

	GESTIÓN DE SERVICIOS ACADÉMICOS Y BIBLIOTECARIOS		CÓDIGO	FO-GS-15	
			VERSIÓN	02	
	<b>ESQUEMA HOJA DE RESUMEN</b>			FECHA	03/04/2017
				PÁGINA	1 de 1
ELABORÓ		REVISÓ		APROBÓ	
Jefe División de Biblioteca		Equipo Operativo de Calidad		Líder de Calidad	

## RESUMEN TRABAJO DE GRADO

AUTOR(ES) NOMBRES Y APELLIDOS

NOMBRES(S): EDINSON YESID APELLIDOS: GOMEZ SERRANO

NOMBRES(S): GEIMY MARCELA APELLIDOS: NAVARRO MONSALVE

FACULTAD: INGENIERÍA

PLAN DE ESTUDIO: INGENIERÍA DE SISTEMAS

DIRECTOR(ES) PROFESOR - ASESOR DEL TRABAJO ACADÉMICO

NOMBRES(S): ING. JUDITH DEL PILAR APELLIDOS: RODRÍGUEZ TENJO

NOMBRES(S): ING. OSCAR ALBERTO APELLIDOS: GALLARDO PÉREZ

TÍTULO DEL TRABAJO (TESIS): SISTEMA DE INFORMACIÓN PARA LA GESTIÓN DE LOS PROCESOS ADMINISTRATIVOS DEL SECTOR AVÍCOLA EN LA FINCA SAN PABLO DE LA UFPS

En el presente trabajo se aborda la apropiación de tecnologías para lograr un mejor aprovechamiento de las mismas, para este caso se enfocó en el sector avícola, se evidenció una problemática en la administración y almacenamiento de los procesos en la finca San Pablo de la Universidad Francisco de Paula Santander, partiendo de esta problemática se inicia un proceso de generación de ideas para lograr mitigar, durante este análisis se realizó una visita a la finca para poder evidenciar cómo se llevaban los procesos y así poder identificar acerca del tratamiento de los datos, los procesos de producción de pollos, los huevos de gallinas y codornices no es el más eficiente, dado que se realizan los seguimientos en bitácoras hechas en papel, además estos registros al crecer exponencialmente con cada ciclo de producción tienden a generar muchos datos que no son viables llevarlos en una bitácora tradicional, la solución planteada para dicho problema, es desarrollar un producto de software que cuente con la capacidad de administrar los registros de producción de huevos, los pollos de engorde, asimismo generando una proyección estadística de estos datos que se suministran en cada ciclo de producción.

PALABRAS CLAVE: AVICOLA, SISTEMA DE INFORMACIÓN, TRATAMIENTO DE DATOS, BITÁCORA

PÁGINAS: 41 PLANOS: \_\_\_\_\_ ILUSTRACIONES: \_\_\_\_\_ CD ROOM: \_\_\_\_\_

SISTEMA DE INFORMACIÓN PARA LA GESTIÓN DE LOS PROCESOS  
ADMINISTRATIVOS DEL SECTOR AVÍCOLA EN LA FINCA SAN PABLO DE LA UFPS

GÓMEZ SERRANO EDINSON YESID  
NAVARRO MONSALVE GEIMY MARCELA

UNIVERSIDAD FRANCISCO DE PAULA SANTANDER  
FACULTAD DE INGENIERÍA  
PLAN DE ESTUDIOS DE INGENIERÍA DE SISTEMAS  
CÚCUTA  
2022

SISTEMA DE INFORMACIÓN PARA LA GESTIÓN DE LOS PROCESOS  
ADMINISTRATIVOS DEL SECTOR AVÍCOLA EN LA FINCA SAN PABLO DE LA UFPS

GÓMEZ SERRANO EDINSON YESID  
NAVARRO MONSALVE GEIMY MARCELA

TRABAJO DE GRADO  
PARA OPTAR AL TÍTULO DE INGENIERO DE SISTEMAS

DIRECTORA  
ING. JUDITH DEL PILAR RODRIGUEZ TENJO

CODIRECTOR  
ING. OSCAR ALBERTO GALLARDO PÉREZ

UNIVERSIDAD FRANCISCO DE PAULA SANTANDER  
FACULTAD DE INGENIERÍA  
PLAN DE ESTUDIOS DE INGENIERÍA DE SISTEMAS  
CÚCUTA

2022

## ACTA DE SUSTENTACIÓN DE TRABAJO DE GRADO

**FECHA:** 13 DE JULIO DE 2022

**HORA:** 2:30 P. M.

**LUGAR:** AUDITORIO JJ MALDONADO 4 PISO AULA SUR - UFPS

**PLAN DE ESTUDIOS:** INGENIERÍA DE SISTEMAS

**TÍTULO DEL TRABAJO DE GRADO:** "SISTEMA DE INFORMACIÓN PARA LA GESTIÓN DE LOS PROCESOS ADMINISTRATIVOS DEL SECTOR AVÍCOLA EN LA FINCA SAN PABLO DE LA UFPS"

### Jurados

ING. CARMEN JANETH PARADA

ING. JAIRO ALBERTO FUENTES CAMARGO

Dr. JUAN FRANCISCO BAUTISTA RODRÍGUEZ

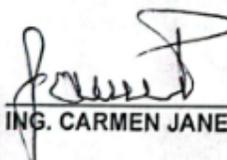
**DIRECTOR:** JUDITH DEL PILAR RODRÍGUEZ TENJO

**CODIRECTOR:** OSCAR ALBERTO GALLARDO PÉREZ

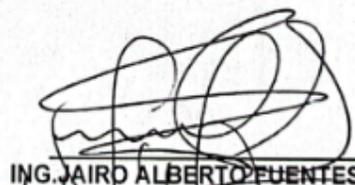
NOMBRE DEL ESTUDIANTE	CÓDIGO	CALIFICACIÓN	
		NÚMERO	LETRA
GEIMY MARCELA NAVARRO MONSALVE	1151276	3,7	TRES, SIETE
ÉDISON YESID GÓMEZ SERRANO	1151301	3,7	TRES, SIETE

## APROBADO

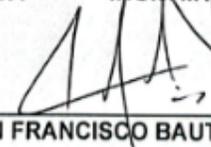
FIRMA DE LOS JURADOS



ING. CARMEN JANETH PARADA



ING. JAIRO ALBERTO FUENTES CAMARGO



Dr. JUAN FRANCISCO BAUTISTA RODRÍGUEZ



JUDITH DEL PILAR RODRIGUEZ TENJO  
Coordinadora - Comité Curricular

## Tabla de contenido

Introducción	10
1. Descripción del problema	11
1.1 Planteamiento del problema	11
1.2 Justificación	11
1.3 Objetivos	12
1.3.1. Objetivo General	12
1.3.2. Objetivo Específicos	12
1.4 Alcance	13
1.5 Metodología de desarrollo	14
1.5.1 Fase 1: Identificación y exploración	15
Figura 1. Tomado de la finca San Pablo de la UFPS - criadero de pollos de engorde	15
Figura 2. - Galpón de pollos de engorde   Figura 3: Tomado de la finca San Pablo de la UFPS-Galpón de gallinas ponedoras	16
Figura 4: Tomado de la finca San Pablo de la UFPS - producción de carne a partir de pollos de engorde.	16
1.5.2 Fase 2: Producción del sistema de información	17
1.5.3 Fase 3: Entrega	17
2. Marco metodológico	17
2.1 Gestión y comprensión del negocio	17
Figura 5. - Proceso del ciclo de vida	19
Tomado del libro de Pressman “Ingeniería del Software, Un Enfoque práctico” 7° edición	19
2.2 Administración de proyecto de software	20

	6
3. Requerimientos	21
3.1 Requerimientos funcionales	21
3.2 Requerimientos no funcionales	22
3.3 Especificación de requerimientos	22
4. Modelo de análisis	29
4.1 Casos de uso generales	29
4.1.1 Diagrama Caso De Uso Crear Usuario	29
4.1.2 Diagrama De Caso De Uso Iniciar Sesión	29
4.1.3 Diagrama De Caso De Uso Registrar Visitante	30
4.1.4 Diagrama De Caso De Uso Registrar Vehículos	30
4.1.5 Diagrama De Caso De Uso Registrar Galpones	31
4.1.6 Diagrama De Caso De Uso Registro De Ponedoras Por Semana	31
4.1.7 Diagrama De Caso De Uso Registro De Codorniz Por Semana	32
5. Arquitectura del sistema	32
Figura 6. Patrón MVC	32
Figura 7. Arquitectura de Angular	33
Imagen tomada de: <a href="https://www.simplilearn.com/ice9/free_resources_article_thumb/Angular_Architecture-What_is_Angular.PNG">https://www.simplilearn.com/ice9/free_resources_article_thumb/Angular_Architecture-What_is_Angular.PNG</a>	33
5.1 Autenticación	33
Figura 8- Diagrama de autenticación de Google	33
Imagen tomada de: <a href="https://cloud.google.com/architecture/authenticating-users-to-firestore-with-identity-platform-and-google-identities?hl=es-419">https://cloud.google.com/architecture/authenticating-users-to-firestore-with-identity-platform-and-google-identities?hl=es-419</a>	33
5.2 Patrones de diseño	34

