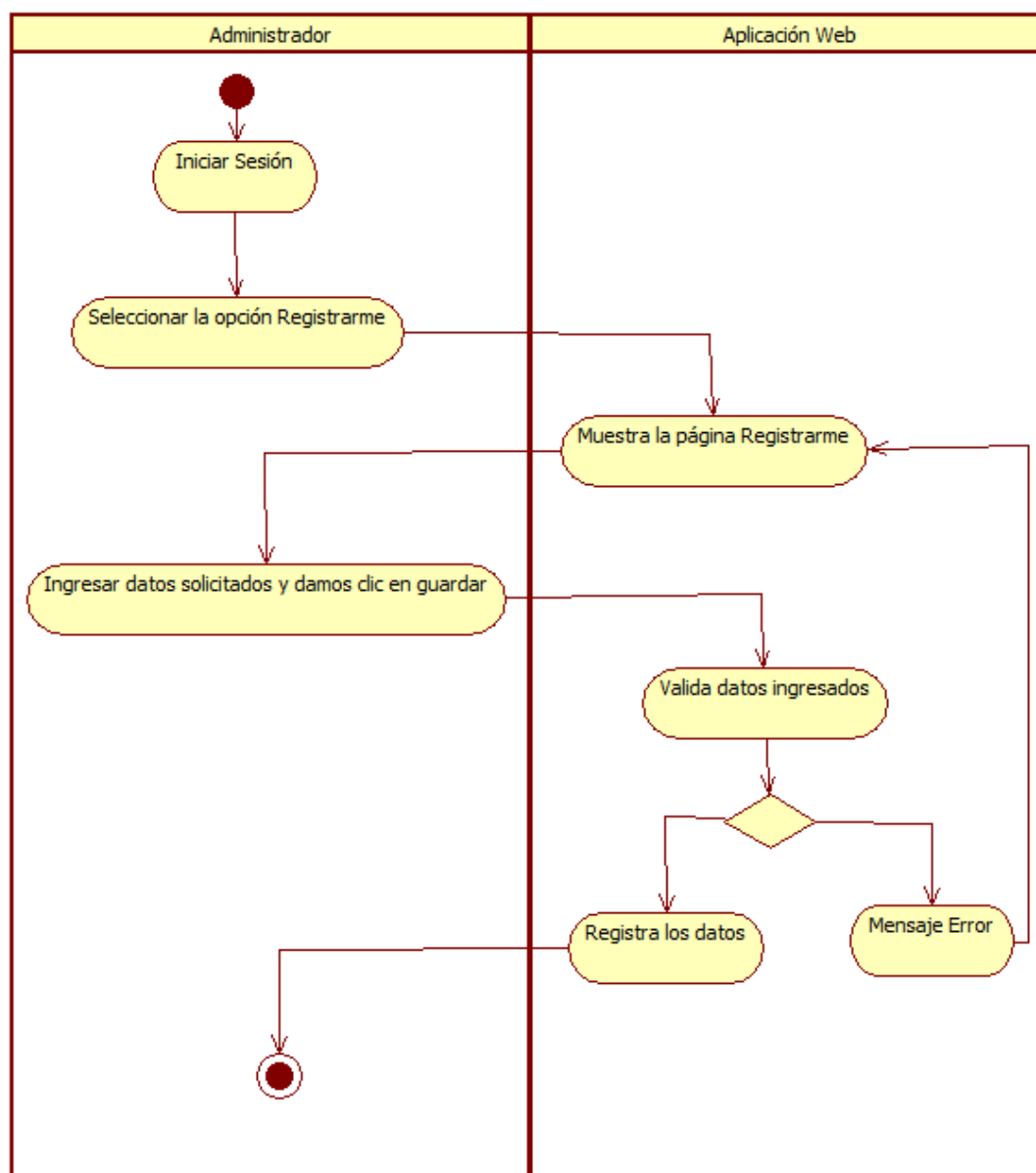


Ilustración 6 Diagrama de Actividades de Registrar Socio

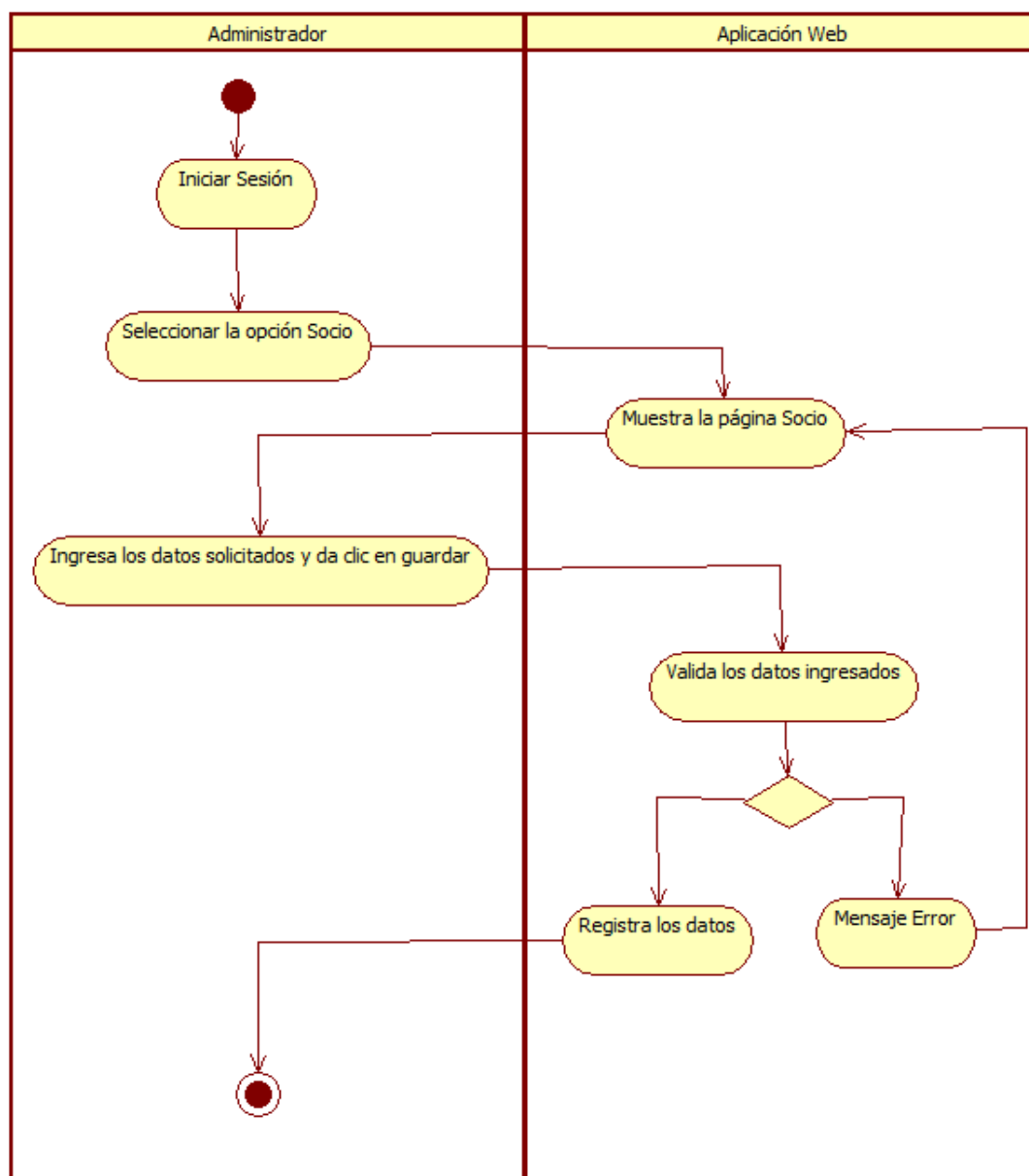


Ilustración 7 Diagrama de Actividades de Registrar Hacienda

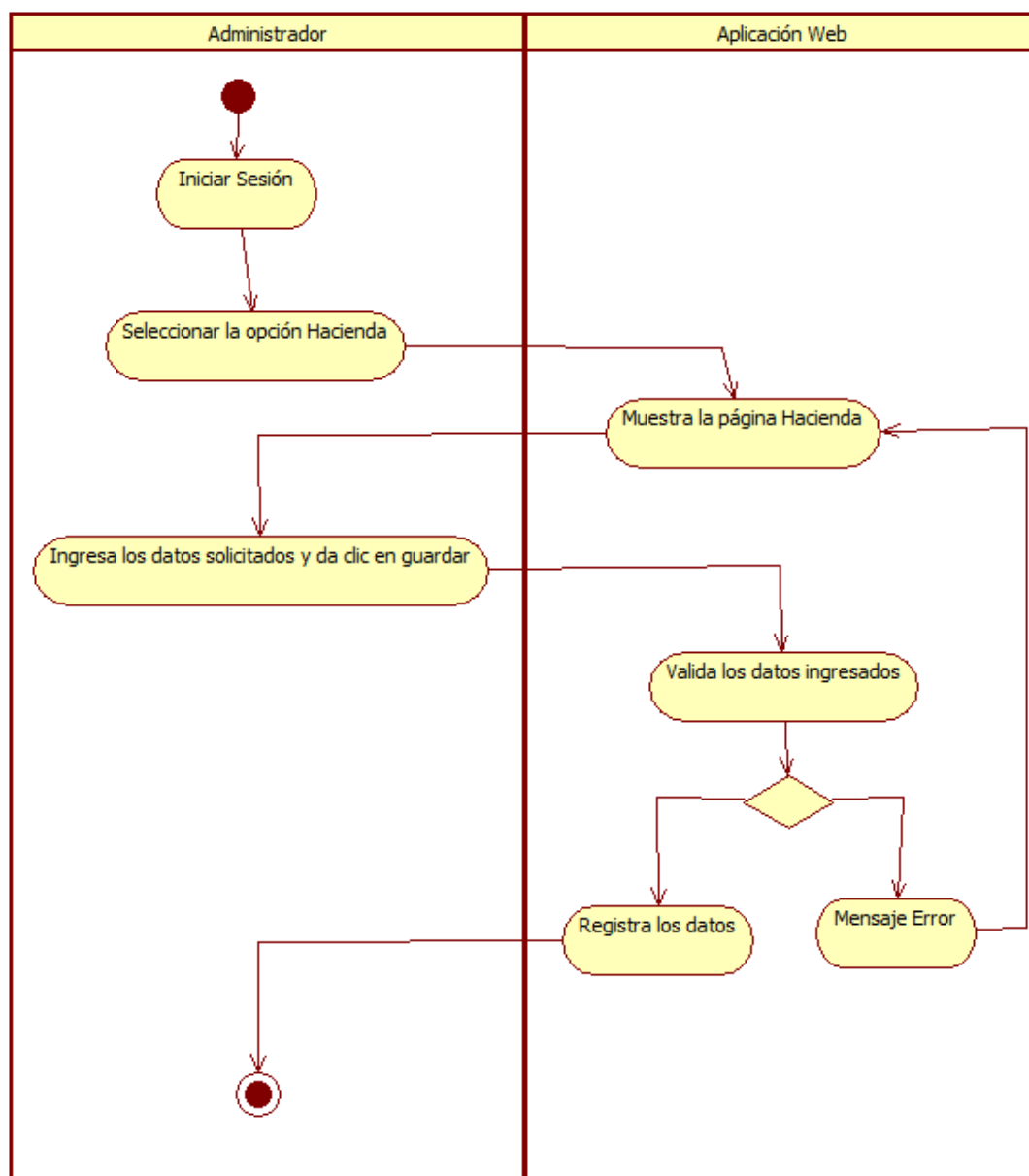


Ilustración 8 Diagrama de Actividades de Asignación de Lotes

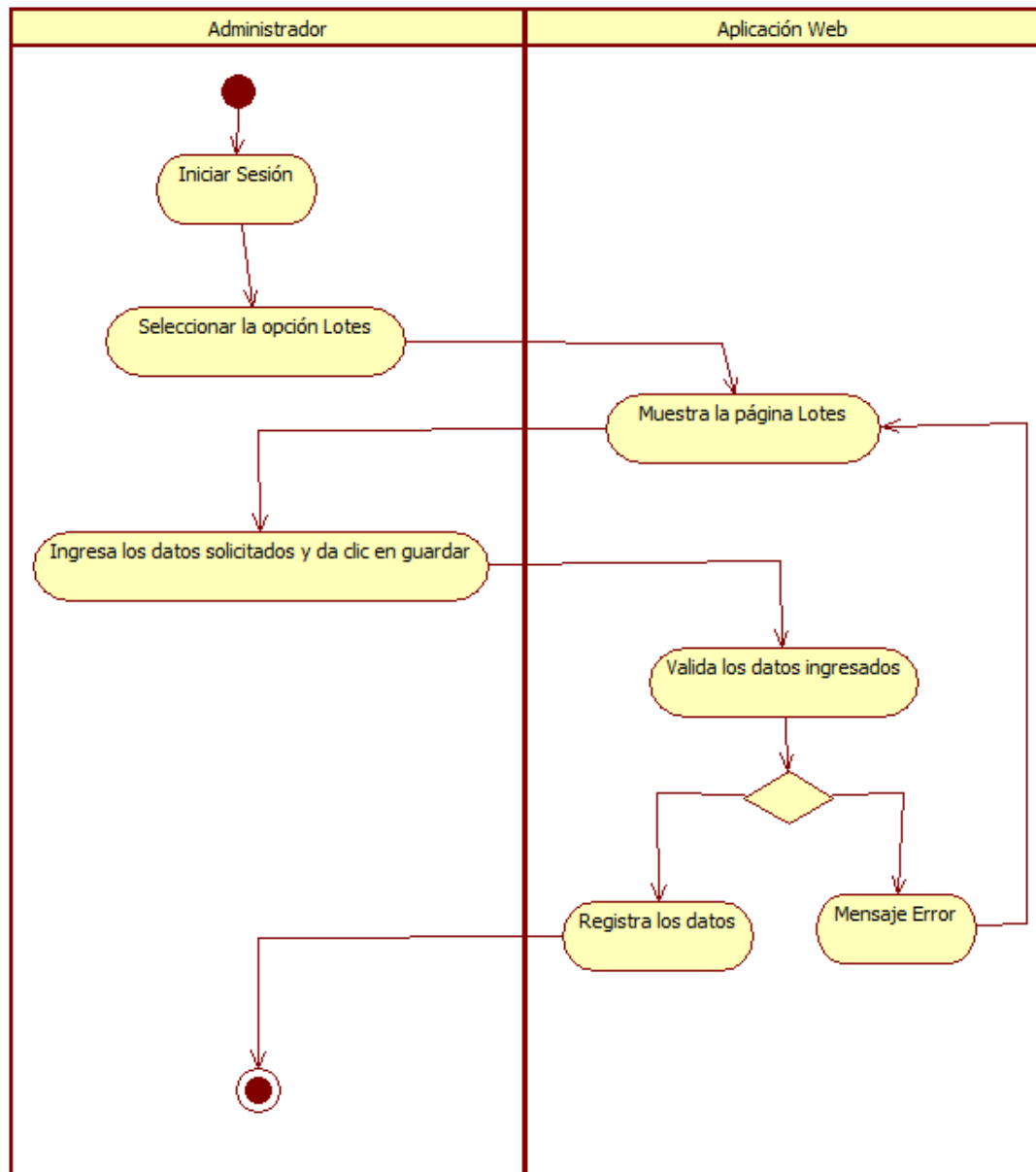


Ilustración 9 Diagrama de Actividades de Registrar Labores Agrícolas