	7
Figura 9- Patrones de diseño usados en la aplicación	34
Imagen tomada de: <a href="https://digibuo.uniovi.es/dspace/bitstream/handle/10651/39133/TFM_PabloSotoMedina.pdf?sequence=9">https://digibuo.uniovi.es/dspace/bitstream/handle/10651/39133/TFM_PabloSotoMedina.pdf?sequence=9</a>	34
6. Diagrama de despliegue	34
Figura 10- Vista de despliegue	35
6.1 Modelo de datos de Cloud Firestore	35
Figura 11. Visualización de colección de datos	36
7. Pruebas del software	36
7.1 Contexto de las Pruebas	36
7.1.1 Elementos de Prueba	36
7.1.2 Diseño de casos de prueba.	37
7.1.3 Técnicas de prueba.	37
7.1.3.4 Pruebas de caja blanca	37
Figura 12. Prueba 1- Validación de componente principal	38
Figura 13. Prueba 2- Autenticación	38
7.1.3.5 Pruebas de caja negra	39
Figura 14. prueba caja negra 1 - Validación de campos numéricos en formulario	39
7.2 Herramientas tecnológicas utilizadas en las pruebas del proyecto	39
7.2.1 Descripción	39
7.2.2 Jira	40
7.2.2.1 Resultados obtenidos	40
Figura 15. Diseño de casos de prueba con Jira	40
7.2.3 Karma (Jasmine)	41
7.2.3.1 Resultados obtenidos	41

	8
Figura 16. resultado de pruebas unitarias	41
Figura 17. porcentaje de código evaluado en cada componente	42
8. Referencias	43

## Lista de Figuras

Figura 1. Criadero de pollos de engorde	14
Figura 2. Galpón de pollos de engorde	15
Figura 3. Galpón de gallinas ponedoras	15
Figura 4. Producción de carne a partir de pollos de engorde	15
Figura 5. Proceso del ciclo de vida	18
Figura 6. Patrón MVC	32
Figura 7. Arquitectura de Angular	32
Figura 8. Diagrama de autenticación de Google	33
Figura 9. Patrones de diseño usados en la aplicación	33
Figura 10. Vista de despliegue	34
Figura 11. Visualización de colección de datos	36
Figura 12. Prueba 1- Validación de componente principal	38
Figura 13. Prueba 2- Autenticación	38
Figura 14. Prueba caja negra 1 - Validación de campos numéricos en formulario	39
Figura 15. Diseño de casos de prueba con Jira	40
Figura 16. Resultado de pruebas unitarias	41
Figura 17. Porcentaje de código evaluado en cada componente	41

## Introducción

En los últimos años se ha evidenciado un gigante cambio tecnológico y es conveniente tener en cuenta el valor que éstas mismas aportan a cualquier tipo de producto de software en desarrollo, en las grandes, pequeñas y medianas empresas colombianas diariamente realizan infinitos procesos los cuales en su gran mayoría se convierten en procesos repetitivos y mecánicos, en el artículo de Gamez Tellez, A. M., Morales Bautista, M. C., & Ramirez Lopez, C. T. (2018) menciona que *“De igual manera, las tecnologías de información y comunicación son importantes porque las empresas que las implementen tienen mayor posibilidad de acceder a incentivos gubernamentales, reducir costos y manejar de manera adecuada sus recursos, ocasionando que las empresas sean más competitivas al reducir barreras”*, esto quiere decir que la falta de inversión en los componentes tecnológicos en las empresas produce impactos negativos en aspectos financieros, debido a que se llevaran los procesos de manera más lenta y quizás menos efectiva, por ello antes, durante y después el desarrollo de este proyecto se plantea la siguiente pregunta hipotética, ¿Los procesos administrativos que la finca realiza de forma manual deben tener una transformación tecnológica? y como solución a esta pregunta se ha desarrollado un sistema en el cual se llevará a cabo un parte de la administración de la finca, siendo estos lo que tengan que ver con el área avícola.

## **1. Descripción del problema**

### **1.1 Planteamiento del problema**

La Universidad Francisco de Paula Santander seccional Cúcuta, cuenta con la finca San Pablo cuya propiedad es un centro de prácticas y de investigación para todos los estudiantes de la Facultad de Ciencias Agrarias y de ambiente de la UFPS, además cabe resaltar que la finca San Pablo está abierta para los estudiantes de otras universidades de la región y del país como un lugar donde puedan realizar sus prácticas en las diferentes actividades en el ámbito de pecuario, agronómico y avícola.

Los procesos administrativos que posee la finca San Pablo se llevan a cabo de forma anticuada, debido a que realizan los controles de los galpones, ingreso de los pollos, el registro de la mortalidad y natalidad, además del registro de las personas que visitan la finca por medio de bitácoras, las cuales se componen de hojas de papel en las que se plasma la información anteriormente mencionada, lo que provoca una forma poco efectiva de gestionar toda esta información, ya que es una gran cantidad de información que se debe de procesar.

### **1.2 Justificación**

La estrategia de tratamiento de datos en la finca consiste en ingresar de forma manual toda la información en relación a los pollos de engorde, gallinas ponedoras y codornices, es decir que los procesos como el registro de ingreso a la finca, en bitácoras y/u hojas de cálculo, lo cual genera demoras en los procesos de registro tanto de ingreso a la finca y el seguimiento de un determinado galpón, lo cual implica que arroje como resultado ineficiencia y obsolescencia respecto a las estrategias que se usan en la actualidad.

Para la producción de los pollos de engorde, gallinas ponedoras y codornices de la finca se requieren de dietas alimenticias, medicamentos controlados para mitigar cualquier enfermedad que se pueda reproducir, es decir tener un control de la sanidad para los animales. Según el ICA, dentro de las etapas de la cadena aviar existen ciertos peligros que pueden ser de tres tipos: biológicos, químicos y físicos. Dentro del programa de Bioseguridad los más significativos son los agentes infecciosos, que producen una serie de riesgos a los que está expuesta la materia prima como son las aves, el propio trabajador, el medio que lo rodea y la misma comunidad. En base a lo anterior el tratamiento y gestión de los procesos que comprenden la producción avícola requieren una muy eficiente forma de conducción, lo que convierte a un sistema de información en la solución más viable para afrontar este problema.

### **1.3 Objetivos**

#### **1.3.1. Objetivo General**

Desarrollar un sistema de información web para el apoyo de los procesos administrativos en la producción de pollos de engorde e ingreso a la finca San Pablo de la UFPS- Seccional Cúcuta.

#### **1.3.2. Objetivo Específicos**

- Identificar los procesos avícolas que se llevan a cabo dentro de la finca San Pablo.
- Identificar los estándares para la producción de los pollos de engorde para el desarrollo del sistema de información.

- Desarrollar los módulos del sistema de información para la finca San Pablo que permita la gestión de los procesos de pollos de engorde e ingreso de visitantes.

#### **1.4 Alcance**

Se entrega un sistema de información web para administrar los procesos de los pollos de engorde, gallinas ponedoras y codornices de la finca San Pablo de la UFPS sede Cúcuta.

Los usuarios del sistema son administradores.

El sistema de información consta de los siguientes tres módulos:

- **El módulo administrador permite:**
  - El sistema permite gestionar los procesos de registro, seguimiento y control de los vehículos que ingresan a la finca.
  - El sistema permite gestionar los procesos de registro, seguimiento y control de los visitantes que ingresan a la finca.
  - El sistema permite gestionar los procesos de registro, seguimiento y control de los pollos de engorde de la finca.
  - El sistema permite gestionar los procesos de registro, seguimiento y control de las gallinas ponedoras de la finca.
  - El sistema permite gestionar los procesos de registro, seguimiento y control de las codornices de la finca.
  - El sistema permite administrar los galpones de los pollos de engorde, gallinas ponedoras y las codornices de la finca.
- **El módulo reportes permite:**

- El sistema permite visualizar datos estadísticos para una interpretación fácil y de manera rápida.
- El sistema permite visualizar reportes de la tasa de mortalidad por fechas.
- El sistema permite visualizar reportes de ingreso a la finca.
- El sistema permite visualizar el reporte de los medicamentos con los que cuenta la finca para el engorde de pollos.
- **El módulo medicamentos o sanidad**
  - El sistema permite gestionar los procesos de registro, seguimiento y control de los medicamentos para la cría de los pollos de engorde.
  - El sistema permite dar de baja a medicamentos que estén caducados.

### **1.5 Metodología de desarrollo**

La metodología que se implementó para el desarrollo del proyecto es la metodología XP (Extreme Programming) es una metodología que se enfoca en el desarrollo de software ágil que tiene como ventaja en potenciar las relaciones interpersonales para el desarrollo del software, permitiendo establecer las mejores prácticas de Ingeniería de Software en el desarrollo de proyectos, así mismo mejora la productividad de los mismos y garantizar la calidad del Software desarrollado, haciendo que el cliente pueda superar sus expectativas con el producto o servicio final obtenido.

Esta metodología se caracteriza por llevar una planificación continua de desarrollo de Software, un diseño que se sigue el principio MS – Manténlo Sencillo, *Pressman* en su libro menciona que “...*El diseño es visto como un artefacto en transición que puede y debe*

*modificarse continuamente a medida que avanza la construcción...”, su codificación es de todo el equipo de trabajo además de que su desarrollo es guiado por pruebas, los cuales suelen ejecutarse con mucha frecuencia y finalmente las pruebas las cuales se realizan con el fin de proponer cambios, Pressman define las pruebas como “...La creación de pruebas unitarias antes de que comience la codificación es un elemento clave del enfoque de XP...*

Para el cumplimiento de los objetivos se definieron las siguientes fases:

### **1.5.1 Fase 1: Identificación y exploración**

Inicialmente se contó con una breve descripción de los procesos administrativos que se realizan dentro de la finca San Pablo, una vez analizada y procesada la información y se identificaron las falencias de igual manera se presentó algunas ideas por parte del grupo de desarrollo para proceder con la formalización del problema, requerimientos funcionales y no funcionales. Posteriormente se realizó una reunión con el administrador de la finca para así realizar la respectiva validación de los datos, así como también se definió el alcance de lo que se quiere lograr con el producto en planeación. Tomando como evidencia audios de la reunión y fotos de la finca suministradas por el administrador las cuales se exponen a continuación:



Figura 1. Tomado de la finca San Pablo de la UFPS - criadero de pollos de engorde



Figura 2. - Galpón de pollos de engorde



Figura 3: Tomado de la finca San Pablo de la UFPS-Galpón de gallinas ponedoras



Figura 4: Tomado de la finca San Pablo de la UFPS - producción de carne a partir de pollos de engorde.