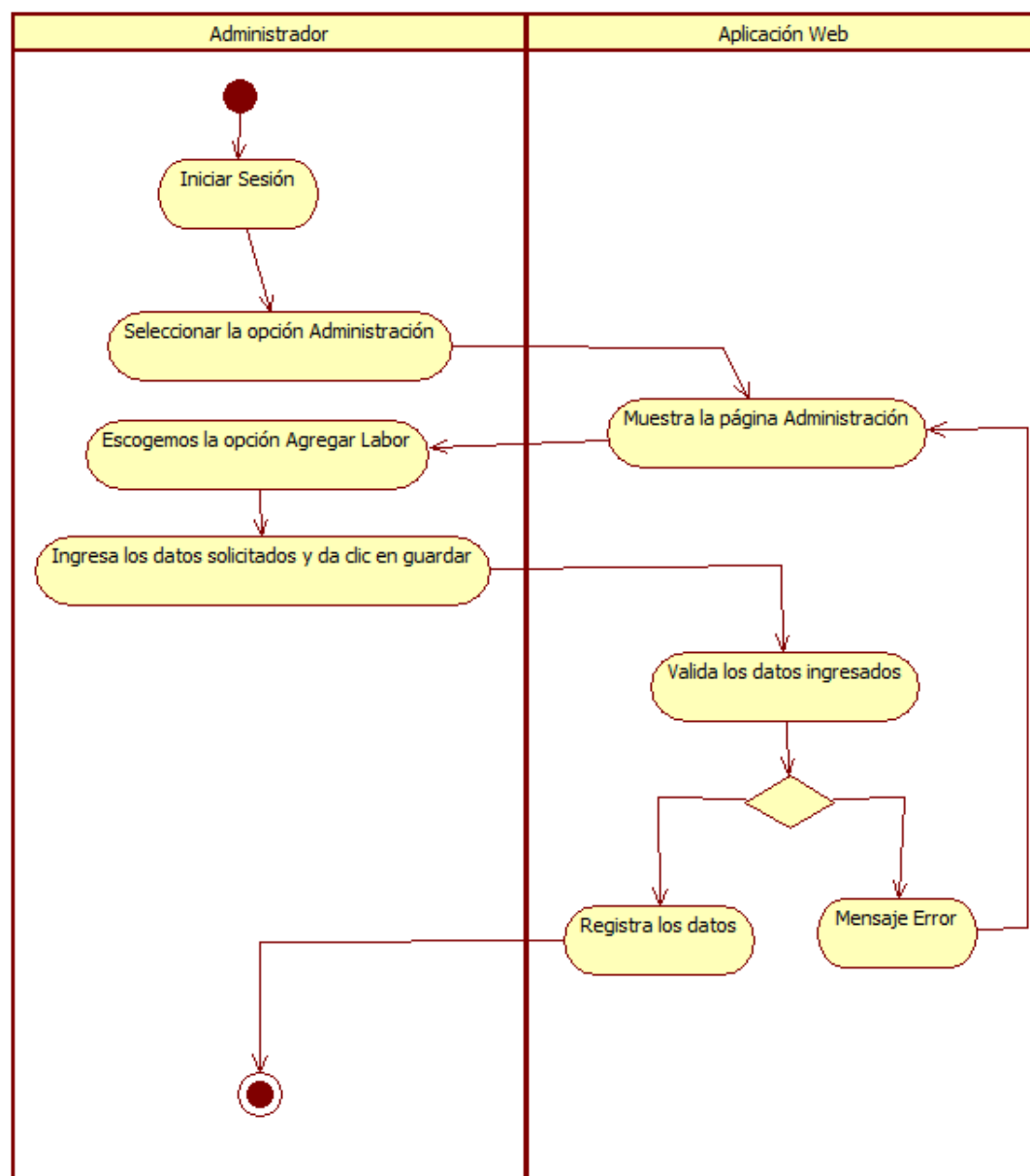


Ilustración 10 Diagrama de Actividades de Registrar Característica de Labores

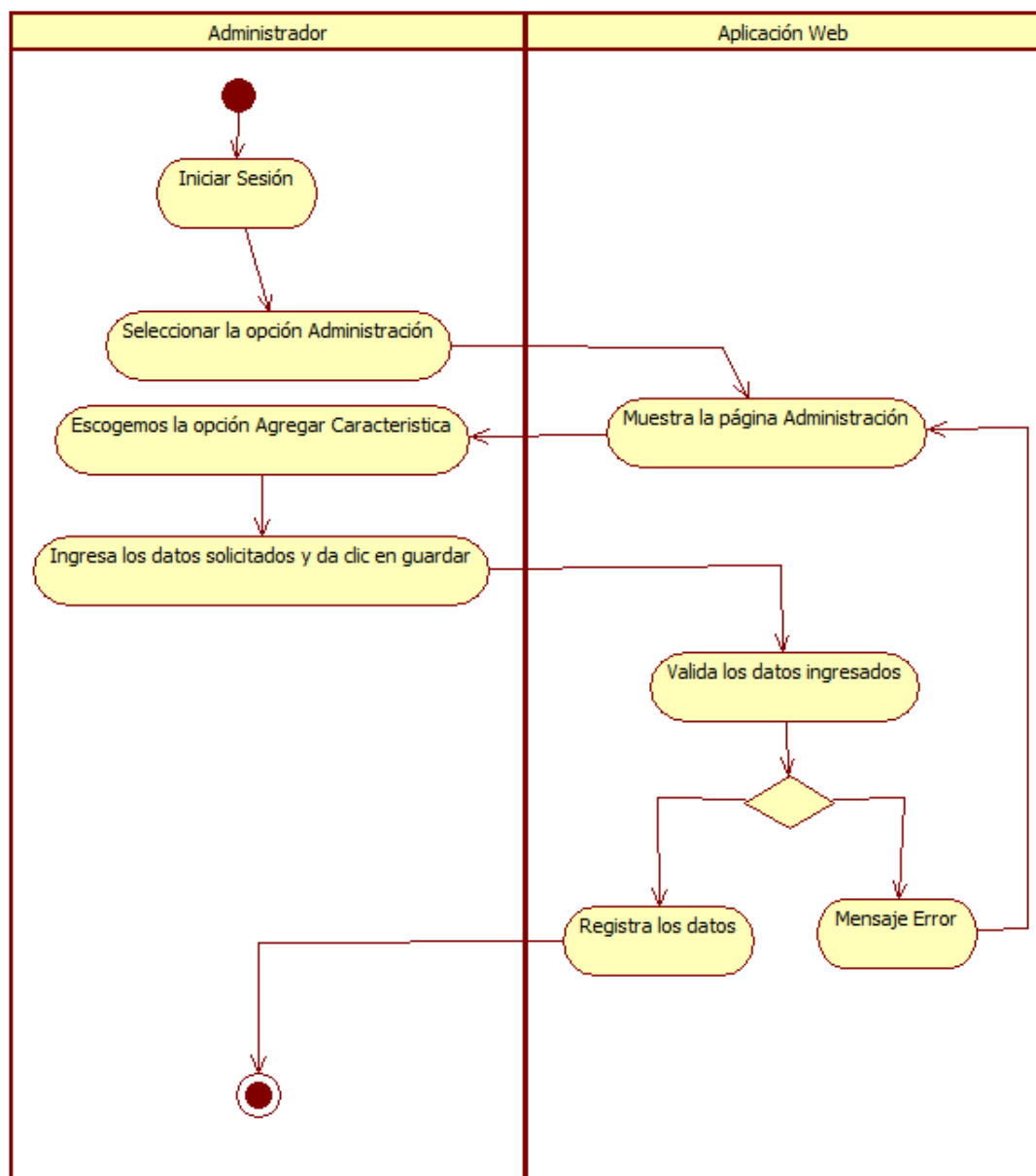


Ilustración 11 Diagrama de Actividades de Planificación de Labores

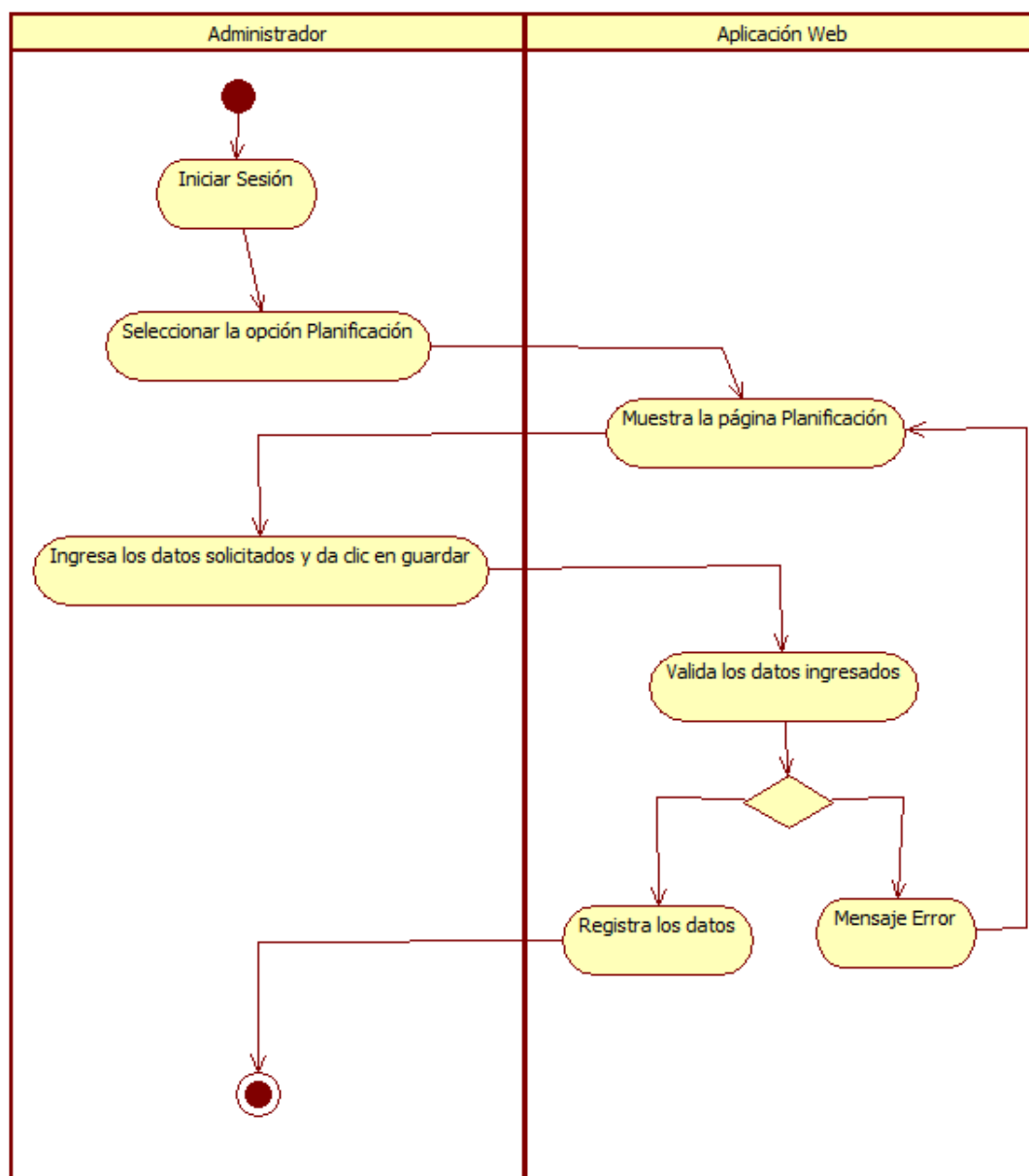


Ilustración 12 Diagrama de Actividades de Control de Planificación

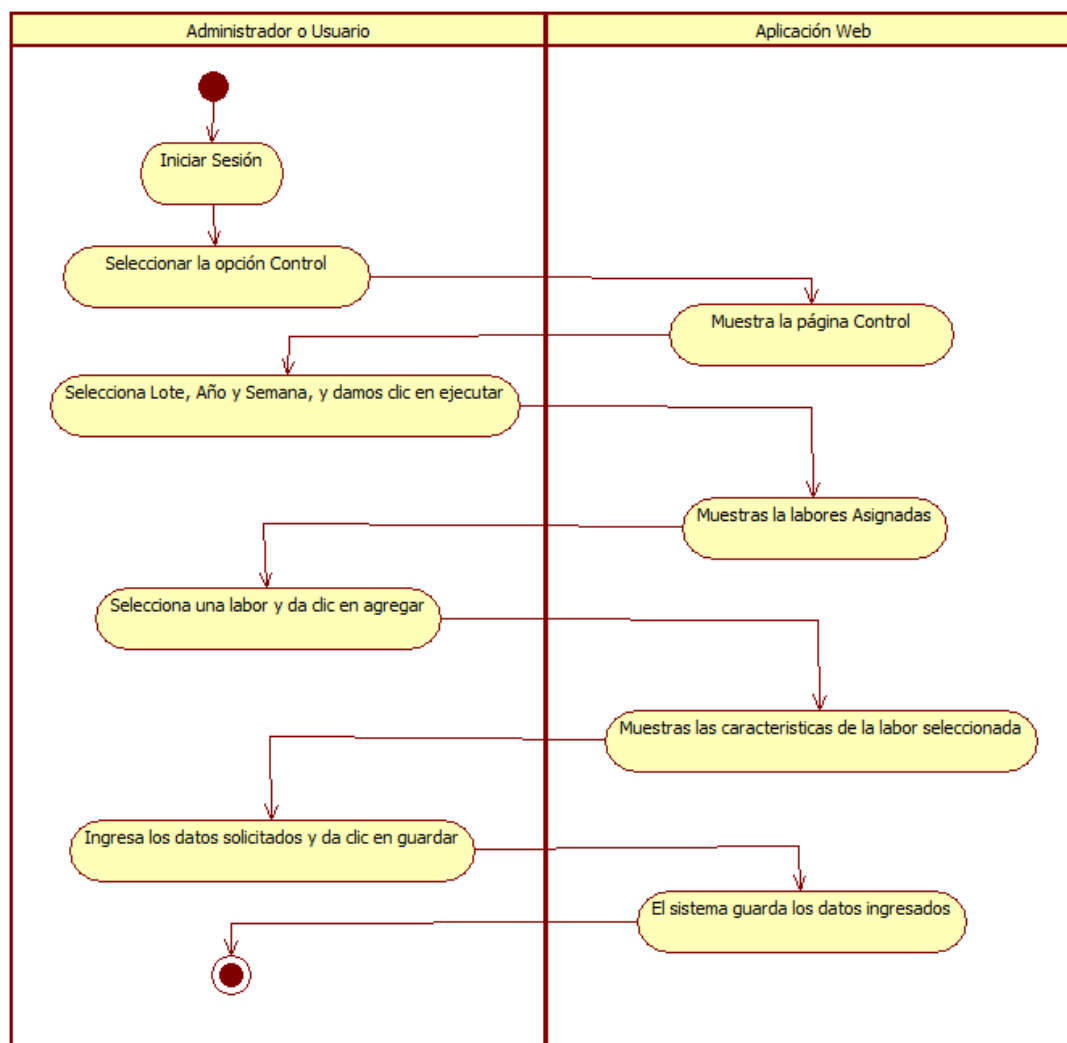


Ilustración 13 Diagrama de Actividades de Registrar Personal

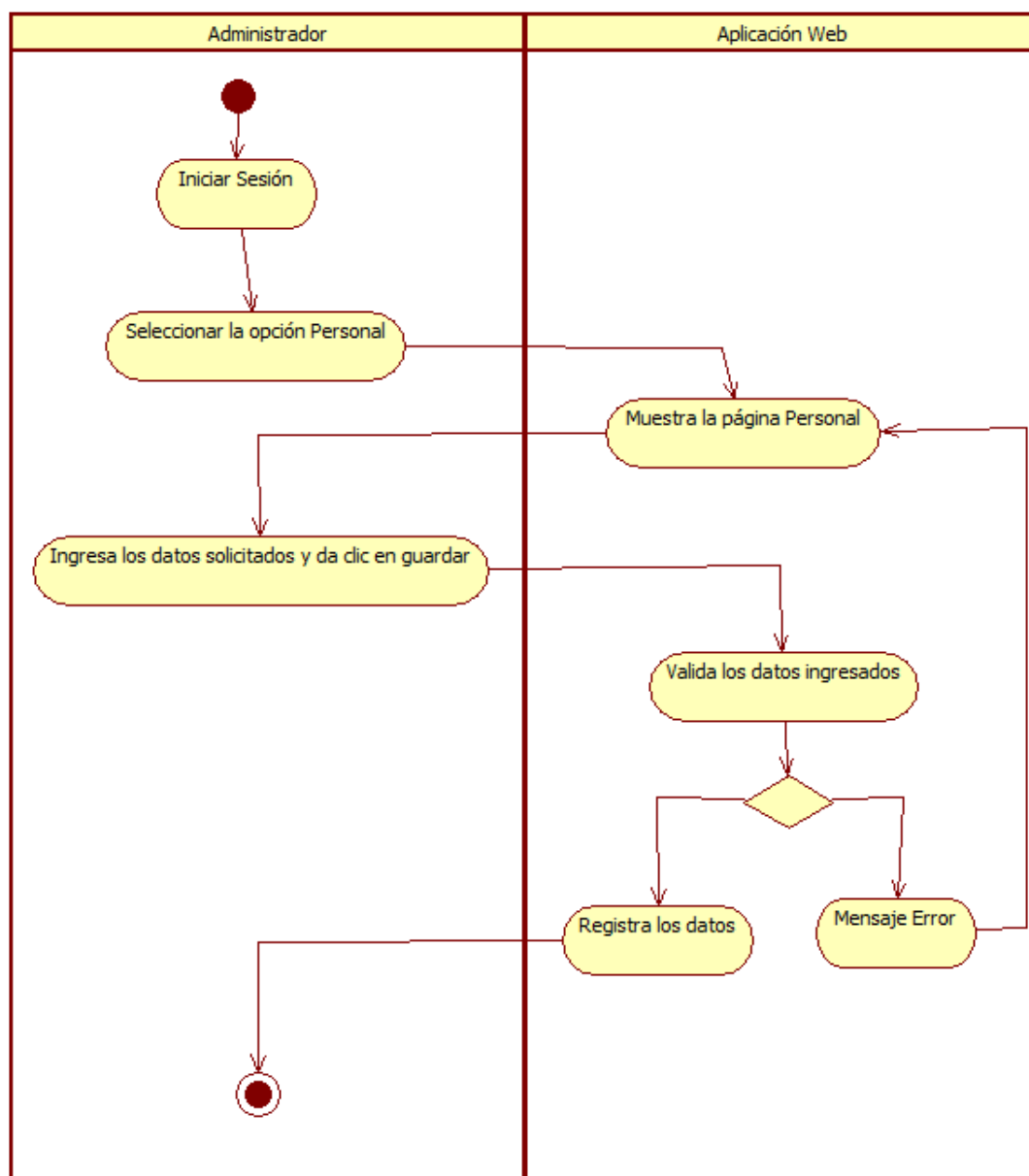


Ilustración 14 Diagrama de Actividades de Registrar Estimación

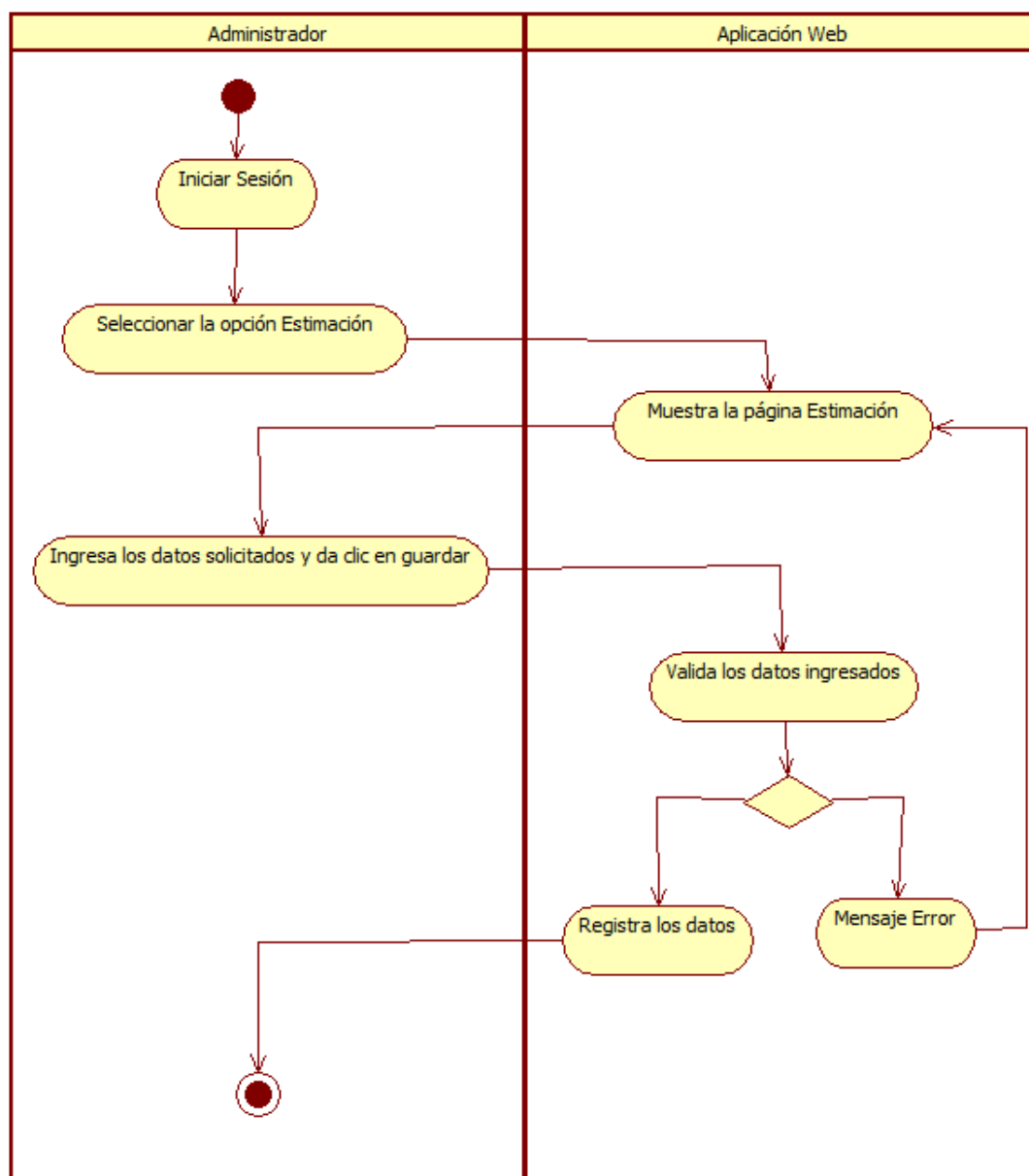


Ilustración 15 Diagrama de Actividades de Registrar Producción

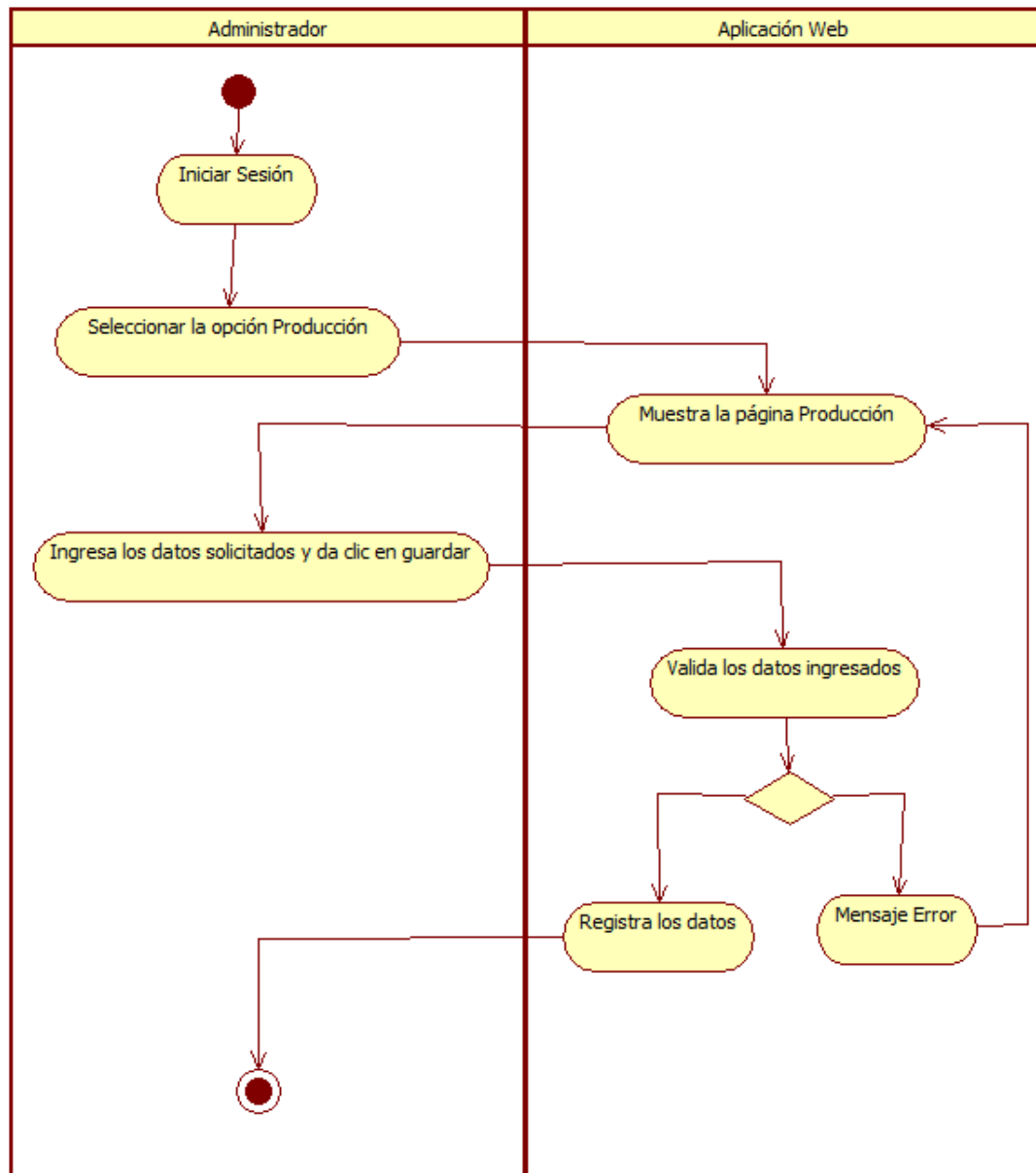


Ilustración 16 Diagrama de Actividades de Recuperar Clave

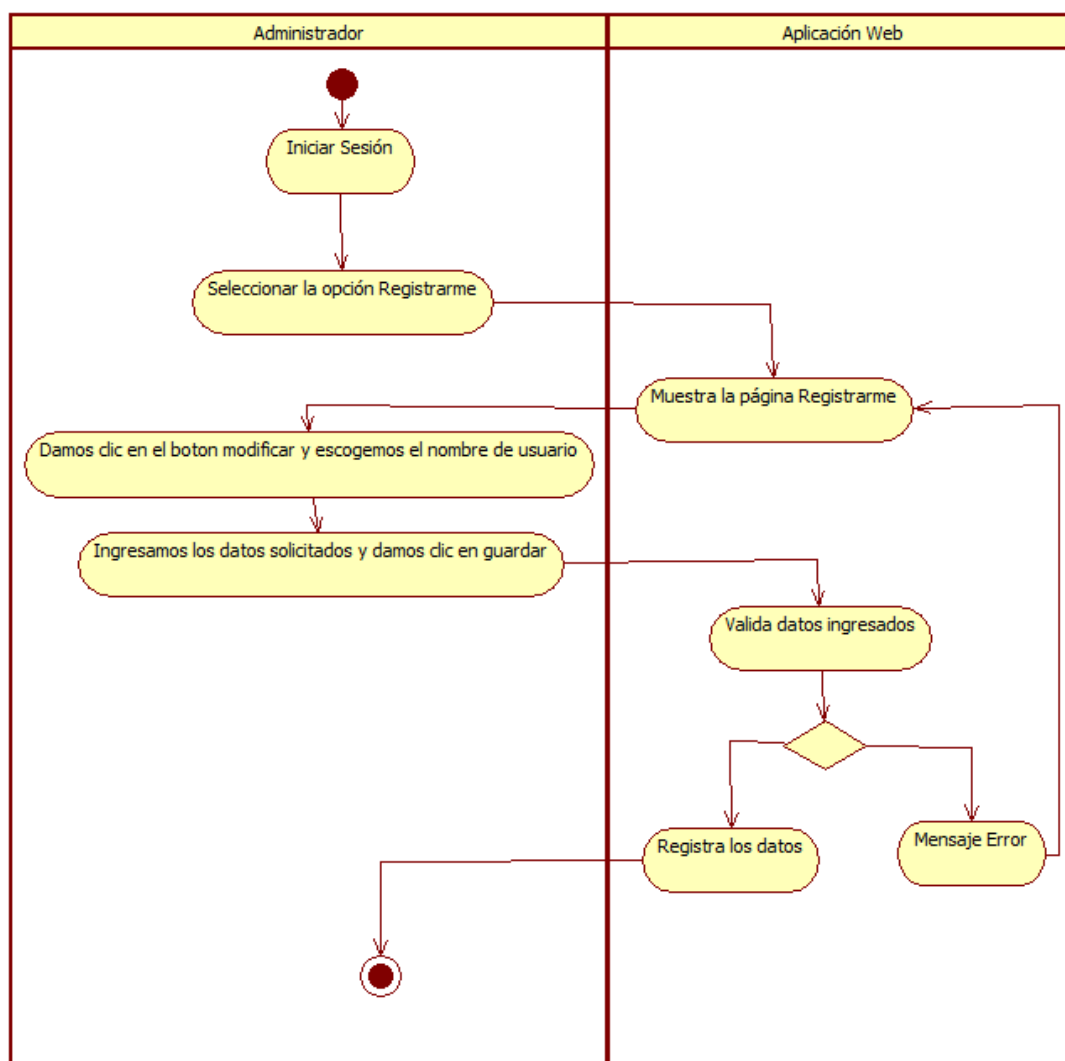
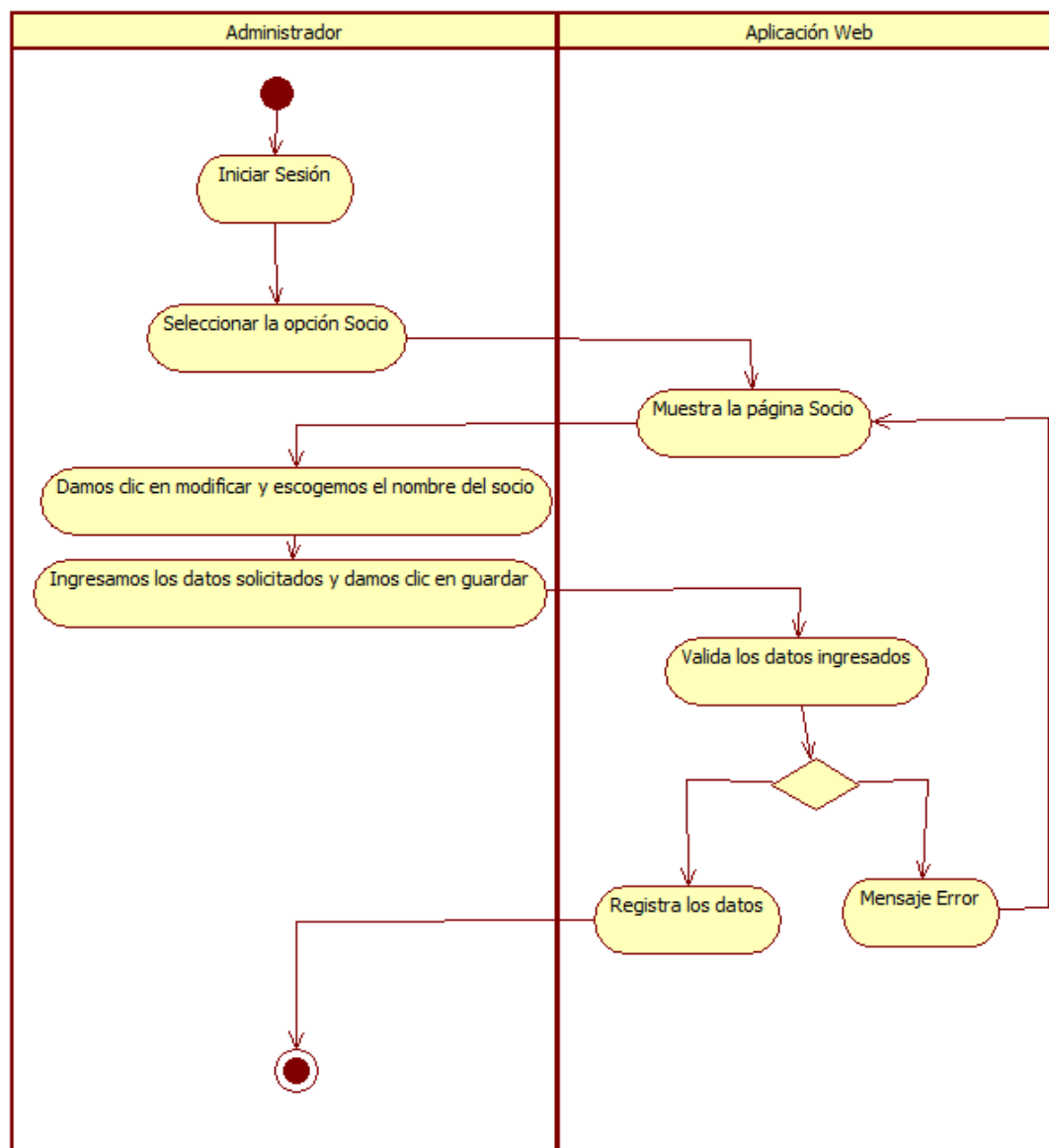