### **1.5.2 Fase 2: Producción del sistema de información**

Para dar inicio a la fase de producción previamente se identificó el problema a abordar y sus principales requerimientos tentativos que se requieran para dar solución al problema en cuestión.

En base a los requerimientos se procedió a la construcción de la base de datos mediante un diagrama de entidad relación, en el cual se plasmó las clases principales con sus respectivos atributos y relaciones que tienen entre sí.

### **1.5.3 Fase 3: Entrega**

En esta etapa se definió el alojamiento del sistema desarrollado, se ejecutará el plan de capacitación para los usuarios, se realizará la entrega de la documentación generada en la fase de producción y por último se hará el levantamiento del acta de entrega del producto a la decana de la facultad de ciencias agrarias y ambiente de la Universidad Francisco de Paula Santander seccional Cúcuta.

## **2. Marco metodológico**

### **2.1 Gestión y comprensión del negocio**

#### **Modelo general de procesos Ciclo de vida del software**

El Modelo de Ciclo de Vida que se implementó en el desarrollo del software se tuvo en cuenta el modelo de la cascada, a veces llamado ciclo de vida clásico, en el cual sugiere un

enfoque sistemático y secuencial para el desarrollo de un proyecto de software. Además, este modelo de cascada también se puede denominar Modelo V. Este ciclo de vida es ágil y su forma de trabajar se adaptan a la necesidad del proyecto el cual tiene un grado de complejidad ya que este sistema lo va a utilizar más de una persona y con múltiples funciones a lo cual no lleva a seguir trabajando en este ciclo de vida por su sencillez de trabajo.

Para ello se debe de tener en cuenta las siguientes fases:

- **Comunicación:** Consiste dar inicio al proyecto con la recolección de los requerimientos, los cuales se realizó una entrevista con el administrador de la finca en el cual nos orientó como es la funcionalidad administrativa que se lleva a cabo en la finca.
- **Planeación:** Esta consiste en realizar una estimación de tiempo, costos y recurso humano, así mismo se realizó una estimación de la programación a desarrollar de los subsistemas del proyecto Sistema de información como apoyo en los procesos administrativos de pollos de engorde y gestión de ingreso en la en la finca San Pablo de la UFPS.
- **Modelado:** Para el modelado se realiza un análisis y diseño detenidamente del proyecto, haciendo énfasis en las etapas de cómo se va a ejecutar el proyecto.
- **Construcción:** En la construcción se tiene en cuenta el código y las pruebas que se le realizan al proyecto.
- **Despliegue:** Esta hace referencia a la entrega del producto 100% funcional, además se hacen retroalimentaciones sobre las posibles fallas que este tenga.

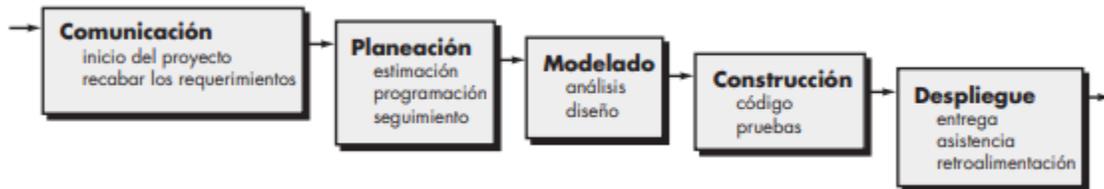


Figura 5. - Proceso del ciclo de vida

Tomado del libro de Pressman “Ingeniería del Software, Un Enfoque práctico” 7° edición

### Tecnología de proceso

Para determinar la tecnología de proceso se debe tener en cuenta el modelo de proceso el cual tiene las siguientes fase: Comunicación, Planeación, Modelado, Construcción y Despliegue, ya teniendo claro esto se puede ejecutar la tecnología del proceso como lo menciona *Roger S. Pressman* en su libro donde “*Las herramientas de tecnología del proceso permiten que una organización de software construya un modelo automatizado de la estructura del proceso, conjuntos de tareas y actividades sombrilla...*” además de ello se puede asignar, vigilar e incluso controlar las actividades y tareas que se van ejecutando en el desarrollo del proyecto, así mismo se puede determinar cuál será el flujo de trabajo para el equipo de desarrollo, igualmente se puede pronosticar el tiempo o costo del proyecto.

Una de las herramientas tecnológicas que se utilizó fue Microsoft Project el cual es un software que brinda el apoyo administrativo para el desarrollo de proyectos en este caso de software ya que tiene la capacidad de desarrollar de planes de acción, asignación de recursos a tareas, dar seguimiento al progreso, administrar presupuesto y analizar cargas de trabajo,

para el desarrollo del software se utilizó Angular que es un framework de diseño de aplicaciones y plataforma de desarrollo para crear aplicaciones de una sola página eficientes y sofisticadas, además se utilizó el Modelo de datos de Cloud Firestore, que es una base de datos NoSQL orientada a los documentos y Firebase Hosting es un servicio de hosting de contenido web con nivel de producción orientado a desarrolladores.

Las tecnologías mencionadas anteriormente se aplicaron en el desarrollo del Sistema de información como apoyo en los procesos administrativos de pollos de engorde y gestión de ingreso en la en la finca San Pablo de la UFPS.

## **2.2 Administración de proyecto de software**

### **Actores y características en general**

#### **- Administrador**

Para el desarrollo de este proyecto, se entregará un usuario administrador. Este usuario podrá disponer de todos los permisos que se desarrollaron dentro del sistema, incluso de la creación de nuevos usuarios administradores para la finca San Pablo de la UFPS – Seccional Cúcuta.

Se encarga de registrar:

- Vehículos, se entiende como vehículos al medio de transporte por el cual las personas externas, estudiantes o docentes visitan la finca.

- Visitantes, se entiende como visitante a cada una de las personas que visita la finca, sean estos estudiantes, docentes o personas ajenas a la universidad.
- Galpones, se entiende como galpón al lugar donde se crían las gallinas ponedoras, pollo de engorde y las codornices.
- Ingreso de registro de ponedoras por semanas, se refiere al registro de los huevos que las gallinas ponedoras ponen por semanas.
- Ingreso de registro de codorniz por semanas, se refiere al registro de los huevos de codorniz que ponen por semana.

### **3. Requerimientos**

#### **3.1 Requerimientos funcionales**

- El sistema debe permitir registrar un galpón.
- El sistema debe permitir ingresar un usuario.
- El sistema debe permitir ingresar vehículos de visitantes y generar reportes específicos.
- El sistema debe permitir ingresar visitantes y generar reportes específicos.
- El sistema debe permitir ingresar la cantidad de pollos de engorde a un galpón y generar reportes específicos.
- El sistema debe permitir ingresar la cantidad de gallinas ponedoras de un galpón y generar reportes específicos.

- El sistema debe permitir ingresar la cantidad de la codorniz de un galpón y generar reportes específicos

### **3.2 Requerimientos no funcionales**

- La interfaz del sistema debe ser fácil de usar.
- El sistema debe contar con el manual de usuario adecuado.
- El sistema debe poseer un diseño Responsivo con la finalidad de garantizar la adecuada visualización en múltiples.
- El sistema debe estar disponible a todas las veces en que un usuario intente acceder.
- El acceso al sistema debe estar restringido con una contraseña y usuario, ya que solo podrán ingresar al sistema las personas que estén registradas.
- El administrador es el único con permisos para realizar cualquier modificación en el sistema

### **3.3 Especificación de requerimientos**

<b>Código CU:</b>	<b>Prioridad:</b>
<b>Nombre:</b>	Inicio de sesión (con correo electrónico o con los servicios de Google)
<b>Descripción:</b>	Loguearse e iniciar sesión en la aplicación, validar el correo electrónico y la contraseña del usuario. Loguearse en el sistema usando el botón de Google
<b>Requerimiento:</b>	Iniciar sesión.
<b>Precondición:</b>	El usuario debe de estar registrado y activo con anterioridad en el sistema
<b>Flujo Normal:</b>	
El usuario ingresa a la página de inicio de sesión, donde se despliega la página de inicio pidiéndole ingresar el correo electrónico y la contraseña con la cual se encuentra registrado en el sistema. El usuario ingresa a la página de inicio de sesión, donde se despliega la página de inicio dónde presiona el botón de Google el cual le solicita que una cuenta de Google para acceder.	
<b>Actor</b>	<b>Sistema</b>
1. El usuario digita el correo electrónico y la clave registrada con anterioridad en el sistema 3. Una vez dentro el usuario tendrá acceso a los módulos según sus permisos.	2. La aplicación validará la información digitada permitiéndole a no ingresar al sistema
<b>Flujo Alterno:</b>	
Si el usuario no puede Loguearse por un dato erróneo debe remitirse a la opción de registro para el sistema reactive su clave con un link enviado al correo	
<b>Actor</b>	<b>Sistema</b>
1.1. El usuario olvidó algún dato de registro por lo que debe ingresar por la opción de olvidaste tu contraseña.	1.2. El sistema le indicará el procedimiento a seguir para el envío de correo de reactivación de clave.
<b>Pos condición:</b>	Luego de realizar el proceso de logueo, el usuario ingresará a los módulos que tenga permiso su perfil.
<b>Requerimientos Especiales:</b>	
<b>Puntos de Extensión:</b>	