Ilustración 17 Diagrama de Actividades de Modificar Socio



4.1.2.3 Diagramas de Secuencia

Ilustración 18 Diagrama de Secuencia de Iniciar Sesión

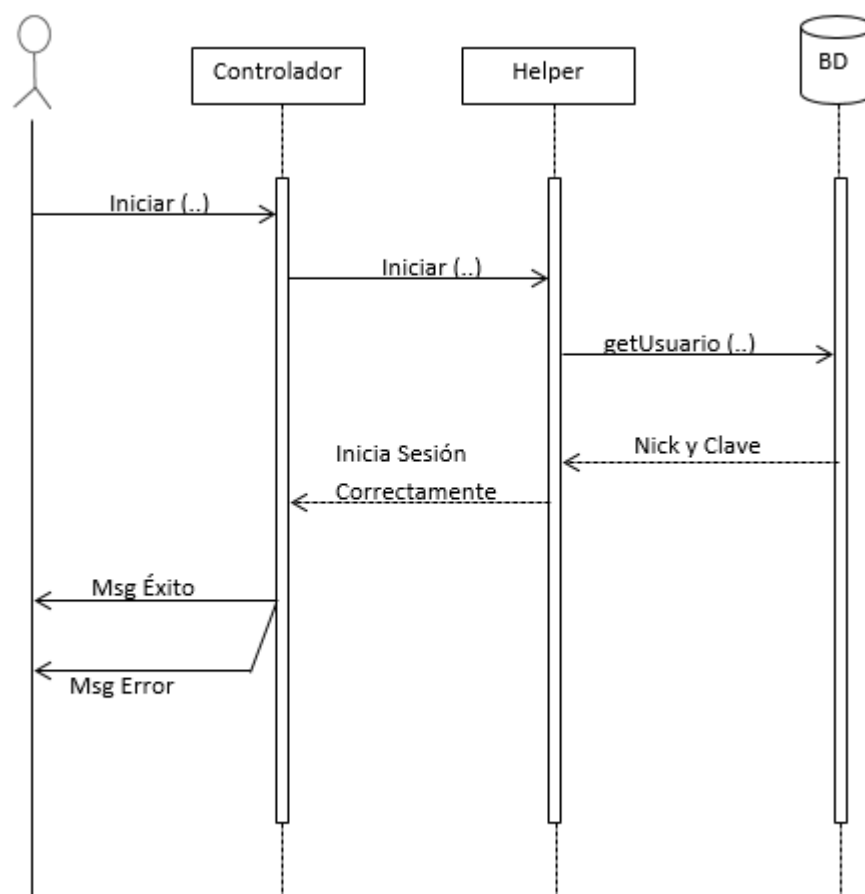


Ilustración 19 Diagrama de Secuencia de Registrar Usuario

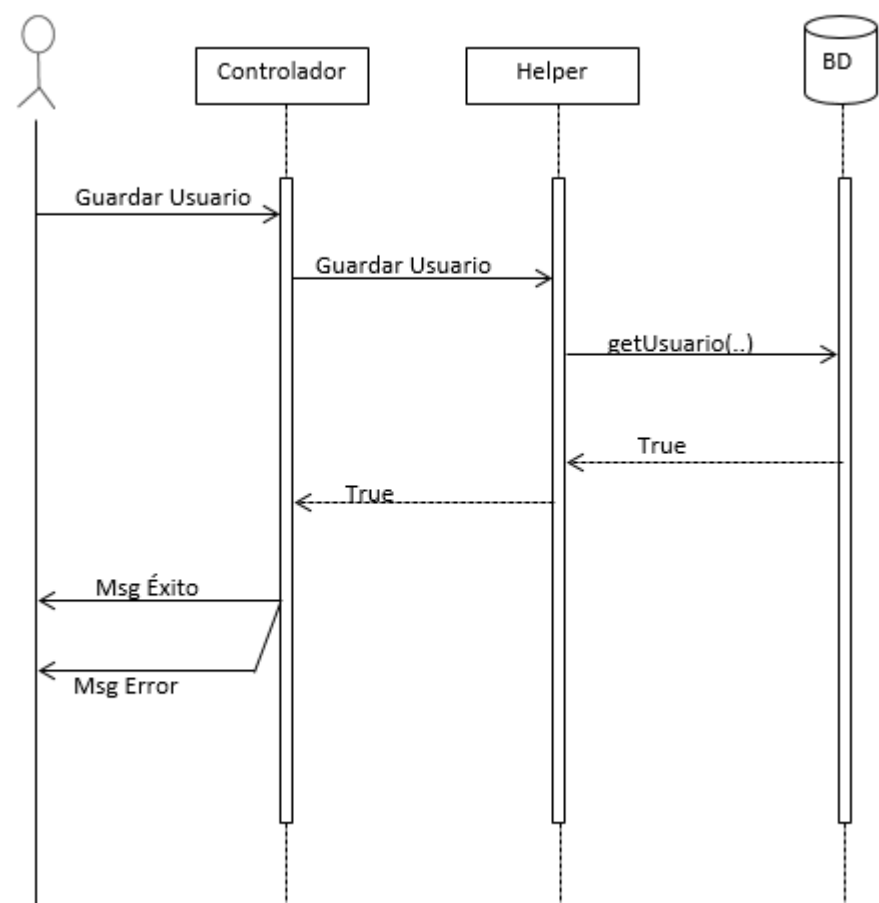


Ilustración 20 Diagrama de Secuencia de Registrar Hacienda

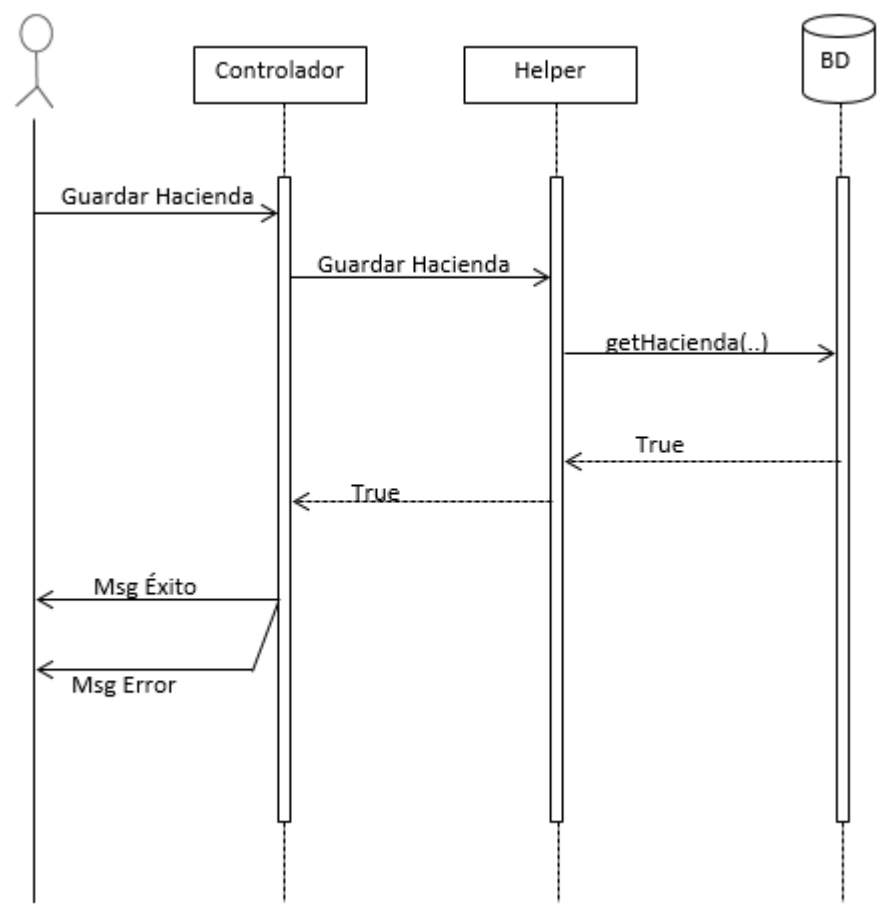


Ilustración 21 Diagrama de Secuencia de Asignación de Lotes

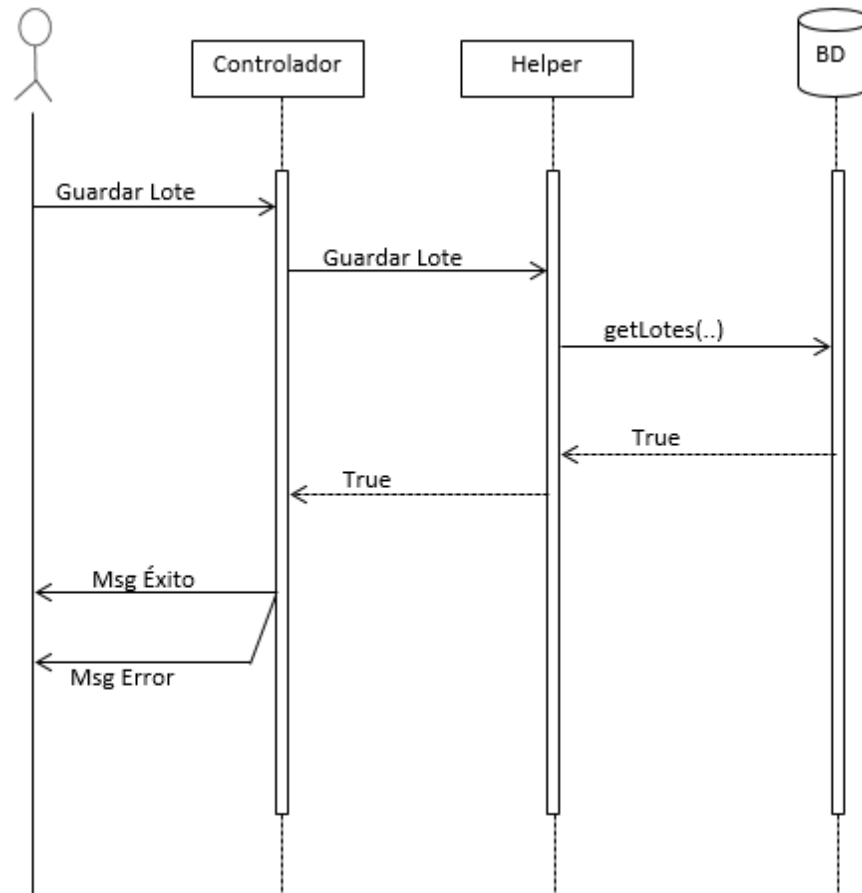


Ilustración 22 Diagrama de Secuencia de Registrar Labores Agrícolas

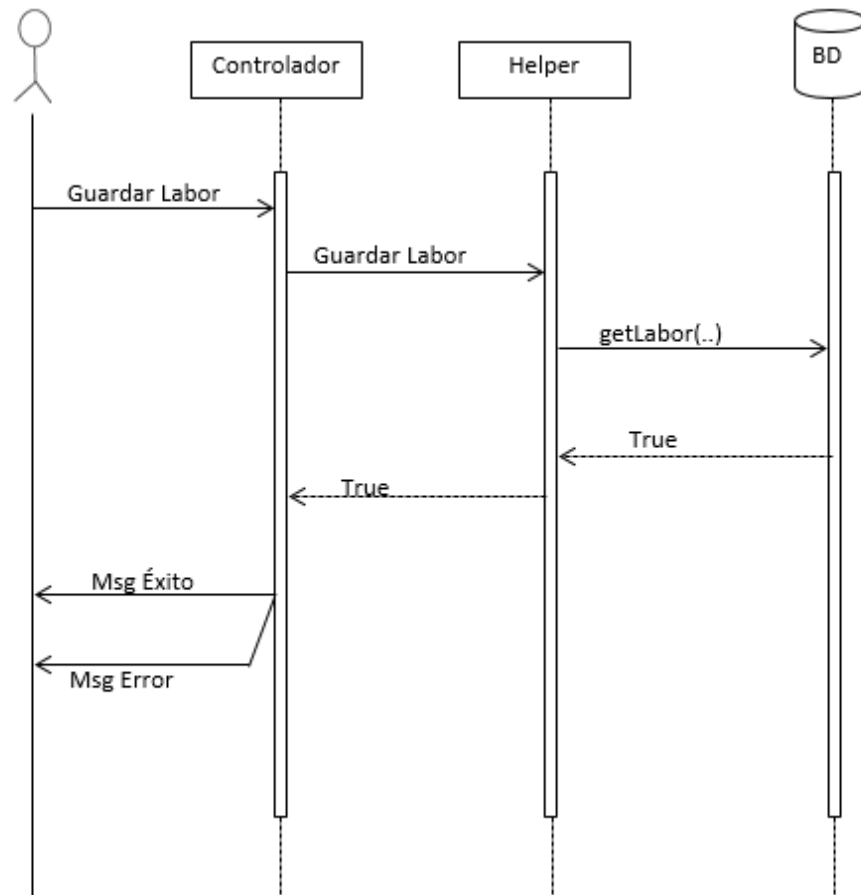


Ilustración 23 Diagrama de Secuencia de Registrar Característica de Labor

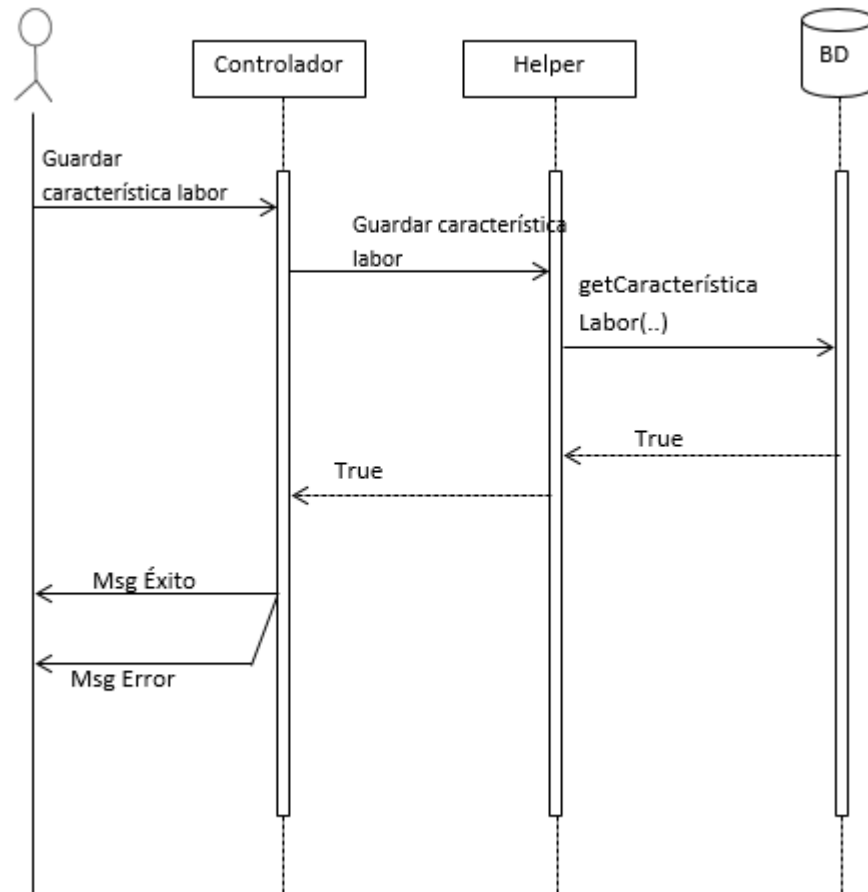


Ilustración 24 Diagrama de Secuencia de Registrar Socio

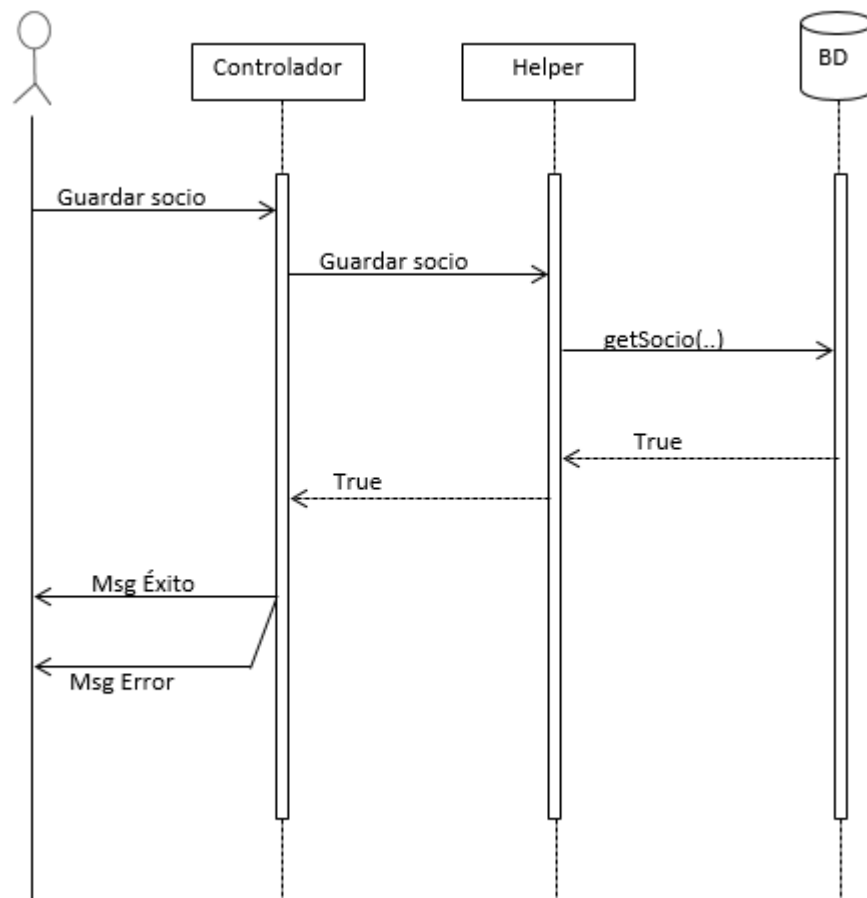


Ilustración 25 Diagrama de Secuencia de Registrar Planificación

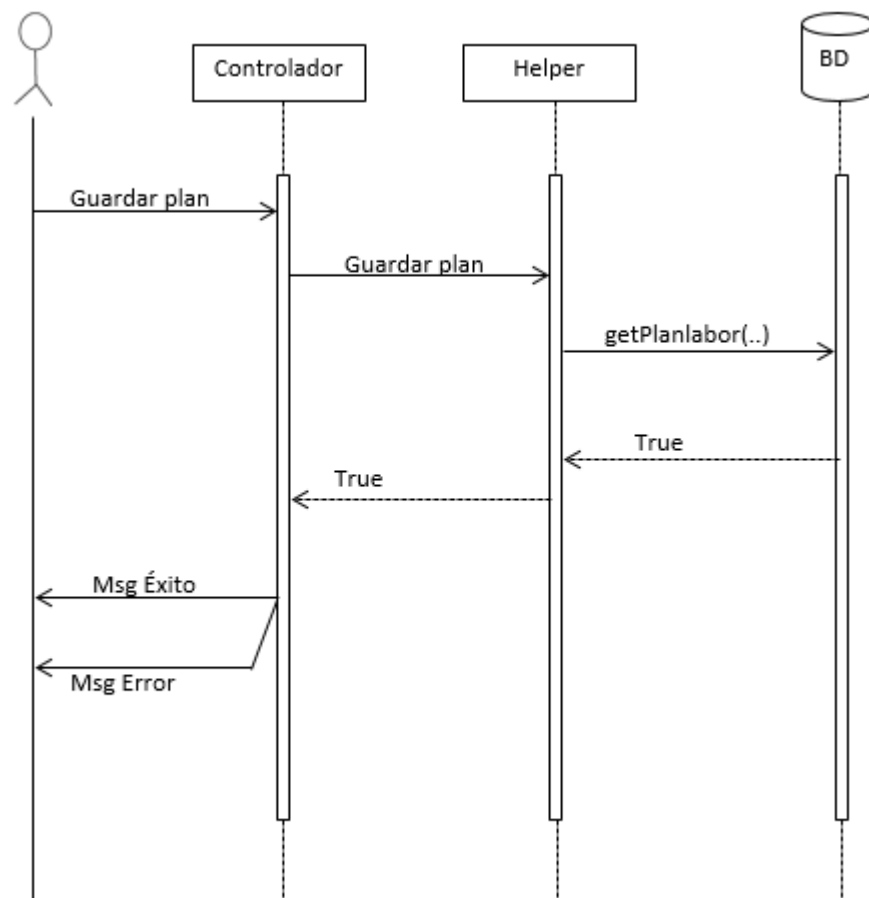


Ilustración 26 Diagrama de Secuencia de Control de Planificación

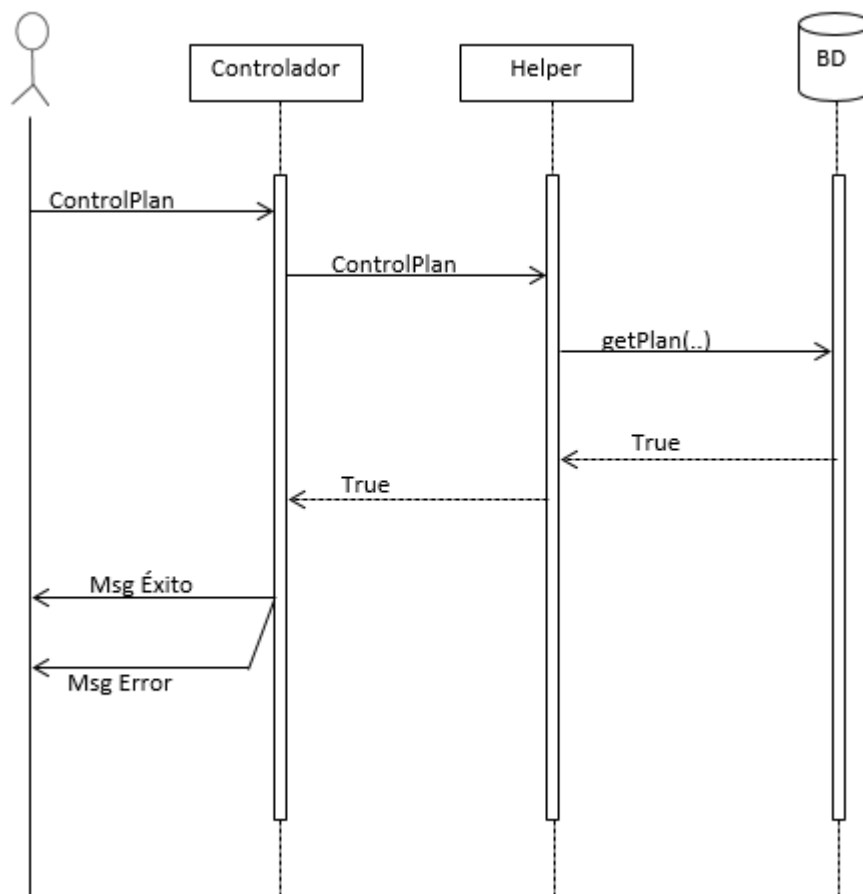


Ilustración 27 Diagrama de Secuencia de Registrar Personal

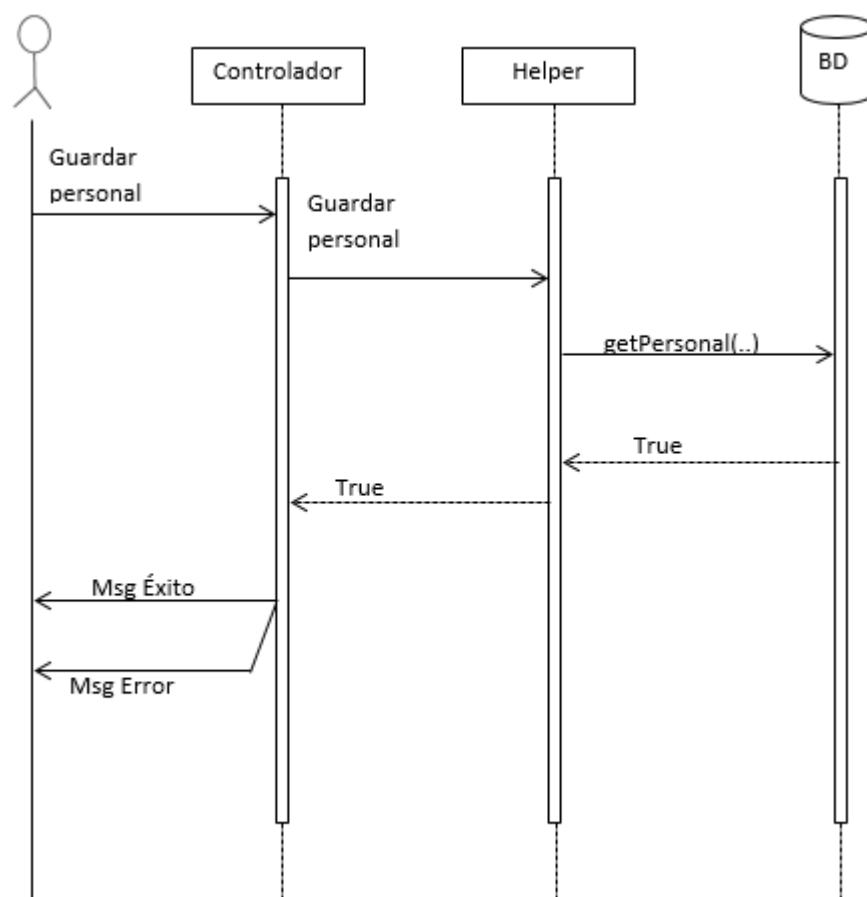


Ilustración 28 Diagrama de Secuencia de Registrar Estimación

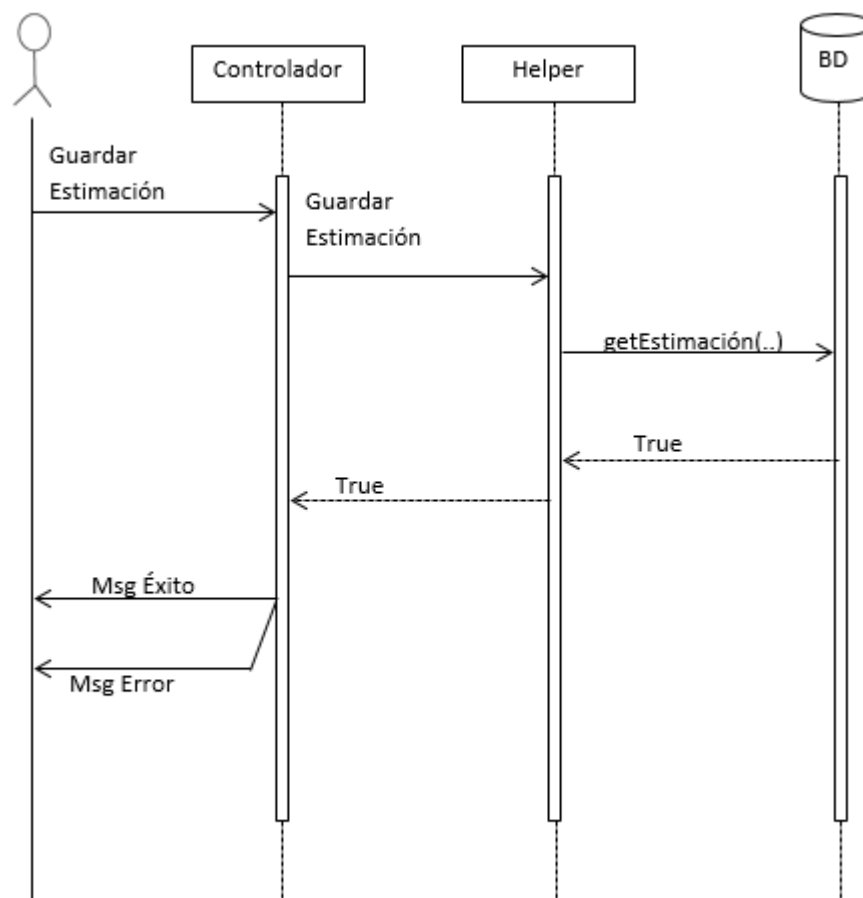


Ilustración 29 Diagrama de Secuencia de Registrar Producción

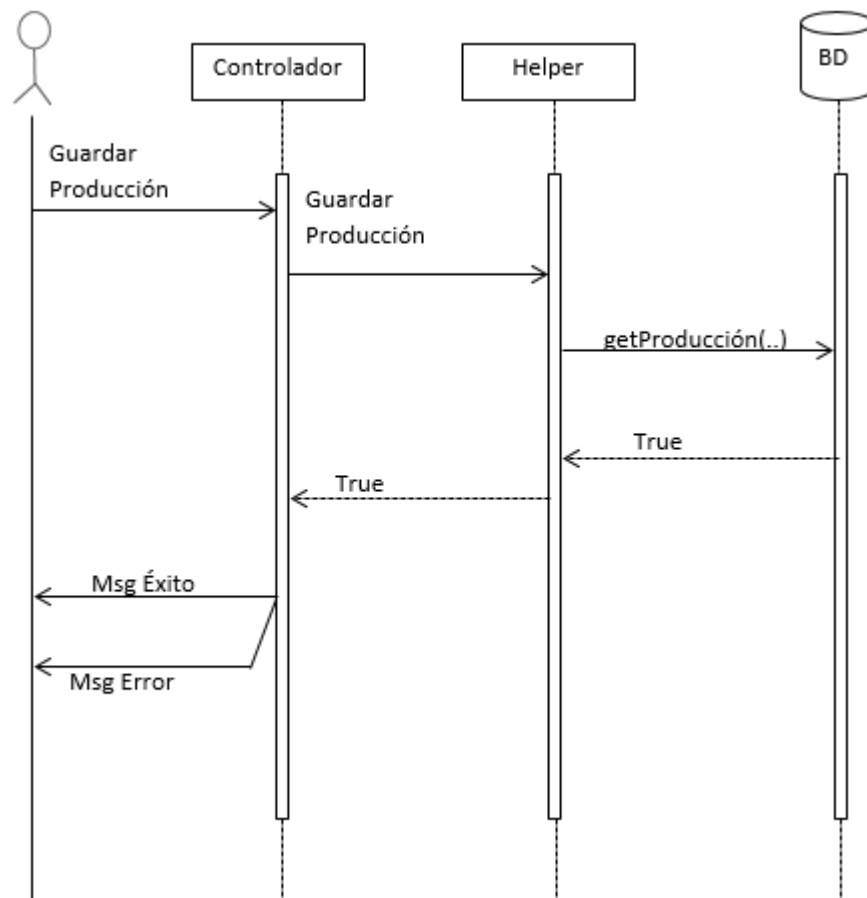


Ilustración 30 Diagrama de Secuencia de Recuperar Clave

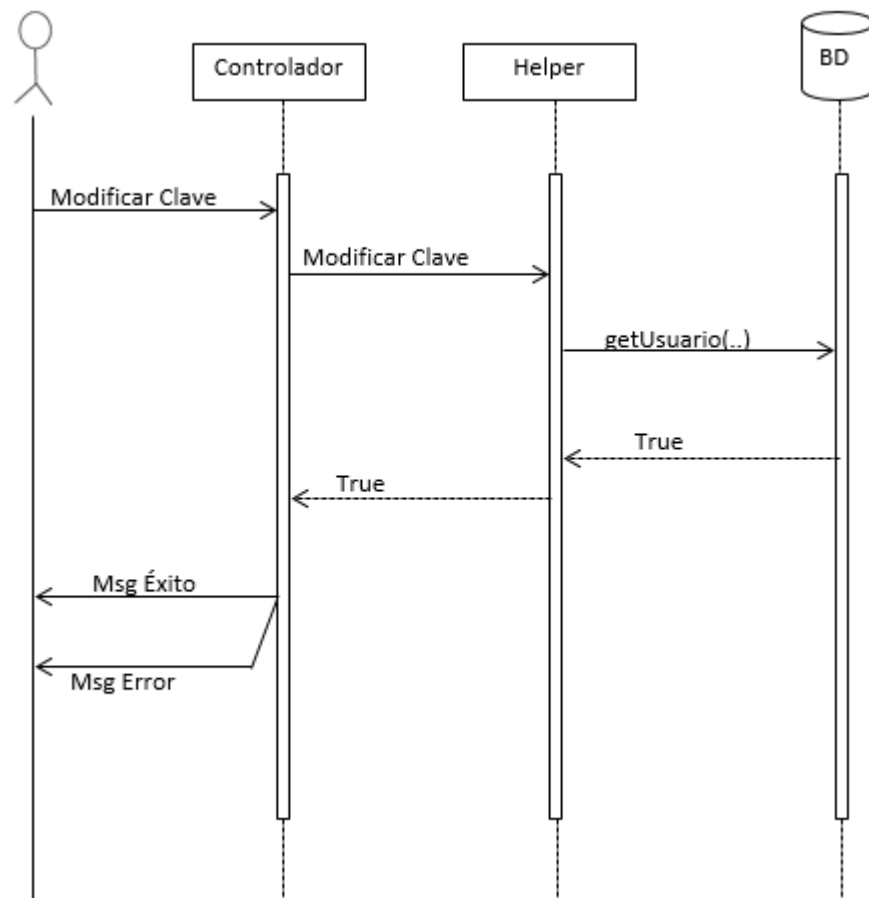
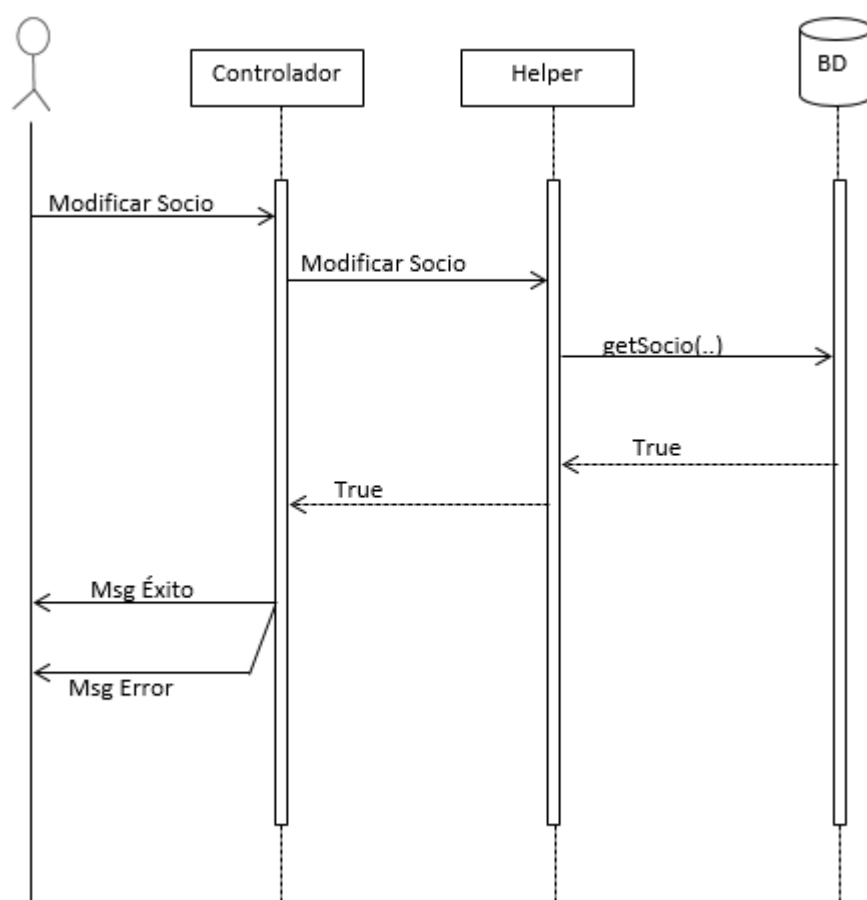
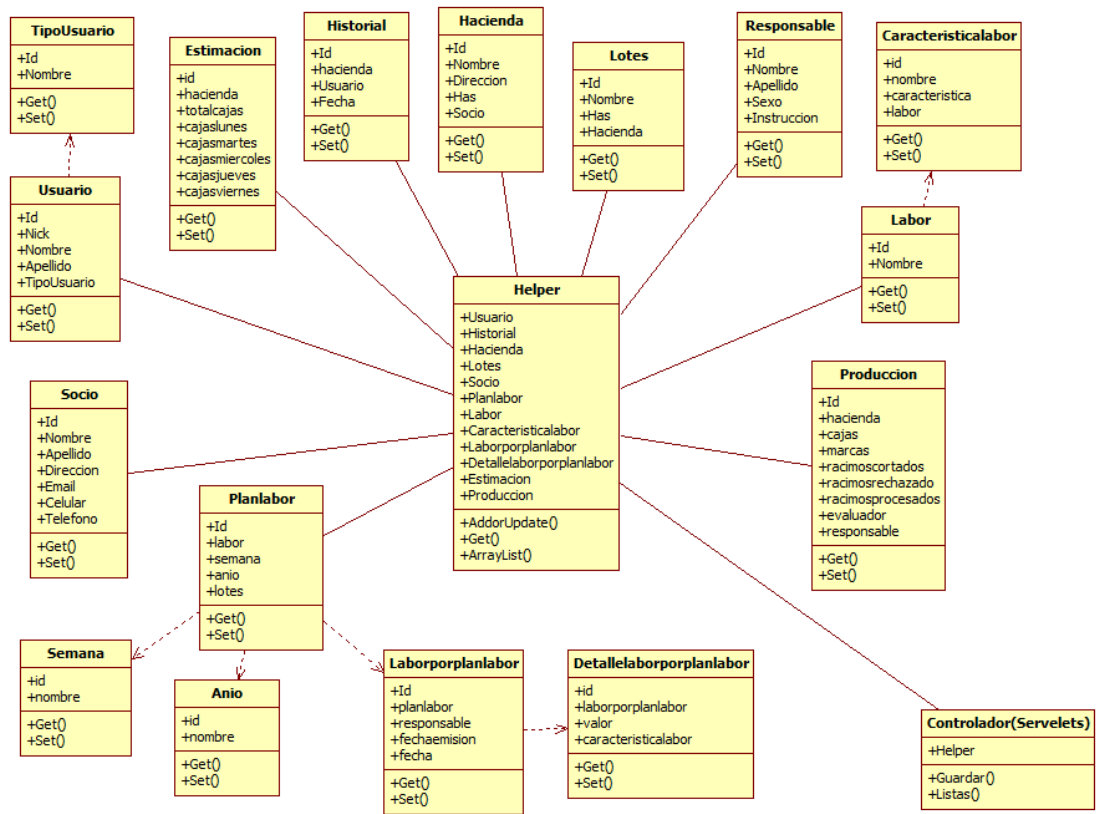