<b>Caso de Uso-01</b>	
<b>Código CU:</b>	<b>Prioridad:</b>

<b>Nombre:</b>	Restablecer Contraseña
<b>Descripción:</b>	El sistema le permite al usuario digitar el correo electrónico para validar que este se encuentre en la base de datos, de esta manera encontrar la contraseña asociada y enviar los datos al correo.
<b>Requerimiento:</b>	Recuperar contraseña
<b>Precondición:</b>	El usuario debe de estar registrado y activo con anterioridad en el sistema.
<b>Flujo Normal:</b>	
El usuario ingresa al link olvidaste tu contraseña, introduce el código y el sistema procede validar si el usuario existe a continuación se envía al correo electrónico los datos para restablecer la contraseña.	
<b>Actor</b>	<b>Sistema</b>
Describir cada paso del flujo realizado por un actor. 1.El usuario digita el código anteriormente registrado en el sistema.	Describir cada paso del flujo realizado por algún recurso del sistema. 2. El sistema valida que el usuario se encuentre registrado en la base de datos. 3. El sistema envía al correo electrónico la contraseña del usuario.
<b>Flujo Alternativo:</b>	
Si el usuario no puede restablecer la contraseña por no estar registrado en el sistema, debe de remitirse a la opción de registro.	
<b>Actor</b>	<b>Sistema</b>
Describir cada paso alternativo del flujo realizado por un actor. 1.1. El código digitado por el usuario para el restablecimiento de contraseña no existe, por lo que debe de ingresar a la opción de registrarse en el sistema.	Describir cada paso alternativo del flujo realizado por algún recurso del sistema. 1.2. El sistema le indicará el procedimiento a seguir para el registro.
<b>Poscondición:</b>	Luego de realizar el proceso de restablecimiento de la contraseña, el usuario procederá a ingresar a los módulos que tenga permiso su perfil.
<b>Requerimientos Especiales:</b>	
<b>Puntos de Extensión:</b>	

<b>Código CU:</b>		<b>Prioridad:</b>
<b>Nombre:</b>	Registro semanal para las gallinas ponedoras	
<b>Descripción:</b>	El administrador se encargará de la recolección de la información que considere pertinente para la realización de este caso de uso.	
<b>Requerimiento:</b>	Fecha, cantidad de gallinas ponedoras, cantidad de huevos del lote alimento por unidad (kilogramo), ver fumigaciones y tratamientos.	
<b>Precondición:</b>	El usuario debe haberse logueado primeramente antes de ver los registros	
<b>Flujo Normal:</b>		
<b>Actor</b>		<b>Sistema</b>
1. Accede al registro del galpón de gallinas de engorde.		1. Recolecta la información. 2. Visualiza la información ingresada.
<b>Flujo Alterno:</b>		
El administrador puede acceder a registrar diferentes lotes de gallinas ponedoras.		
<b>Poscondición:</b>		
<b>Requerimientos Especiales:</b>		
<b>Puntos de Extensión:</b>		

<b>Caso de Uso-03</b>		
<b>Código CU:</b>		<b>Prioridad:</b>
<b>Nombre:</b>	Registrar un galpón para los pollos de engorde	
<b>Descripción:</b>	El administrador se encargará de la recolección de la información que considere pertinente para la realización de este caso de uso.	
<b>Requerimiento:</b>	Número de galpón, cantidad de pollos, peso al sacrificio, muertes pre sacrificio, vacunas y medicamentos, fecha de ingreso y egreso.	
<b>Precondición:</b>	El usuario debe haberse logueado primeramente antes de ver los registros	
<b>Flujo Normal:</b>		
<b>Actor</b>		<b>Sistema</b>
2. Accede al registro del galpón de pollos de engorde.		3. Recolecta la información. 4. Visualiza la información ingresada.

<b>Flujo Alternativo:</b>	
El administrador puede acceder a registrar varios galpones.	
<b>Poscondición:</b>	
<b>Requerimientos Especiales:</b>	
<b>Puntos de Extensión:</b>	

<b>Caso de Uso-04</b>	
<b>Código CU:</b>	<b>Prioridad:</b>
<b>Nombre:</b>	Registrar un galpón para las codornices
<b>Descripción:</b>	El administrador se encargará de la recolección de la información que considere pertinente para la realización de este caso de uso.
<b>Requerimiento:</b>	Fecha, cantidad de codornices, alimento por unidad (kilogramo), cantidad de huevos del lote, vermifugaciones y tratamientos.
<b>Precondición:</b>	El usuario debe haberse logueado primeramente antes de ver los registros
<b>Flujo Normal:</b>	
.	
<b>Actor</b>	<b>Sistema</b>
1. Accede al registro del galpón de las codornices.	2. Recolecta la información. 3. Visualiza la información ingresada. 4. Guarda la información
<b>Flujo Alternativo:</b>	
El administrador puede acceder a registrar varios galpones.	
<b>Poscondición:</b>	
<b>Requerimientos Especiales:</b>	
<b>Puntos de Extensión:</b>	

<b>Caso de Uso-06</b>	
<b>Código CU:</b>	<b>Prioridad:</b>
<b>Nombre:</b>	Registrar vehículos
<b>Descripción:</b>	El administrador se encargará de la recolección de la información que considere pertinente para la realización de este caso de uso.
<b>Requerimiento:</b>	Información acerca del vehículo tales como placa del vehículo, nombres y apellidos del propietario, fecha y hora de ingreso.

<b>Precondición:</b>	El usuario debe haberse logueado primeramente antes de ver los registros	
<b>Flujo Normal:</b>		
El administrador se encarga del registro de los vehículos que ingresan diariamente a la finca.		
<b>Actor</b>		<b>Sistema</b>
1. Registrar datos de los vehículos.		2. Obtener datos del vehículo 3. Verificar los datos 4. Visualizar información
<b>Flujo Alternativo:</b>		
El sistema podrá registrar datos de vehículo		
<b>Poscondición:</b>		
<b>Requerimientos Especiales:</b>		
<b>Puntos de Extensión:</b>		

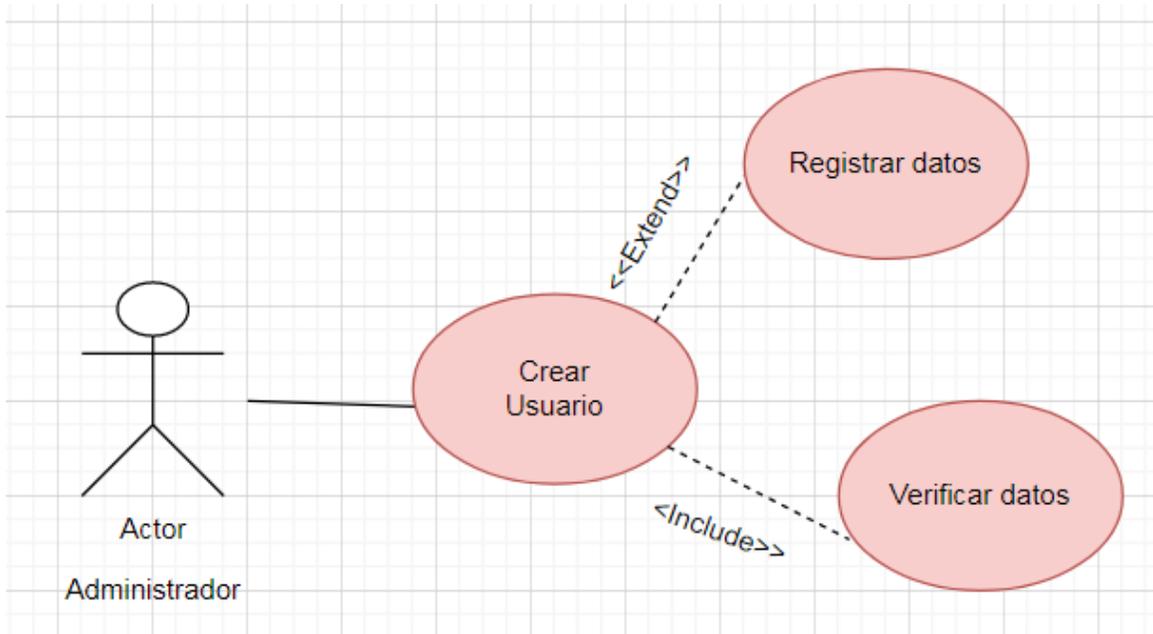
<b>Caso de Uso-07</b>	
<b>Código CU:</b>	<b>Prioridad:</b>
<b>Nombre:</b>	Registrar visitantes
<b>Descripción:</b>	El administrador se encargará de la recolección de la información que considere pertinente para la realización de este caso de uso.
<b>Requerimiento:</b>	Información acerca del visitante nombres, apellidos, código estudiantil, rol de la universidad, fecha y hora de ingreso.
<b>Precondición:</b>	El usuario debe haberse logueado primeramente antes de ver los registros
<b>Flujo Normal:</b>	
El administrador se encarga del registro de los visitantes que ingresan diariamente a la finca.	
<b>Actor</b>	<b>Sistema</b>
5. Registrar datos de los visitantes.	6. Obtener datos del visitante 7. Verificar los datos 8. Visualizar información
<b>Flujo Alternativo:</b>	
El sistema podrá registrar datos de visitantes.	
<b>Poscondición:</b>	
<b>Requerimientos Especiales:</b>	
<b>Puntos de Extensión:</b>	

<b>Caso de Uso-08</b>	
<b>Código CU:</b>	<b>Prioridad:</b>
<b>Nombre:</b>	Generar reporte por número de galpón
<b>Descripción:</b>	El administrador se encargará de la recolección de la información que considere pertinente para la realización de este caso de uso.
<b>Requerimiento:</b>	Información para generar el reporte por galpones para pollos de engorde es el Id del galpón.
<b>Precondición:</b>	El usuario debe haberse logueado primeramente antes de ver los registros
<b>Flujo Normal:</b>	
El administrador se encargará de ingresar las fechas entre las que desea generar el reporte.	
<b>Actor</b>	<b>Sistema</b>
9. Ingresar las fechas para generar el reporte.	10. Validar las fechas para generar el reporte 11. Obtener datos del reporte
<b>Flujo Alterno:</b>	
<b>Poscondición:</b>	
<b>Requerimientos Especiales:</b>	
<b>Puntos de Extensión:</b>	

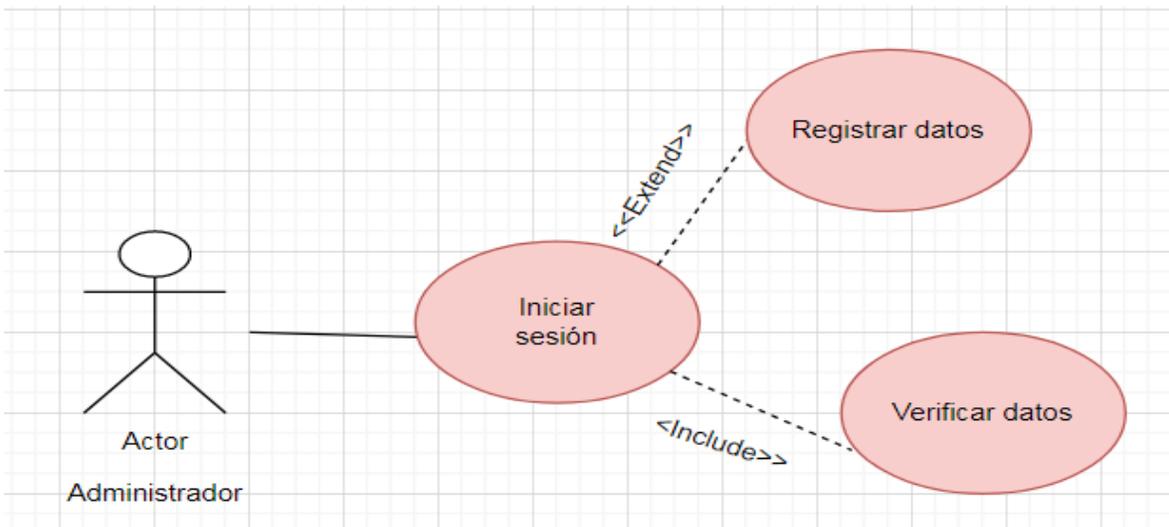
## 4. Modelo de análisis

### 4.1 Casos de uso generales

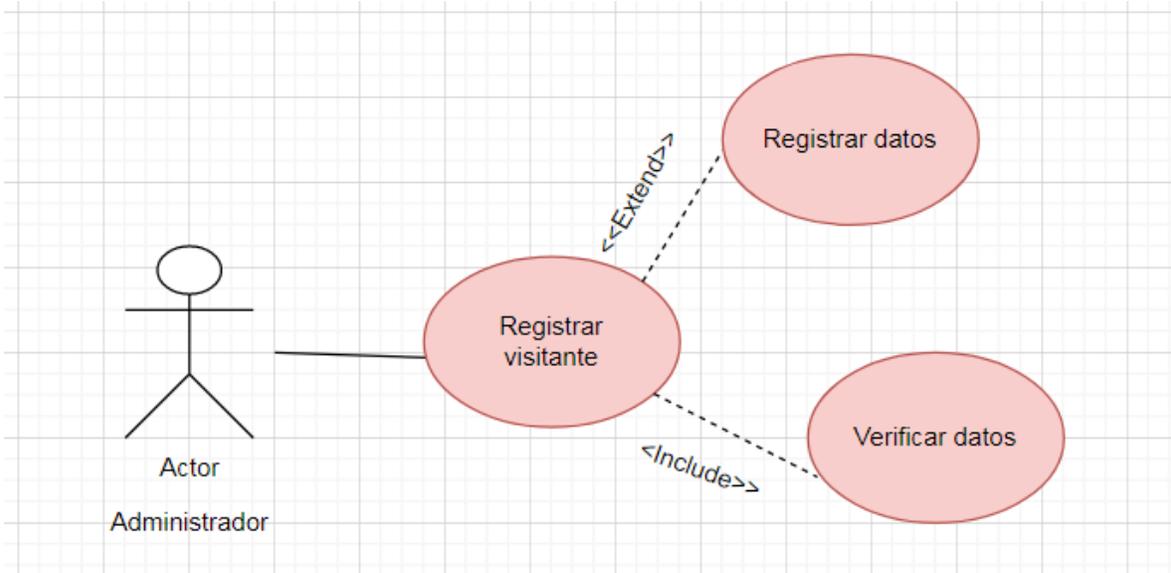
#### 4.1.1 Diagrama Caso De Uso Crear Usuario



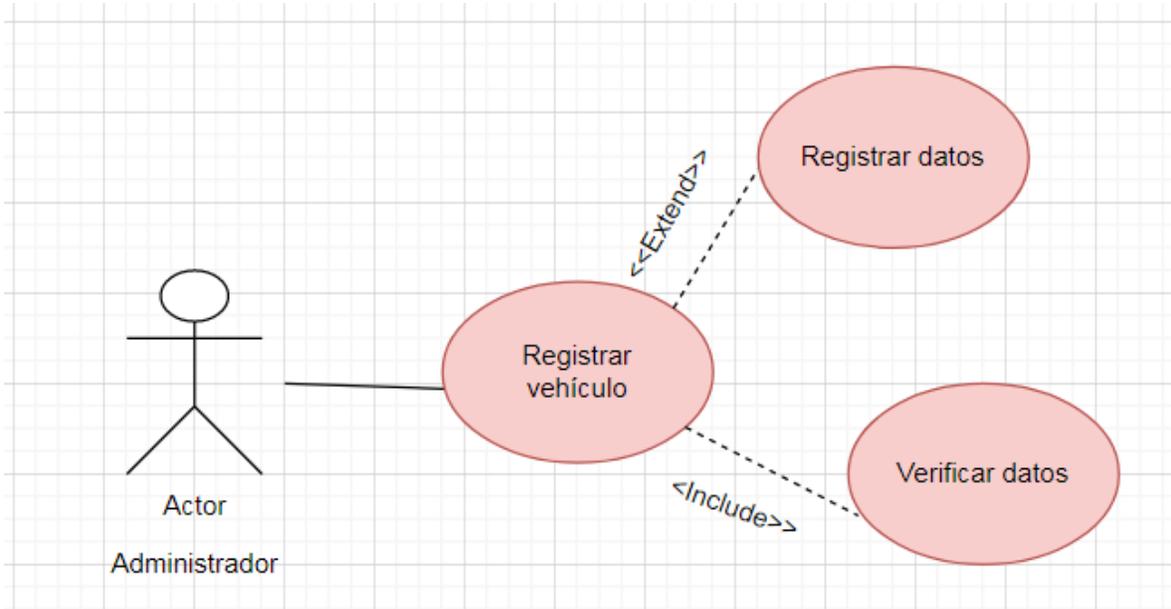
#### 4.1.2 Diagrama De Caso De Uso Iniciar Sesión



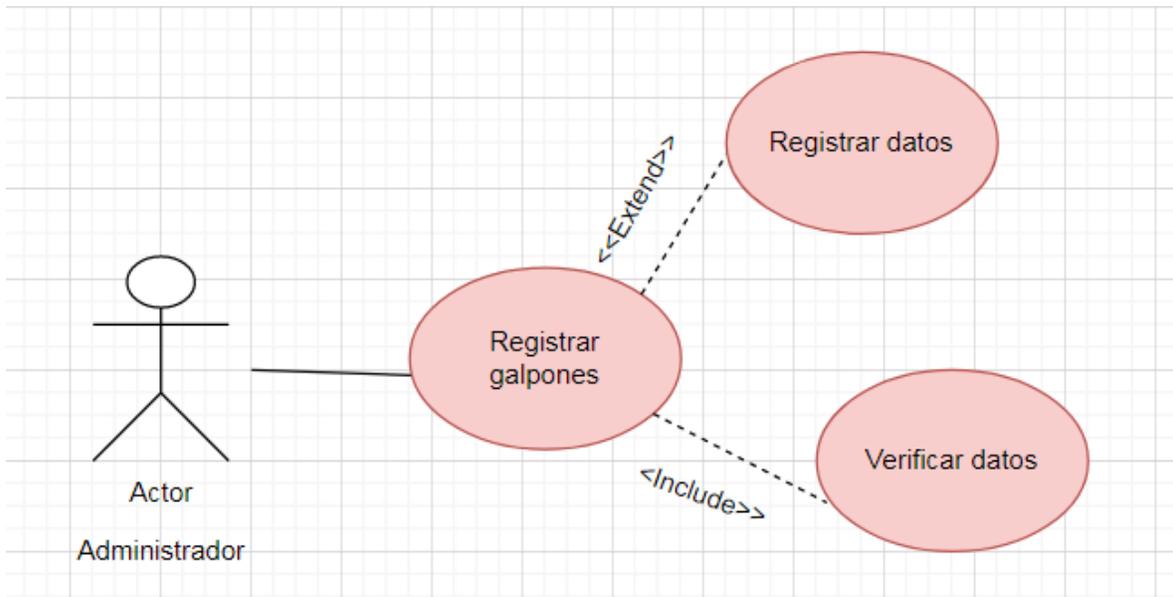
#### 4.1.3 Diagrama De Caso De Uso Registrar Visitante