Ilustración 31 Diagrama de Secuencia de Modificar Socio



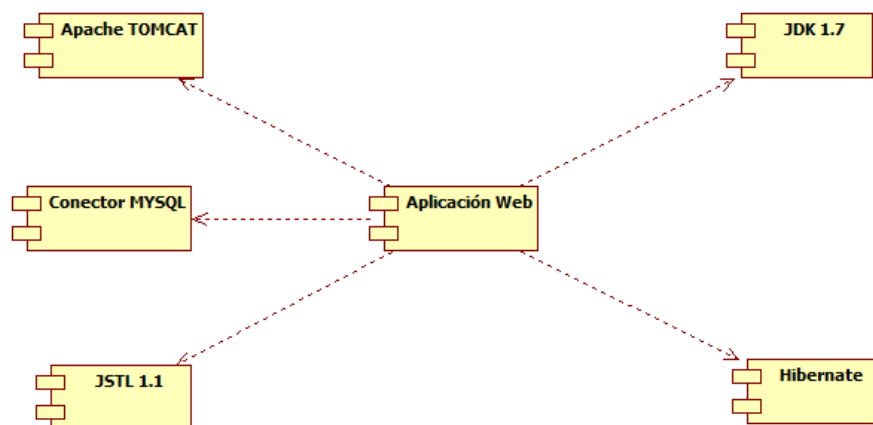
4.1.2.4 Diagrama de Clase

Ilustración 32 Diagrama de Clase



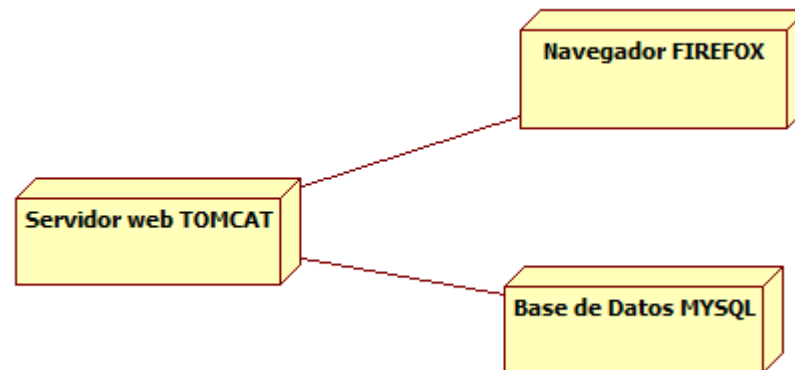
4.1.2.4 Diagrama de Componente

Ilustración 33 Diagrama de Componente



4.1.2.5 Diagrama de Despliegue

Ilustración 34 Diagrama de Despliegue

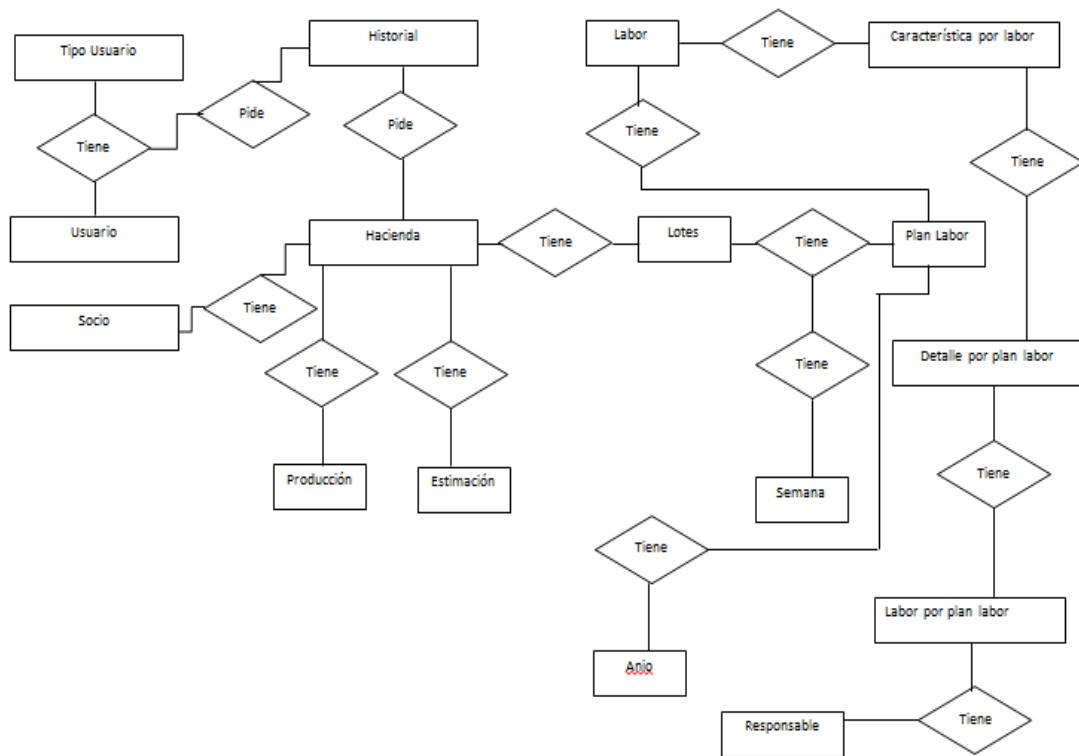


4.1.2 Construcción

4.1.2.1 Capa 1 Base de Datos.

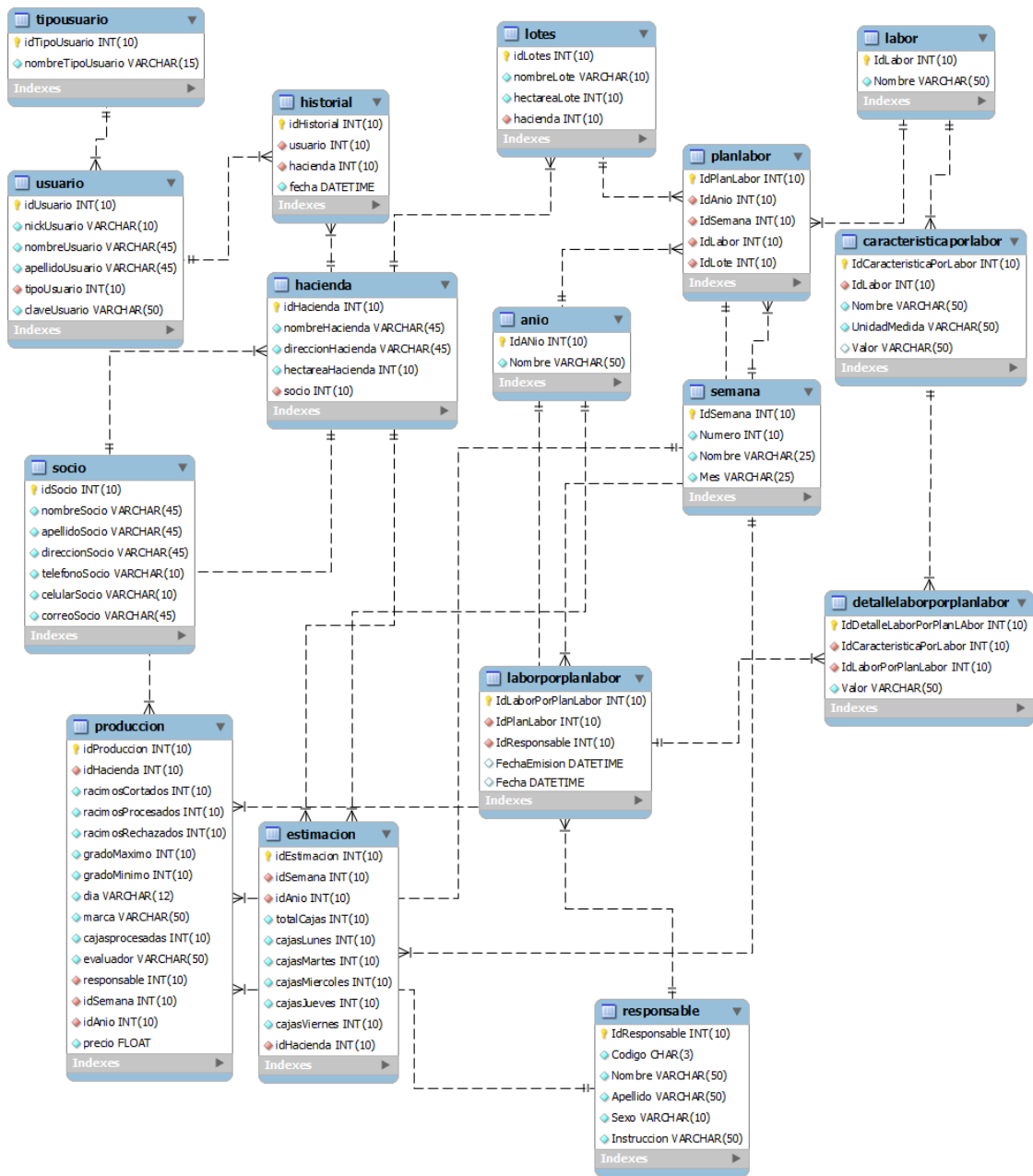
Para la elaboración de la base de datos de nuestra aplicación se utilizó el entorno de MySQL 5.0, primeramente se desarrolló un diseño físico de la BD. Como se muestra en la ilustración 35.

Ilustración 35 Diseño Lógico de la BD



Una vez definido nuestro diseño físico procedemos a elaborar nuestro modelo Entidad – Relación, se utilizó la herramienta MySQL Workbench 5.2 CE. En este modelo se detalla todas las tablas con sus respectivos campos se muestra en la ilustración 36.

Ilustración 36 Diseño Físico de la BD



4.1.2.2 Capa 2 Reglas de Negocio.

En el desarrollo de esta etapa se utilizó el framework Hibernate, que ayuda en el enlace entre la aplicación y la Base de Datos; facilita el mapeo de los atributos de la BD y el modelo de objetos de una aplicación MVC. Se utilizó varios componentes que a continuación se detalla:

- ✚ Conector MySQL.
- ✚ Hibernate
- ✚ JSTL 1.1
- ✚ JDK 1.7
- ✚ Apache Tomcat 7.0.34.0
- ✚ JasperReport.

El mapeo generado por Hibernate por cada tabla crea dos archivos:

- ✚ Archivo.java.- Con propiedades accesible mediante los métodos setter y getter.
- ✚ Archivo.hbm.xml.- Se usa para mapearlo a la BD. Se declaran las propiedades del POJO y sus correspondientes columnas de la BD y las relaciones entre las tablas.

Las clases de datos utilizadas en el desarrollo de esta aplicación son:

- ✚ Anio.java – Anio.hbm.xml.
- ✚ Caracteristicaporlabor.java – Caracteristicaporlabor.hbm.xml.
- ✚ Detallelaborporplanlabor.java – Detallelaborporplanlabor.hbm.xml.
- ✚ Estimacion.java – Estimacion.hbm.xml.
- ✚ Hacienda.java – Hacienda.hbm.xml.
- ✚ Helper.java.
- ✚ HibernateUtil.java.
- ✚ Historial.java – Historial.hbm.xml.
- ✚ Labor.java – Labor.hbm.xml.
- ✚ Laborporplanlabor.java – Laborporplanlabor.hbm.xml.

- + Lotes.java – Lotes.hbm.xml.
- + Planlabor.java – Planlabor.hbm.xml.
- + Produccion.java – Produccion.hbm.xml.
- + Responsable.java – Responsable.hbm.xml.
- + Semana.java – Semana.hbm.xml.
- + Socio.java – Socio.hbm.xml.
- + Tipodeusuario.java – Tipodeusuario.hbm.xml.
- + Usuario.java – Usuario.hbm.xml.
- + Hibernate.reveng.xml.
- + Controlador.java (Servlets).
- + OpcionMenu.java.
- + FechaActual.java.
- + UtilsSeguridad.java

De las clases de datos enunciados las de mayor utilidad son las HibernateUtil.java que sirve para crear el objeto SessionFactory, generando un método estático SessionFactory getSessionFactory(), que permite acceder a SessionFactory desde cualquier clase como si se tratara de un objeto global.

Hibernate.reveng.xml es la ingeniería inversa que le aplicamos a nuestra BD, es aquí donde se generan las clases POJO y mapeos XML.

UtilsSeguridad.java es la que utilizamos para la encriptación de las claves al momento de almacenar en la BD, y desencriptarla para comparar si es igual a la clave ingresada por el usuario al momento de iniciar sesión.

OpcionMenu.java es la que valida las diferentes opciones para cada tipo de usuario de nuestra aplicación.

Fechaactual capturar la fecha al momento de almacenarla en la BD.

Helper.java esta es nuestra clase maestra desde aquí se opera todos los accesos a nuestra BD. Aquí se forma todas las sentencias Hibernate Query Language (HQL). Se detallan los métodos utilizados:

- + addOrUpdate. Este método nos permite realizar todos los ingresos a nuestra BD.
- + ArrayList. Este método por medio de una sentencia HQL nos devuelve en una lista los datos de la BD.
- + get. Este método por medio de una sentencia HQL nos devuelve un valor de la BD.
- + delete. Este método nos permite borrar registros de nuestra BD.

Cabe recalcar que para cada ingreso o consulta con cualquiera de los métodos anteriormente mencionado se debe generar individualmente.

El servlet depende de los métodos que se hayan registrado en la clase Helper, es aquí donde validamos el inicio de sesión y las opciones para los tipos de usuario; es aquí donde se genera toda la lógica de nuestra aplicación donde se valida todos los ingresos y se generan las listas. Es aquí donde el servlet genera los hilos de validaciones y las excepciones. Se detalla los métodos que tiene el servlet.

Tabla 5 Tabla de métodos utilizados en la Aplicación Web

Métodos	
Nombres	Descripción
CargarElemento	Carga la lista de hacienda para poder iniciar sesión.
Iniciar	Valida la hacienda, usuario y clave para ingresar a la aplicación.
OpciónMenú	Obtiene el tipo de usuario para mostrar las opciones
Cerrarsesión	Cierra la aplicación

GuardarUsuario	Almacena los datos de usuarios en la BD.
GuardarHacienda	Almacena los datos de hacienda en la BD.
GuardarLote	Almacena los lotes por hacienda en la BD.
OpciónLote	Muestra la lista de lotes asignado a la hacienda.
GuardarPersonal.	Guarda los datos de personal en la BD.
GuardarLabor	Almacena los código y nombre de la labores en la BD.
OpciónLabor	Muestra las labores registradas en la aplicación.
OpciónCaracterística	Muestras la características
GuardarCaracterística	Almacena las características por cada labor.
Guardar Plan.	Almacena la semana, año, lote y labor
Opción	Almacena el idPlanlabor con el responsable y las características de la labor.
ConsultarPlan	Muestra las labores planificada por semana.
_Labores	Muestra las características de labor guardada en la planificación.
ConsultarCaracterística	Consulta las características de una labor seleccionada.
GuardarEstimación	Almacena la posible producción de la hacienda.
GuardarProducción	Almacena la producción generada en la hacienda
Opciónplanlabor	Muestra las opciones para guardar un plan.
doGet	Genera hilos con sus respectivas excepciones.

doPost	Genera hilos con sus respectivas excepciones.
LimpiarMensaje	Ponen en blanco todos los mensajes que genera nuestra aplicación.

4.1.2.3 Capa 3 Vista

En esta etapa se realizó el diseño de nuestra aplicación. Esta será la parte visible al usuario, se utilizó código HTML y CSS para el diseño. Se creó javascript para las validaciones de la aplicación.

En la mayoría de las páginas se utilizó el método Post para que cierta información no sea visible al usuario en el navegador. Para poder acceder a nuestra aplicación es necesario que se cargue la lista de hacienda y con un usuario registrado en el sistema.

4.2. RESULTADO DE COMPROBACIÓN DE HIPÓTESIS

Con la finalidad de comprobar la hipótesis, se ha establecido aplicar la prueba T-Student, para determinar si existe una mayor eficiencia; evaluando los siguientes indicadores:

- 📊 Tasa de pérdida de documentos.
- 📊 Control continuo del manejo agrícola.

4.2.1 Tasa de Pérdida de Documentos.

Para saber la cantidad de veces que se ha perdido documentos, se encuestó a 10 administradores al azar pertenecientes a APROBAVEN que hayan entregado documentos en oficina.

Tabla 6 Datos de documentos perdidos

N°	# documentos perdidos	# documentos perdidos
1	5	3
2	4	2
3	6	1
4	8	0
5	6	1
6	7	2
7	4	3
8	6	1
9	5	1
10	4	0
Σ	55	14
Media	5,5	1,4

Tabla 7 Datos para la prueba T - Student

Rango	antes	después	d	d-m	(d-m)^2
1	5	3	2	2,1	4,41
2	4	2	2	2,1	4,41
3	6	1	5	-0,9	0,81
4	8	0	8	-3,9	15,21
5	6	1	5	-0,9	0,81
6	7	2	5	-0,9	0,81
7	4	3	1	3,1	9,61
8	6	1	5	-0,9	0,81
9	5	1	4	0,1	0,01
10	4	0	4	0,1	0,01
sumatoria	55	14	41	0	36,9
media	5,5	1,4	4,1		

Calculamos la media aritmética

$$d = \frac{\sum d}{N} \quad d = \frac{41}{10} \quad d = 4.1$$

Calculamos la desviación estándar

$$\times d = \sqrt{\frac{\sum (d - d)^2}{N - 1}} \quad \times d = \sqrt{\frac{36.9}{10 - 1}} \quad \times d = 2.0248$$

Prueba T- Student

$$T = \frac{d}{\frac{\times d}{\sqrt{N}}} \quad T = \frac{4.1}{\frac{2.0248}{\sqrt{10}}} \quad T = 6.40$$

Grados de libertad

$$gl = N - 1 \quad gl = 10 - 1 \quad gl = 9$$

Una vez obtenidos los valores se puede determinar el valor correspondiente en la **Tabla T** (Ver anexo 1) con un nivel de 0.05 y 9 grados de libertad, el valor para T de dos cola es 2.26 y para T de una cola es 1.83, donde podemos concluir que la Hipótesis se aprueba, debido a que la razón calculada de 6.40 es mayor que 2.26, lo cual demuestra que la diferencia entre los grupos es mayor que el valor que se necesitaba para rechazar la hipótesis de nulidad al nivel de significación de 0.05. Por lo tanto, los datos son lo suficientemente significativos para llegar a la conclusión que la perdida de documentos disminuye. Por lo tanto se procede a aceptar la hipótesis.

Tabla 8 Prueba t con Función de Excel

	<i>Variable 1</i>	<i>Variable 2</i>
Media	5,5	1,4
Varianza	1,83333333	1,15555556
Observaciones	10	10
	-	
Coeficiente de correlación de Pearson	0,38169051	
Diferencia hipotética de las medias	0	
Grados de libertad	9	
Estadístico t	6,40312424	
P(T<=t) una cola	6,2407E-05	
Valor crítico de t (una cola)	1,83311293	
P(T<=t) dos colas	0,00012481	
Valor crítico de t (dos colas)	2,26215716	

6.4 > 1.8 = Hipótesis Alternativa

4.2.2 Control Continuo del Manejo Agrícola

Para obtener los datos del manejo de labores de producción de banano se encuestó a 10 socios (dueños) al azar pertenecientes a APROBAVEN.