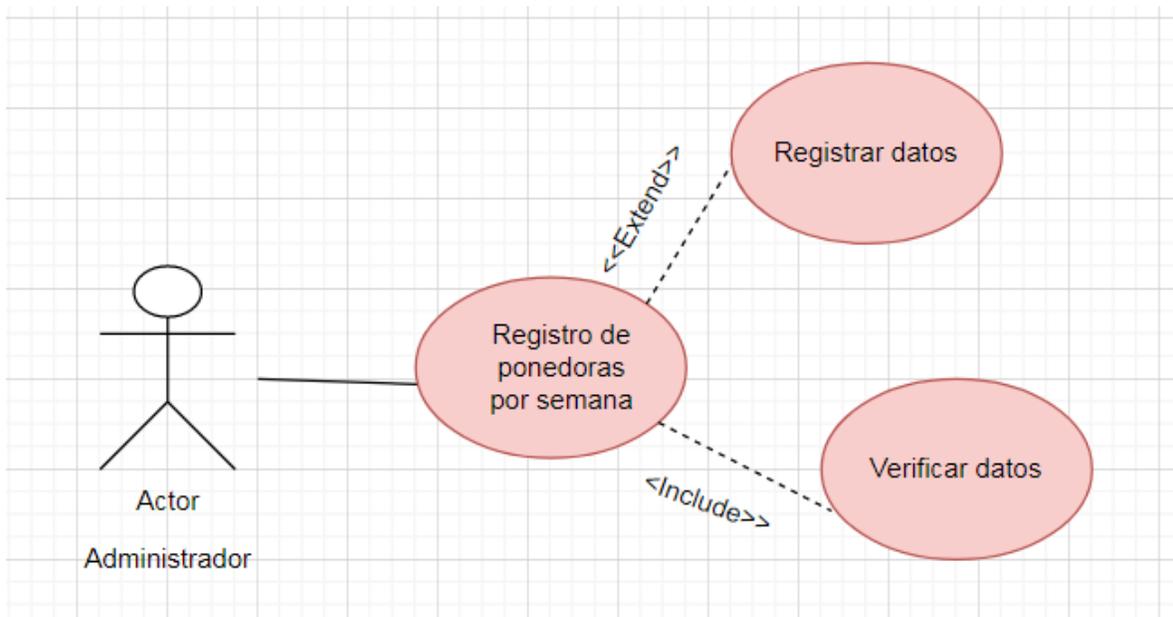
#### 4.1.4 Diagrama De Caso De Uso Registrar Vehículos



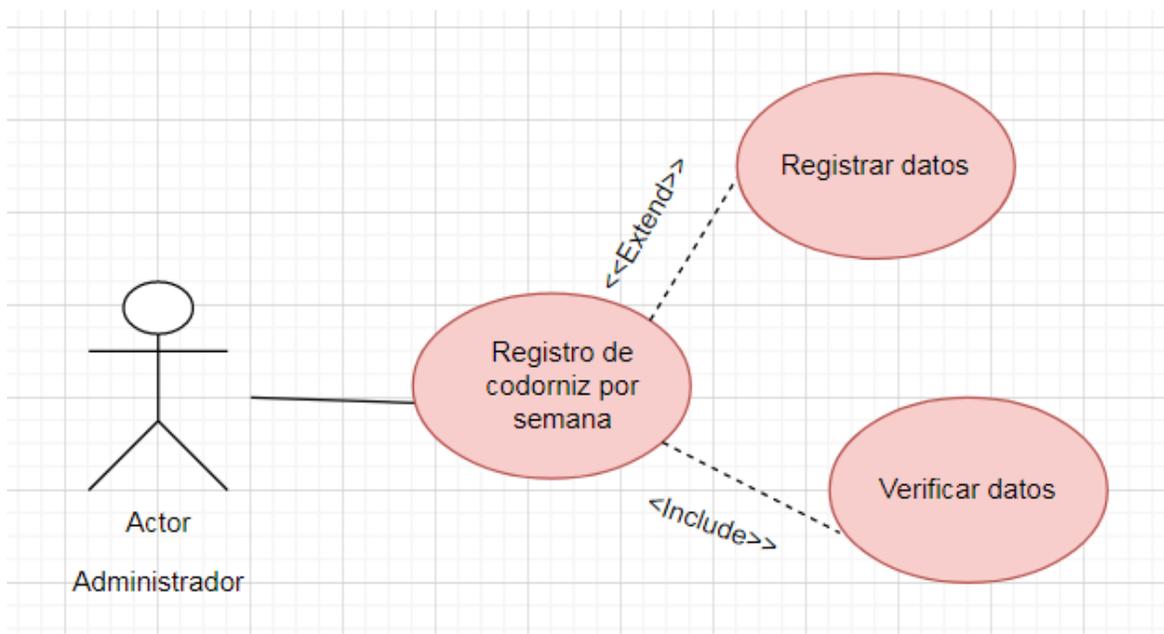
#### 4.1.5 Diagrama De Caso De Uso Registrar Galpones



#### 4.1.6 Diagrama De Caso De Uso Registro De Ponedoras Por Semana



#### 4.1.7 Diagrama De Caso De Uso Registro De Codorniz Por Semana



### 5. Arquitectura del sistema

La arquitectura que implementa en el desarrollo del sistema fue angular el cual posee un marco completo de modelo - vista - controlador (MVC), el cual proporciona una guía más clara sobre cómo debe estructurarse el sistema y así mismo ofrece un flujo de datos bidireccional al tiempo que proporciona DOM real, a continuación una imagen en cual se visualiza el patrón MVC.

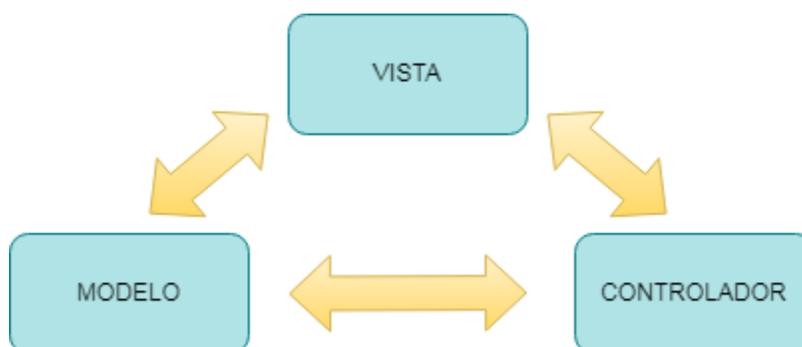


Figura 6. Patrón MVC

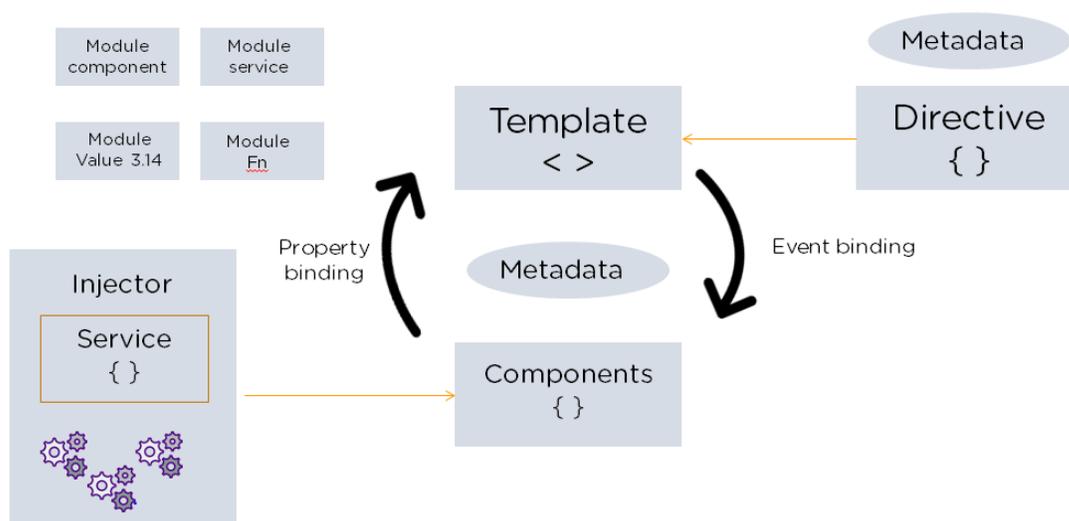


Figura 7. Arquitectura de Angular

Imagen tomada

de: [https://www.simplilearn.com/ice9/free\\_resources\\_article\\_thumb/Angular\\_Architecture-What\\_is\\_Angular.PNG](https://www.simplilearn.com/ice9/free_resources_article_thumb/Angular_Architecture-What_is_Angular.PNG)

## 5.1 Autenticación

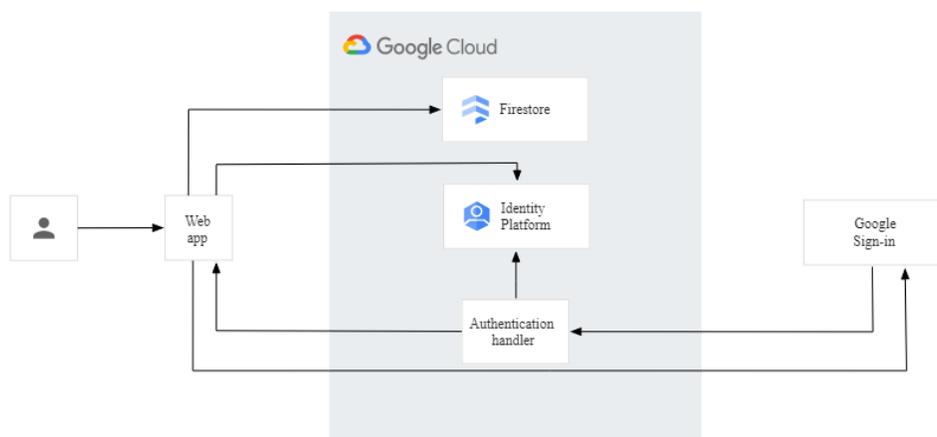


Figura 8- Diagrama de autenticación de Google

Imagen tomada de:

<https://cloud.google.com/architecture/authenticating-users-to-firestore-with-identity-platform-and-google-identities?hl=es-419>

## 5.2 Patrones de diseño

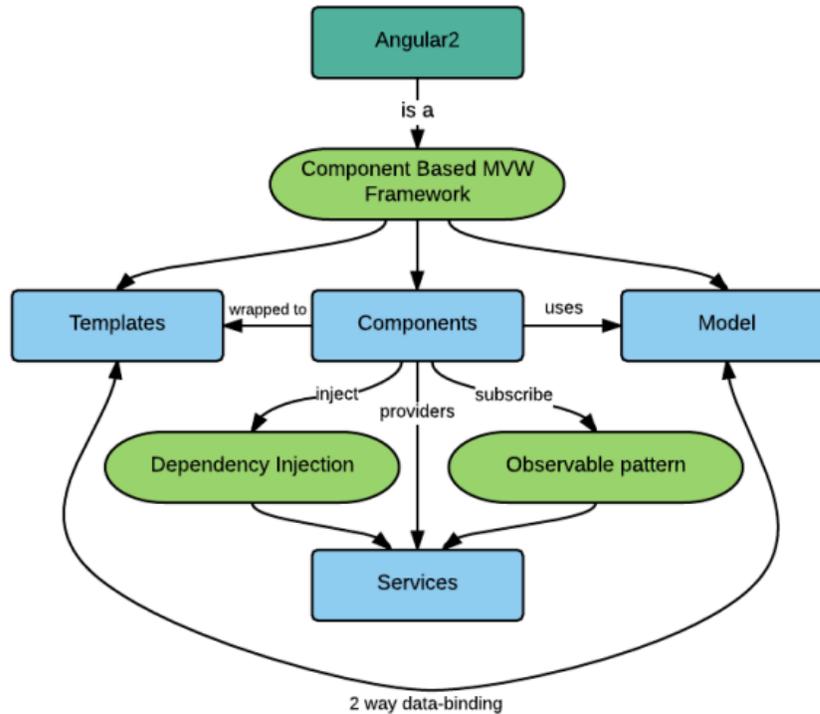


Figura 9- Patrones de diseño usados en la aplicación

Imagen tomada de:

[https://digibuo.uniovi.es/dspace/bitstream/handle/10651/39133/TFM\\_PabloSotoMedina.pdf?sequence=9](https://digibuo.uniovi.es/dspace/bitstream/handle/10651/39133/TFM_PabloSotoMedina.pdf?sequence=9)

## 6. Diagrama de despliegue

A continuación se muestra mediante un gráfico donde se describe el modelo despliegue que se implementó en el desarrollo del proyecto, se puede observar el comportamiento de los nodos y sus respectivos paquetes, así como interactúan entre ellos.

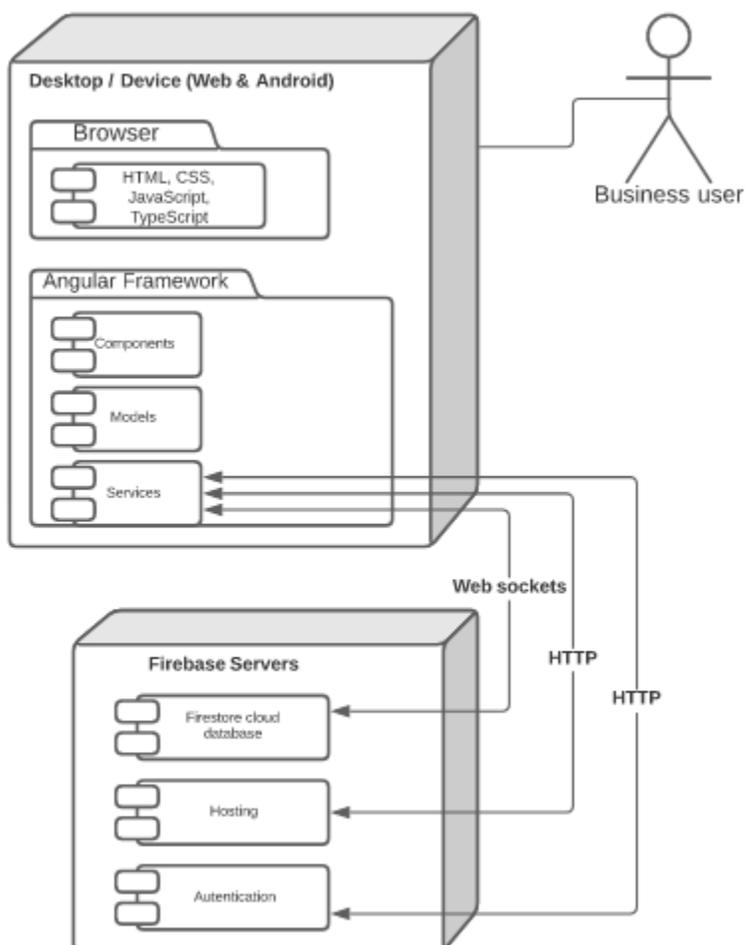


Figura 10- Vista de despliegue

### 6.1 Modelo de datos de Cloud Firestore

Cloud Firestore es una base de datos NoSQL orientada a los documentos. A diferencia de una base de datos SQL, no hay tablas ni filas; En su lugar, almacenas los datos en documentos, que se organizan en colecciones.

Cada documento contiene un conjunto de pares clave-valor. Cloud Firestore está optimizado para almacenar grandes colecciones de documentos pequeños.

Todos los documentos se deben almacenar en colecciones, y pueden contener subcolecciones y objetos anidados. Además, ambos pueden incluir campos primitivos, como strings, o tipos de objetos complejos, como listas.

Las colecciones y los documentos se crean de manera implícita en Cloud Firestore; solo debes asignar datos a un documento dentro de una colección. Si la colección o el documento no existen, Cloud Firestore los crea.

<https://firebase.google.com/docs/firestore/data-model?hl=es-419>

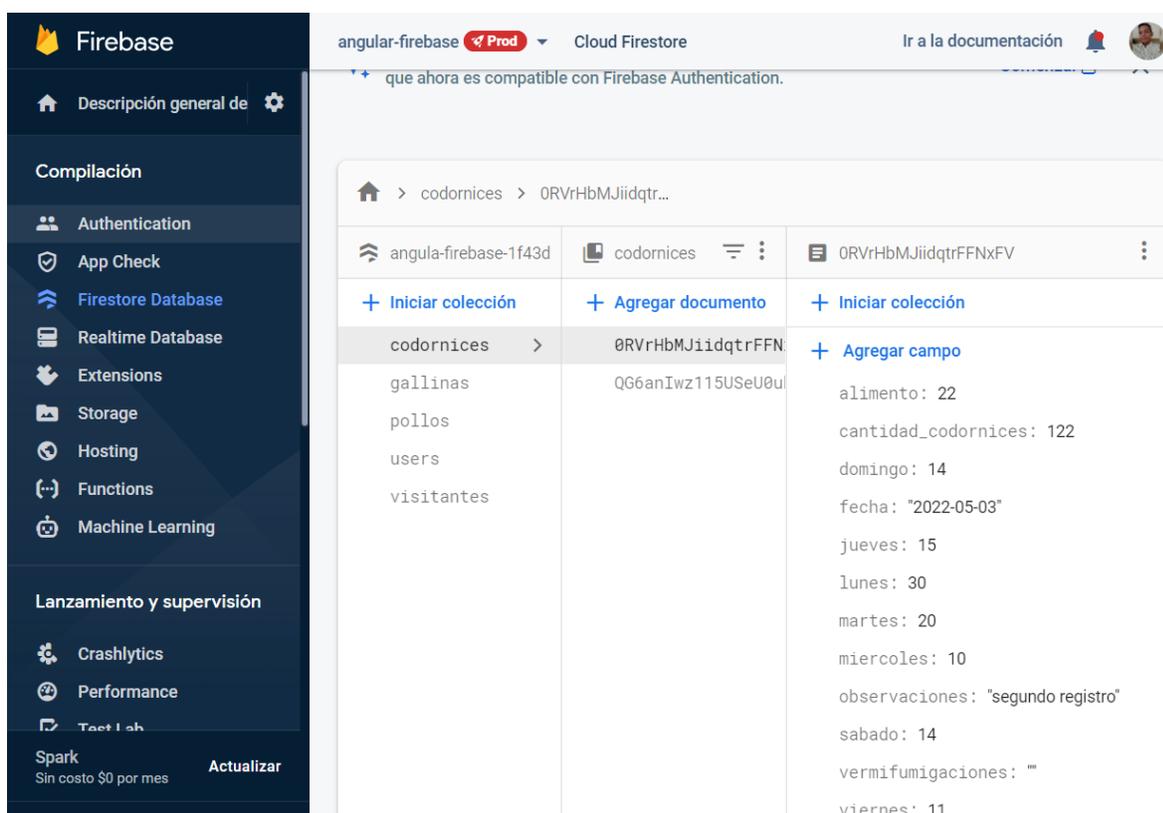


Figura 11. Visualización de colección de datos

## 7. Pruebas del software

### 7.1 Contexto de las Pruebas

#### 7.1.1 Elementos de Prueba

Se realizarán pruebas a los siguientes componentes:

- Componente login.
- Componente granjero.
- Componente reportes.
- Componente visitantes.