Tabla 9 Datos de Encuesta a Socios

N°	Conocimiento (sin aplicación)	Conocimiento (con aplicación)
1	si	si
2	algo	si
3	no	si
4	si	algo
5	no	algo
6	no	no
7	si	no
8	algo	si
9	algo	si
10	si	si

Como se puede observar de los datos tomados al grupo de encuestado es de tipo escrito por lo cual se implementará la tabla de frecuencia para validar las hipótesis especificadas a continuación:

H₀ La aplicación web aumentará la eficiencia administrativa en el manejo de labores de producción de banano.

H₁ La aplicación web aumentará la eficiencia en la producción.

Tabla 10 Tabla de Frecuencia de H0

VALIDOS	FRECUENCIA	PORCENTAJE	PORCENTAJE VALIDO	PORCENTAJE ACUMULADO
si	4	40,00	40	40,00
no	3	30,00	30	60,00
algo	3	30,00	30	
Total	10	100,00	100	100,00

Ilustración 37 Gráfico de frecuencia de H_0

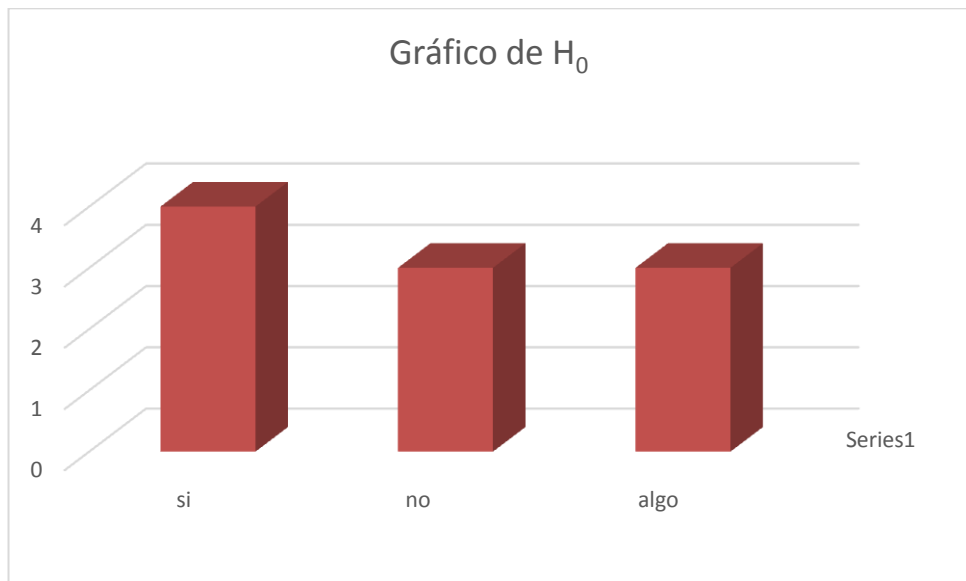
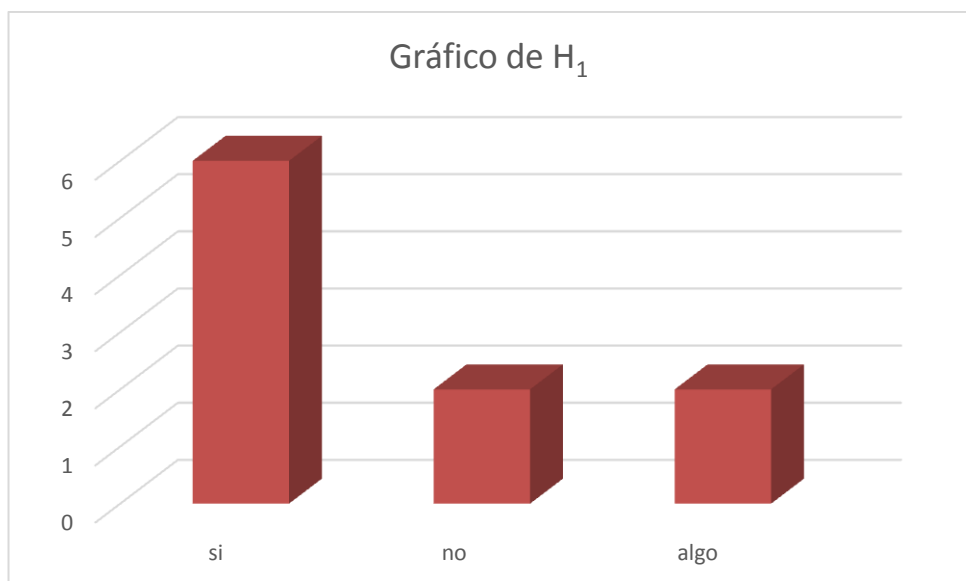


Tabla 11 Tabla de Frecuencia de H_1

VALIDOS	FRECUENCIA	PORCENTAJE	PORCENTAJE VALIDO	PORCENTAJE ACUMULADO
si	6	60,00	60,00	60,00
no	2	20,00	20,00	40,00
algo	2	20,00	20,00	
Total	10	100,00	100,00	100,00

Ilustración 38 Gráfico de Frecuencia de H_1



Por medio de las tablas 10 y 11 podemos comprobar que los socios que contestaron con la respuesta “SI” en $H_1 = 6$, es mayor a los que respondieron en $H_0 = 4$, los que respondieron “Algo” disminuyó: $H_1 = 2$ es menor a $H_0 = 3$; y los encuestados que respondieron que no conocían el control del manejo agrícola disminuyó de $H_1 = 2$ es menor a $H_0 = 5$, con estos datos queda comprobado que los socios han adquirido un mejor conocimiento del control del manejo agrícola de una plantación de banano. Esto nos conduce a afirmar que la utilización de la Aplicación web para el control del manejo agrícola de APROBAVEN influye eficientemente.

Con los datos obtenidos anteriormente se ha verificado que las 2 hipótesis alternativas han sido aceptadas, por lo tanto se acepta que la aplicación web ha mejorado considerablemente la eficiencia en el control de manejo agrícola.

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

- ✚ Las actividades agrícolas que desarrolla APROBAVEN agronómicamente son eficientes pero desconocen que se puede automatizar la guía de actividades a través de un aplicativo.
- ✚ Los socios miembros de la entidad desconocen las existencias de todos los procesos informáticos para una buena gestión dentro de la cadena de producción de banano; en la aplicación web, están incluidos los procesos para la inspección de las labores realizadas dentro de la finca de una forma eficiente, eficaz y sin necesidad de desplazarse a ellas.
- ✚ La aplicación web contiene todas las actividades con sus respectivos parámetros para el manejo agrícola de una plantación de banano; por medio de la planificación proporcionada por la aplicación, se tiene un mayor control y organización de las tareas que se realizan en la plantación. De la creación de usuarios está habilitado únicamente a usuarios administradores de la aplicación web, evitando que tengan acceso indebidos al sistema y para mayor seguridad los solicitantes que pierdan la clave de ingreso, solo podrán recuperarla a través de un usuario con acceso de administrador al aplicativo web.
- ✚ Que la toma de decisiones sea nivel de asociación para que la información generada a través de los reportes de campo tenga plena confiabilidad en los procesos informáticos.
- ✚ El desarrollo de este software mejora eficientemente el control efectivo de las operaciones de la cadena de producción de banano mediante las opciones que posee, lo cual ya no será necesario ir al campo para verificar las labores de las fincas.

5.2. RECOMENDACIONES

- ✚ En primera instancia la aplicación puede ser utilizada solo por la Asociación de Productores Bananeros de Ventanas “APROBAVEN”.
- ✚ Socializar todas las actividades ejecutadas en la entidad ya mencionada y las funcionalidades que brinda el aplicativo web con el fin de mantener informados a los socios y actualizada la aplicación.
- ✚ Establecer los procesos de una manera ordenada y eficiente en una asamblea con todo el personal de APROBAVEN.
- ✚ Designar usuarios administradores para mantener actualizado constantemente el aplicativo y dar soporte en caso de ser necesario a usuarios que lo necesiten; realizar respaldos semanales o diarios para que la información no se pierda en caso de imprevistos y así mantener la confiabilidad de la aplicación.
- ✚ Cuando se creen nuevos reportes, es muy sustancial tener presente los indicadores de la eficacia de labores, que comprueben el cumplimiento en las fincas ante los socios y la entidad.
- ✚ Desarrollar un nuevo módulo para el manejo administrativo y de inventario, mediante la implementación de la arquitectura MVC, y que por ningún motivo consten las reglas de negocio impregnadas en la capa vista mucho menos cadena de conexión a la BD.

CAPÍTULO VI

BIBLIOGRAFÍA

6.1 LITERATURA CITADA

Bozada, A. (2010). *Análisis y Diseño e Implementación de una Guía Operativa de Cultivo para Productos de Exportación*. Sangolquí.

Gonzales, A. (s.f.). *Diseño y Programación Orientada a Objetos* .

Lima, L. F. (2010). *Java 6*. Anaga.

6.2 LINKOGRAFÍA

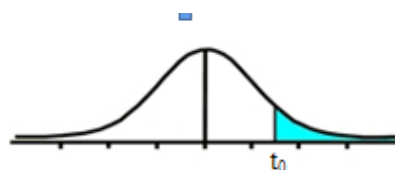
- AEBE, A. d. (2012). *Asociación de Exportadores de Banano del Ecuador*. Obtenido de Asociación de Exportadores de Banano del Ecuador:
http://www.aebe.com.ec/data/file/Noticias2011/AEBE/INDUSTRIA_BANANERA_2010_act_enero_2011.pdf
- Andalucia, J. d. (2013). *Marco de Desarrollo de la Junta de Andalucia*. Obtenido de Marco de Desarrollo de la Junta de Andalucia:
<http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/122>
- Castle Food, T. F. (2012). *Castle Food, Tropical Fruit Export*. Obtenido de Castle Food, Tropical Fruit Export:
<http://www.castlefoods.ec/index.php/es/cavendish-banano>
- Castro, L. (2014). *About.com Internet Básico*. Obtenido de About.com Internet Básico: <http://aprenderinternet.about.com/od/Glosario/g/Que-Es-Java.htm>
- Equipo, R. U. (2012). *Rational Unified Process Equipo*. Obtenido de Rational Unified Process Equipo: <http://rupequipo1.blogspot.com/2012/12/que-es-rup.html>
- Macape. (2009). *Macape*. Obtenido de Macape:
<http://www.ecologico.blogspot.com/2009/03/labores-agricolas.html>
- Palacio, M. (2009). *Calidad del Banano*. Obtenido de Calidad del Banano:
<http://meda-calidadempacadora.blogspot.com/2009/04/labores-de-cosecha-en-cultivo-de-banano.html>
- Rodríguez, E. (2012). *METODOLOGIA RUP: FASES DE DISEÑO Y CONSTRUCCIÓN*. Obtenido de METODOLOGIA RUP: FASES DE DISEÑO Y CONSTRUCCIÓN: <http://metodologirup.blogspot.com/>
- Sebastián, J. (2010). *Comusoft*. Obtenido de Comusoft.:
<http://www.comusoft.com/modelo-vista-controlador-definicion-y-caracteristicas>
- Suarez, H. (2003). *Java Hispano*. Obtenido de Java Hispano:
<http://www.javahispano.org/storage/contenidos/ManualHibernate.pdf>
- Tecnológico, M. (2011). *Mi tecnológico*. Obtenido de Mi tecnológico:
<http://www.mitecnologico.com/Main/ElModeloIncremental>

CAPÍTULO VI

ANEXOS

Anexo 1

Tabla t-Student



Grados de libertad	0.25	0.1	0.05	0.025	0.01	0.005
1	1.0000	3.0777	6.3137	12.7062	31.8210	63.6559
2	0.8165	1.8856	2.9200	4.3027	6.9645	9.9250
3	0.7649	1.6377	2.3534	3.1824	4.5407	5.8408
4	0.7407	1.5332	2.1318	2.7765	3.7469	4.6041
5	0.7267	1.4759	2.0150	2.5706	3.3649	4.0321
6	0.7176	1.4398	1.9432	2.4469	3.1427	3.7074
7	0.7111	1.4149	1.8946	2.3646	2.9979	3.4995
8	0.7064	1.3968	1.8595	2.3060	2.8965	3.3554
9	0.7027	1.3830	1.8331	2.2622	2.8214	3.2498
10	0.6998	1.3722	1.8125	2.2281	2.7638	3.1693
11	0.6974	1.3634	1.7959	2.2010	2.7181	3.1058
12	0.6955	1.3562	1.7823	2.1788	2.6810	3.0545
13	0.6938	1.3502	1.7709	2.1604	2.6503	3.0123
14	0.6924	1.3450	1.7613	2.1448	2.6245	2.9768
15	0.6912	1.3406	1.7531	2.1315	2.6025	2.9467
16	0.6901	1.3368	1.7459	2.1199	2.5835	2.9208
17	0.6892	1.3334	1.7396	2.1098	2.5669	2.8982
18	0.6884	1.3304	1.7341	2.1009	2.5524	2.8784
19	0.6876	1.3277	1.7291	2.0930	2.5395	2.8609
20	0.6870	1.3253	1.7247	2.0860	2.5280	2.8453
21	0.6864	1.3232	1.7207	2.0796	2.5176	2.8314
22	0.6858	1.3212	1.7171	2.0739	2.5083	2.8188
23	0.6853	1.3195	1.7139	2.0687	2.4999	2.8073
24	0.6848	1.3178	1.7109	2.0639	2.4922	2.7970
25	0.6844	1.3163	1.7081	2.0595	2.4851	2.7874
26	0.6840	1.3150	1.7056	2.0555	2.4786	2.7787
27	0.6837	1.3137	1.7033	2.0518	2.4727	2.7707
28	0.6834	1.3125	1.7011	2.0484	2.4671	2.7633
29	0.6830	1.3114	1.6991	2.0452	2.4620	2.7564
30	0.6828	1.3104	1.6973	2.0423	2.4573	2.7500
31	0.6825	1.3095	1.6955	2.0395	2.4528	2.7440
32	0.6822	1.3086	1.6939	2.0369	2.4487	2.7385
33	0.6820	1.3077	1.6924	2.0345	2.4448	2.7333
34	0.6818	1.3070	1.6909	2.0322	2.4411	2.7284
35	0.6816	1.3062	1.6896	2.0301	2.4377	2.7238
36	0.6814	1.3055	1.6883	2.0281	2.4345	2.7195
37	0.6812	1.3049	1.6871	2.0262	2.4314	2.7154
38	0.6810	1.3042	1.6860	2.0244	2.4286	2.7116
39	0.6808	1.3036	1.6849	2.0227	2.4258	2.7079
40	0.6807	1.3031	1.6839	2.0211	2.4233	2.7045
41	0.6805	1.3025	1.6829	2.0195	2.4208	2.7012
42	0.6804	1.3020	1.6820	2.0181	2.4185	2.6981
43	0.6802	1.3016	1.6811	2.0167	2.4163	2.6951
44	0.6801	1.3011	1.6802	2.0154	2.4141	2.6923
45	0.6800	1.3007	1.6794	2.0141	2.4121	2.6896
46	0.6799	1.3002	1.6787	2.0129	2.4102	2.6870
47	0.6797	1.2998	1.6779	2.0117	2.4083	2.6846
48	0.6796	1.2994	1.6772	2.0106	2.4066	2.6822
49	0.6795	1.2991	1.6766	2.0096	2.4049	2.6800

50	0.6794	1.2987	1.6759	2.0086	2.4033	2.6778
51	0.6793	1.2984	1.6753	2.0076	2.4017	2.6757
52	0.6792	1.2980	1.6747	2.0066	2.4002	2.6737
53	0.6791	1.2977	1.6741	2.0057	2.3988	2.6718
54	0.6791	1.2974	1.6736	2.0049	2.3974	2.6700
55	0.6790	1.2971	1.6730	2.0040	2.3961	2.6682
56	0.6789	1.2969	1.6725	2.0032	2.3948	2.6665
57	0.6788	1.2966	1.6720	2.0025	2.3936	2.6649
58	0.6787	1.2963	1.6716	2.0017	2.3924	2.6633
59	0.6787	1.2961	1.6711	2.0010	2.3912	2.6618
60	0.6786	1.2958	1.6706	2.0003	2.3901	2.6603
61	0.6785	1.2956	1.6702	1.9996	2.3890	2.6589
62	0.6785	1.2954	1.6698	1.9990	2.3880	2.6575
63	0.6784	1.2951	1.6694	1.9983	2.3870	2.6561
64	0.6783	1.2949	1.6690	1.9977	2.3860	2.6549
65	0.6783	1.2947	1.6686	1.9971	2.3851	2.6536
66	0.6782	1.2945	1.6683	1.9966	2.3842	2.6524
67	0.6782	1.2943	1.6679	1.9960	2.3833	2.6512
68	0.6781	1.2941	1.6676	1.9955	2.3824	2.6501
69	0.6781	1.2939	1.6672	1.9949	2.3816	2.6490
70	0.6780	1.2938	1.6669	1.9944	2.3808	2.6479
71	0.6780	1.2936	1.6666	1.9939	2.3800	2.6469
72	0.6779	1.2934	1.6663	1.9935	2.3793	2.6458
73	0.6779	1.2933	1.6660	1.9930	2.3785	2.6449
74	0.6778	1.2931	1.6657	1.9925	2.3778	2.6439
75	0.6778	1.2929	1.6654	1.9921	2.3771	2.6430
76	0.6777	1.2928	1.6652	1.9917	2.3764	2.6421
77	0.6777	1.2926	1.6649	1.9913	2.3758	2.6412
78	0.6776	1.2925	1.6646	1.9908	2.3751	2.6403
79	0.6776	1.2924	1.6644	1.9905	2.3745	2.6395
80	0.6776	1.2922	1.6641	1.9901	2.3739	2.6387
81	0.6775	1.2921	1.6639	1.9897	2.3733	2.6379
82	0.6775	1.2920	1.6636	1.9893	2.3727	2.6371
83	0.6775	1.2918	1.6634	1.9890	2.3721	2.6364
84	0.6774	1.2917	1.6632	1.9886	2.3716	2.6356
85	0.6774	1.2916	1.6630	1.9883	2.3710	2.6349
86	0.6774	1.2915	1.6628	1.9879	2.3705	2.6342
87	0.6773	1.2914	1.6626	1.9876	2.3700	2.6335
88	0.6773	1.2912	1.6624	1.9873	2.3695	2.6329
89	0.6773	1.2911	1.6622	1.9870	2.3690	2.6322
90	0.6772	1.2910	1.6620	1.9867	2.3685	2.6316
91	0.6772	1.2909	1.6618	1.9864	2.3680	2.6309
92	0.6772	1.2908	1.6616	1.9861	2.3676	2.6303
93	0.6771	1.2907	1.6614	1.9858	2.3671	2.6297
94	0.6771	1.2906	1.6612	1.9855	2.3667	2.6291
95	0.6771	1.2905	1.6611	1.9852	2.3662	2.6286
96	0.6771	1.2904	1.6609	1.9850	2.3658	2.6280
97	0.6770	1.2903	1.6607	1.9847	2.3654	2.6275
98	0.6770	1.2903	1.6606	1.9845	2.3650	2.6269
99	0.6770	1.2902	1.6604	1.9842	2.3646	2.6264
100	0.6770	1.2901	1.6602	1.9840	2.3642	2.6259
∞	0.6745	1.2816	1.6449	1.9600	2.3263	2.5758

Anexo 2

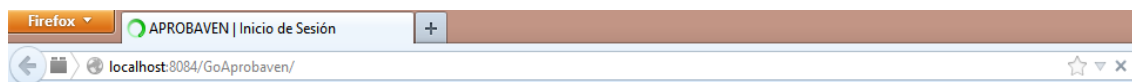
Manual de Usuario

El objetivo de la elaboración de este manual es dar a conocer a los socios de APROBAVEN el fácil acceso a la información y permitir que los usuarios se habitúen a las características y formas del funcionamiento durante la navegación de la aplicación web.