### **7.1.2 Diseño de casos de prueba.**

El documento de especificación y diseño de casos de prueba se encuentra en el siguiente enlace:

<https://docs.google.com/spreadsheets/d/1TMWfurx1run8248jjFjjcSC9MAkiXwE3/edit?usp=sharing&oid=101251633943748870081&rtpof=true&sd=true>

### **7.1.3 Técnicas de prueba.**

A continuación se menciona el estilo de técnica que se utilizarán para el diseño de los casos de prueba

- Pruebas de Casos de Uso
- Técnica de prueba de condición
- Prueba de caja blanca y de caja negra

#### **7.1.3.4 Pruebas de caja blanca**

A continuación se expone el desarrollo de las pruebas unitarias realizadas al código fuente como parte del desarrollo de pruebas de caja blanca, en busca de medir la eficiencia del código y determinar si cumple con el requerimiento del caso de uso.

- **Prueba 1 - Validación de componente principal**

Basada en la técnica de prueba de condición, para asegurarnos que nuestro componente principal trabaja correctamente y no vamos a tener inconvenientes con la dependencia de otros componentes.

```

describe('AppComponent', () => {
  beforeEach(async () => {
    await TestBed.configureTestingModule({
      declarations: [
        AppComponent,
      ],
      imports: [ RouterTestingModule ],
    }).compileComponents();
  });

  it('Debe ser un componente valido', () => {
    const fixture = TestBed.createComponent(AppComponent);
    const app = fixture.componentInstance;
    expect(app).toBeTruthy();
  });
});

```

Figura 12. Prueba 1- Validación de componente principal

- **Prueba 2 - Autenticación**

En esta prueba se toma en cuenta el caso de uso “Iniciar sesión”, se evalúa el comportamiento del componente con su respectivo servicio asociado utilizando valores de prueba, dando como resultado el valor esperado que es el usuario autenticado con su correo y contraseña.

```

it('Autentica Correo y contraseña ->iniciarSesion(email:string, password:string)', () => {
  const fixture = TestBed.createComponent(LoginComponent);
  const app = fixture.componentInstance;
  const mockedData = {email:'yezid16@gmail.com', password: 'contraseña123'};
  const expectedResult = true;
  const result = app.iniciarSesion(mockedData.email, mockedData.password);
  expect(result).toEqual(expectedResult);
});

```

Figura 13. Prueba 2- Autenticación

### 7.1.3.5 Pruebas de caja negra

A continuación se expone el desarrollo de las pruebas realizadas al aplicativo como parte del desarrollo de pruebas de caja negra, en busca de medir el completo cumplimiento de los requerimientos funcionales.

- **Prueba 1 - Validación de campos numéricos en formulario**

El propósito de esta prueba es evaluar el comportamiento de un requerimiento funcional en específico el cual es que para el registro de ponedoras en el campo “cantidad de ponedoras” solo se admiten valores numéricos para realizar el posterior conteo.

**Nuevo registro semanal (Ponedoras)**

Los campos con \* son obligatorios

**Fecha:\***  
05/06/2022

**Cant de ponedoras: \***  
Ingrese solo valores numericos  
Cantidad de ponedora

**Alimento x unidad (Kg):**  
Cantidad de alimento

**Vermifugaciones:**  
vermifugaciones y tratamientos

**Cantidad de huevos del lote**

lunes	Martes	Miercoles	Jueves
Viernes	Sabado	Domingo	

**Observaciones:**  
observaciones

**registrar**

Figura 14. prueba caja negra 1 - Validación de campos numéricos en formulario

## 7.2 Herramientas tecnológicas utilizadas en las pruebas del proyecto

### 7.2.1 Descripción

Para la realización del diseño de casos de prueba y la ejecución de los mismos se utilizaron las técnicas y funcionalidades de las herramientas de software más adecuadas para

llevar a cabo esta tarea de testing, con el fin de proveer un amplio volumen de información para su respectivo análisis y tratamiento.

## 7.2.2 Jira

Provee herramientas de planificación y hojas de ruta para que los equipos de desarrollo logren gestionar a los interesados, los presupuestos y los requisitos de las funcionalidades de sus productos de software.

### 7.2.2.1 Resultados obtenidos

Key	Summary	P	Status	Assignee
GRAD-TC-8	Reporte de galpón	↑	Done	Edinson Gomez
GRAD-TC-7	Reporte de huevos	↑	Done	Edinson Gomez
GRAD-TC-6	Registro de codornices	↑	Done	Edinson Gomez
GRAD-TC-5	Registro de ponedoras	↑	Done	Edinson Gomez
GRAD-TC-4	Registro de galpón de pollos	↑	Done	Edinson Gomez
GRAD-TC-3	Registro de vehículos	↑	Done	Edinson Gomez
GRAD-TC-2	Registro de visitantes	↑	Done	Edinson Gomez
GRAD-TC-1	Login con los servicios de google	↑	Done	Edinson Gomez

Figura 15. Diseño de casos de prueba con Jira

Se creó un ciclo de pruebas único en el cual se ejecutaron la totalidad de casos de prueba arrojando un 100% de éxito en los casos de prueba.

### 7.2.3 Karma (Jasmine)

Jasmine es una suite de testing que sigue la metodología Behavior Driven Development. Tiene cosas muy buenas como que no requiere un DOM para hacer los tests y la sintaxis es bastante sencilla de entender.

Karma es el test-runner, es decir, el módulo que permite automatizar algunas de las tareas de las suites de testing, como Jasmine. Karma, además, ha sido desarrollado directamente por el equipo de Angular, lo cual, en cierto modo, nos da alguna garantía de que va a seguir existiendo de aquí a un tiempo, por lo que parece una buena opción.

#### 7.2.3.1 Resultados obtenidos

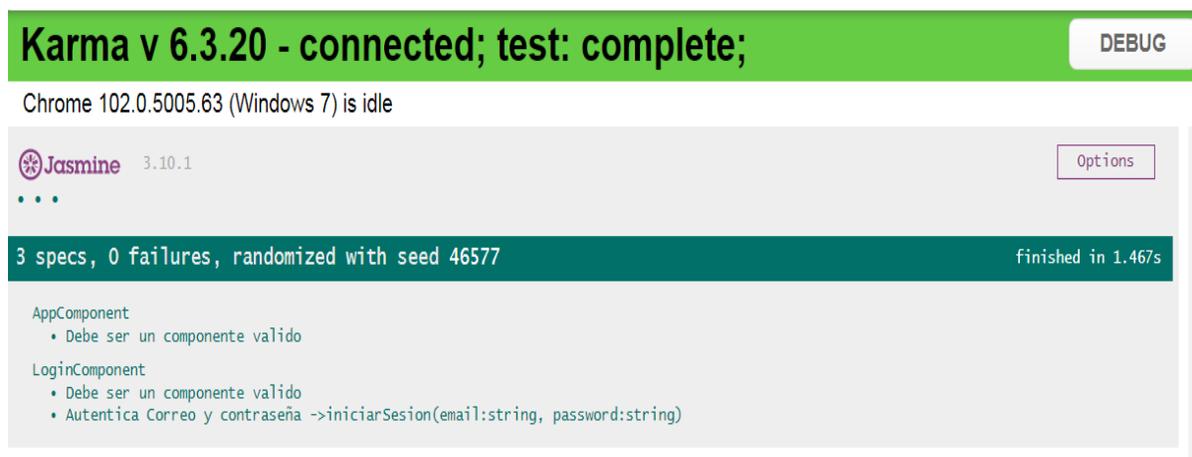


Figura 16. resultado de pruebas unitarias

**All files**

45.83% Statements 33/72   42.85% Branches 3/7   21.21% Functions 7/33   42.64% Lines 29/68

Press *n* or *j* to go to the next uncovered block, *b*, *p* or *k* for the previous block.

Filter:

File		Statements	Branches	Functions	Lines				
app		100%	3/3	100%	0/0	100%	1/1	100%	2/2
app/components/login		90%	18/20	100%	0/0	50%	3/6	88.88%	16/18
app/services/autenticacion		22.91%	11/48	42.85%	3/7	11.53%	3/26	21.27%	10/47
environments		100%	1/1	100%	0/0	100%	0/0	100%	1/1

Figura 17. porcentaje de código evaluado en cada componente

## 8. Referencias

Gamez, A. , Morales, M. , & Ramirez, C.(2018). Estado del arte sobre problemáticas financieras de las pymes en Bogotá, Colombia y América Latina. *ECONÓMICAS CUC*, 39(2), 77–94.