Requerimientos Básicos.

- 🌐 Navegador Chrome desde la versión 10 en adelante.
- 🌐 Navegador Firefox desde la versión 10 en adelante.
- 🌐 Adobe Flash Player desde la versión 10 en adelante.
- 🌐 Java Runtime desde la versión 10 en adelante.

1 Abrir el navegador y digitar la siguiente dirección:



2 A continuación se mostrará la página de inicio de sesión de la aplicación web.



3 Se muestra la página principal, los usuarios pueden acceder a todas las opciones disponibles.



4. Al hacer clic en la pestaña planificación en el tab; seleccionamos un Lote, Año, Semana, Labor y damos clic en el botón guardar. Esta opción nos sirve para planificar las labores durante las 52 semanas del año.

5. Al dar clic en la pestaña Estimación en el tab, ingresamos los datos que nos solicita y damos clic en botón guardar. Nos sirve para llevar un control de una posible producción y poder comercializarla.



6. Al dar clic en la pestaña Producción en el tab, ingresamos los datos que nos solicita y damos clic en el botón guardar. Nos sirve para llevar un registro de la producción de nuestra hacienda.

The screenshot shows the APROBAVEN 'Guía Operativa' interface. At the top, there is a blue header with a banana logo and the text 'APROBAVEN Guía Operativa'. Below the header, a navigation bar contains links: 'Principal', 'Fincas', 'Lotes', 'Control', 'Administración', 'Reportes', 'Quiénes Somos', and 'Cerrar sesión'. The 'Producción' tab is selected, and the 'Planificación' and 'Estimación' tabs are also visible. The main form area contains the following fields: 'Hacienda:' (text box with 'LA MIRIAN'), 'Semana:' (dropdown), 'Año:' (dropdown), 'Marca:' (text box with 'Chiquita, Bonanza...etc'), 'Evaluador:' (text box with 'Nombre Apellido'), 'Día:' (text box), 'Cajas:' (text box), 'Racimos cortados:' (text box), 'Racimos procesados:' (text box), 'Racimos rechazados:' (text box), 'Grado máximo:' (text box), 'Grado mínimo:' (text box), 'Precio:' (text box with '0.00'), and 'Responsable:' (dropdown). A 'Guardar' button is located at the bottom right of the form. The footer of the page reads 'APROBAVEN | Todos los derechos reservados © 2013'.

7. Al hacer clic en la opción Fincas del menú principal se nos abre la página de registrar hacienda e ingresamos todos los datos que nos piden y damos clic en botón guardar para registrar la hacienda en el sistema.

The screenshot shows the APROBAVEN 'Guía Operativa' interface. At the top, there is a blue header with a banana logo and the text 'APROBAVEN Guía Operativa'. Below the header, a navigation bar contains links: 'Principal', 'Fincas', 'Lotes', 'Control', 'Administración', 'Reportes', 'Quiénes Somos', and 'Cerrar sesión'. The 'Fincas' tab is selected. The main form area contains the following fields: 'Id_Hacienda:' (text box), 'Nombre hacienda:' (text box with 'Nombre de Hacienda'), 'Dirección:' (text box), 'Hectáreas:' (text box), and 'Socio:' (dropdown). 'Guardar' and 'Cancelar' buttons are located at the bottom of the form. The footer of the page reads 'APROBAVEN | Todos los derechos reservados © 2013'.

8. Al hacer clic en la opción Lotes del menú principal se nos abre la página de Asignar los lotes a la hacienda que hemos iniciado sesión e ingresamos todos los datos que nos solicita y damos clic en el botón guardar, no se podrá registrar hectáreas de los lotes mayores a las hectáreas registrada de la hacienda.



Nombre de Lote:

Hectáreas c/u Lotes:

Hacienda:

Id lote	Nombre	Hectareas
1	LOTE1	25
2	LOTE2	25

APROBAVEN | Todos los derechos reservados © 2013

9. Al hacer clic en la opción Control del menú principal se nos abre la página de controlar la planificación. Escogemos el lote, año, semana y damos clic en el botón mostrar se nos mostrará lo planificado para ese lote en dicha semana.



Lote: Año:

Semana:

1 - LOTE1 1-Semana1 2-2014

DESHIJE
OPA
FERTILIZACI
ENFUNDE
ENFUNDE

APROBAVEN | Todos los derechos reservados © 2013

10. Para ingresar datos a la labor planificada seleccionamos una labor y pulsamos el botón control y se nos abre una ventana e ingresamos los datos de dicha labor y damos clic en el botón guardar.



APROBAVEN
Guía Operativa

Bienvenido(a): ADMINISTRADOR Hacienda LA MIRIAN

Principal Fincas Lotes **Control** Administración Reportes Quienes Somos Cerrar sesion

Plan: 2
Responsable:
Fecha: Calendario
CANTIDAD PLANTAS(Unidades)
Guardar

APROBAVEN | Todos los derechos reservados © 2013

11. Al hacer clic en la opción Administración del menú principal se nos abre la página de Administración esta opción está disponible solo para usuarios administradores .Nos aparece varias opciones en un tab.



APROBAVEN
Guía Operativa

Bienvenido(a): ADMINISTRADOR Hacienda LA MIRIAN

Principal Fincas Lotes Control **Administración** Reportes Quienes Somos Cerrar sesion

Registrarme **Socio** Personal Labores Característica

Id:
Nick:
Clave:
Nombre:
Apellido:
Tipo:
Guardar Cancelar Modificar

APROBAVEN | Todos los derechos reservados © 2013

12. Al hacer clic en la pestaña registrarme en el tab se nos abre la página de registrar usuarios e ingresamos los datos solicitados y damos clic en el botón guardar para almacenar los datos ingresados.

The screenshot shows the APROBAVEN 'Guía Operativa' web application. At the top, there is a blue header with a cartoon banana logo and the text 'APROBAVEN Guía Operativa'. Below the header, a blue bar displays the user greeting: 'Bienvenido(a): ADMINISTRADOR Hacienda LA MIRIAN'. A green navigation bar contains the following links: 'Principal', 'Fincas', 'Lotes', 'Control', 'Administración' (highlighted), 'Reportes', 'Quienes Somos', and 'Cerrar sesion'. The main content area features a tabbed interface with five tabs: 'Registrarme' (active), 'Socio', 'Personal', 'Labores', and 'Característica'. The 'Registrarme' form includes the following fields: 'Id:', 'Nick:', 'Clave:', 'Nombre:', 'Apellido:', and 'Tipo:' (a dropdown menu). At the bottom of the form are three buttons: 'Guardar', 'Cancelar', and 'Modificar'. The footer is a blue bar with the text 'APROBAVEN | Todos los derechos reservados © 2013'.

13. Al hacer clic en la pestaña socio en el tab se nos abre la página de registrar socios e ingresamos los datos solicitados y damos clic en el botón guardar para almacenar los datos ingresados.

The screenshot shows the APROBAVEN 'Guía Operativa' web application. At the top, there is a blue header with a cartoon banana logo and the text 'APROBAVEN Guía Operativa'. Below the header, a blue bar displays the user greeting: 'Bienvenido(a): ADMINISTRADOR Hacienda LA MIRIAN'. A green navigation bar contains the following links: 'Principal', 'Fincas', 'Lotes', 'Control', 'Administración' (highlighted), 'Reportes', 'Quienes Somos', and 'Cerrar sesion'. The main content area features a tabbed interface with five tabs: 'Registrarme', 'Socio' (active), 'Personal', 'Labores', and 'Característica'. The 'Socio' form includes the following fields: 'Código:', 'Nombres:', 'Apellidos', 'Dirección:', 'Telefono:', 'Celular:', and 'Correo:'. At the bottom of the form are three buttons: 'Guardar', 'Cancelar', and 'Modificar'. The footer is a blue bar with the text 'APROBAVEN | Todos los derechos reservados © 2013'.

14. Al hacer clic en la pestaña personal en el tab se nos abre la página de registrar personal e ingresamos los datos solicitados y damos clic en el botón guardar para almacenar los datos ingresados.

The screenshot shows the APROBAVEN 'Guía Operativa' interface. At the top, there is a blue header with a cartoon banana logo and the text 'APROBAVEN Guía Operativa'. Below this is a navigation bar with the text 'Bienvendio(a): ADMINISTRADOR Hacienda LA MIRIAN' and a series of tabs: 'Principal', 'Fincas', 'Lotes', 'Control', 'Administración', 'Reportes', 'Quienes Somos', and 'Cerrar sesion'. The 'Personal' tab is selected. Below the navigation bar, there is a form with five tabs: 'Registrarme', 'Socio', 'Personal', 'Labores', and 'Característica'. The 'Personal' tab is active, displaying a registration form with the following fields: 'Id_Personal:', 'Nombres:', 'Apellidos:', 'Sexo:' (with a dropdown arrow), and 'Instrucción:'. At the bottom of the form are two buttons: 'Guardar' and 'Cancelar'. The footer of the page reads 'APROBAVEN | Todos los derechos reservados © 2013'.

15. Al hacer clic en la pestaña labores en el tab se nos abre la página de registrar labores e ingresamos los datos solicitados y damos clic en el botón guardar para almacenar los datos ingresados.

The screenshot shows the APROBAVEN 'Guía Operativa' interface, similar to the previous one. The 'Labores' tab is now selected in the main navigation bar. In the sub-navigation bar, the 'Labores' tab is active, displaying a registration form with a single field: 'Nombres:'. At the bottom of the form are two buttons: 'Guardar' and 'Cancelar'. The footer of the page remains 'APROBAVEN | Todos los derechos reservados © 2013'.

16. Al hacer clic en la pestaña Característica en el tab se nos abre la página de registrar característica de labores, seleccionamos una labor e ingresamos los datos que nos solicitan y damos clic en el botón guardar para almacenar los datos ingresados.

The screenshot shows the APROBAVEN 'Guía Operativa' interface. At the top, there is a blue header with a banana logo and the text 'APROBAVEN Guía Operativa'. Below the header, a blue bar displays the user's name: 'Bienvendio(a): ADMINISTRADOR Hacienda LA MIRIAN'. A green navigation bar contains the following links: 'Principal', 'Fincas', 'Lotes', 'Control', 'Administración', 'Reportes', 'Quienes Somos', and 'Cerrar sesion'. The main content area features a tabbed interface with five tabs: 'Regístrame', 'Socio', 'Personal', 'Labores', and 'Característica'. The 'Característica' tab is active, showing a form with the following fields: 'Labor:' with a dropdown menu set to 'ENFUNDE', 'Nombre:' with an empty text box, and 'Unidad de Medida:' with a dropdown menu set to 'Centimetro/Unidades/'. Below these fields are two buttons: 'Guardar' and 'Cancelar'. At the bottom of the page, a blue footer contains the text 'APROBAVEN | Todos los derechos reservados © 2013'.

17. Al hacer clic en la opción Reportes del menú principal se nos abre la página de Reportes .Nos aparece varios opciones en un tab con todos los reportes que emite la aplicación.

The screenshot shows the APROBAVEN 'Guía Operativa' interface. At the top, there is a blue header with a banana logo and the text 'APROBAVEN Guía Operativa'. Below the header, a blue bar displays the user's name: 'Bienvendio(a): ADMINISTRADOR Hacienda LA MIRIAN'. A green navigation bar contains the following links: 'Principal', 'Fincas', 'Lotes', 'Control', 'Administración', 'Reportes', 'Quienes Somos', and 'Cerrar sesion'. The main content area features a tabbed interface with two rows of tabs. The first row contains: 'Generales', 'Socios & Haciendas', 'Planificación', 'Control', and 'Enfunde'. The second row contains: 'Control_Labor', 'Estimación', and 'Producción'. The 'Reportes' tab is active, showing a form with three buttons: 'Haciendas', 'Socios', and 'Historial'. At the bottom of the page, a blue footer contains the text 'APROBAVEN | Todos los derechos reservados © 2013'.

18. Al hacer clic en la opción Quienes somos del menú principal se nos abre la página con información de APROBAVEN.



19. Para recuperar la contraseña, se debe iniciar sesión con un usuario administrador, escogemos la opción Registrarme y damos clic en el botón modificar, seleccionamos el usuario, damos clic en el botón buscar e ingresamos una nueva contraseña y por seguridad volvemos a escoger el tipo de usuario y damos clic en guardar.



APROBAVEN
Guía Operativa

3

4

Usuario:
ADMINISTRADOR ▾
Buscar

Id: 0
Nick: ADMIN
Nombre: ADMINISTRADOR
Apellido: SISTEMA
Contraseña:
Tipo Usuario: ▾

5

6

7

Guardar

22. Para modificar los datos de socios, se debe iniciar sesión con un usuario administrador, escogemos la opción Socio y damos clic en el botón modificar, seleccionamos el socio, damos clic en el botón buscar y podemos modificar la dirección, teléfono, celular, e-mail y damos clic en guardar.

APROBAVEN
Guía Operativa

Bienvenido(a): ADMINISTRADOR Hacienda LA MIRIAN

1

Principal Fincas Lotes Control **Administración** Reportes Quienes Somos Cerrar sesion

2

Registrar Socio Personal Labores Característica

Código:
Nombres:
Apellidos:
Dirección:
Telefono:
Celular:
Correo:
Guardar Cancelar **Modificar**

3

APROBAVEN | Todos los derechos reservados © 2013

The screenshot shows the APROBAVEN application interface. At the top is a blue header with a cartoon banana logo on the left and the text "APROBAVEN" in large green letters and "Guía Operativa" in white below it. The main content area is white and contains a form with the following fields and buttons:

- Socio:** A dropdown menu showing "Admin Admin" with a downward arrow. A line labeled "4" points to this dropdown.
- Buscar:** A button below the "Socio" dropdown. A line labeled "5" points to this button.
- Id:** A text input field containing the number "1".
- Nombre:** A text input field containing the text "Admin".
- Apellido:** A text input field containing the text "Admin".
- Dirección:** A text input field containing the text "123456789". A line labeled "6" points to this field.
- Telefono:** A text input field containing the text "0000". A line labeled "7" points to this field.
- Celular:** A text input field containing the text "0000". A line labeled "8" points to this field.
- E-mail:** A text input field containing the text "aprobaven@gmail.com". A line labeled "9" points to this field.
- Guardar:** A button below the "E-mail" field. A line labeled "10" points to this button.

21. Al dar clic en cerrar sesión se nos cierra la aplicación.

UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO
FACULTAD DE CIENCIAS DE LA INGENIERÍA
F O R M A T O

DATOS PERSONALES								
CÉDULA	NOMBRES	APELLIDOS	PROVINCIA	CANTÓN	PARROQUIA	DIRECCIÓN	TELÉFONO	CORREO ELECTRONICO
120547686-2	César Fabián	Varas Beltrán	Los Ríos	Ventanas	Ventanas	Cdla. Las Palmeras Mz: J Villa 5	2970 - 540	cesarvarasbeltran@hotmail.com

DATOS DEL EGRESADO			
TÍTULO A OBTENER	CARRERA	PARALELO	FECHA DE EGRESO
Ingeniero en Sistema	Ingeniería en Sistema	"A"	Marzo 26 del 2013

DATOS DE TESIS						
TEMA DE TESIS	RES. APROB. DIRECCIÓN DE TESIS	RES. FECHA DE SUSTENTACIÓN	FECHA DE SUSTENTACIÓN	CÉDULA DEL TUTOR	NOMBRE DEL TUTOR	FECHA DE INCORPORACIÓN
Construcción de una guía operativa basada en software para la Gestión Control de la Producción y Comercialización de la Asociación de Productores Bananeros de Ventanas APROBAVEN	26-HCD-FCI-2013	012-HCD-FCI-2014	Marzo 21 del 2014	030111490-6	Ing. Ariosto Vicuña Msc.	Marzo 21 del 2014