

**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO**

**CARRERA:
COMPUTACIÓN**

**Trabajo de titulación previo a la obtención del título de:
Ingenieros en Ciencias de la Computación**

**TEMA:
DESARROLLO DE APLICATIVO WEB PARA AGRICULTURA DE PRECISIÓN EN
EMPRESAS FLORÍCOLAS UTILIZANDO DATOS DE CULTIVOS DE FLORES DE
LA EMPRESA “AGROSCAN”.**

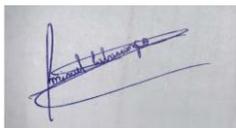
**AUTORES:
MISAELE DUARDO CABASCANGO GARCIA
DIEGO GONZALO OSORIO GUAMINGA**

**TUTOR:
HOLGER RAÚL ORTEGA MARTÍNEZ**

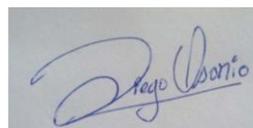
Quito, octubre del 2021

CESIÓN DE DERECHOS DE AUTOR

Nosotros Misael Eduardo Cabascango Garcia, Diego Gonzalo Osorio Guaminga con documento de identificación N° 1755043328 y N° 1723677702, manifestamos nuestra voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación intitulado: DESARROLLO DE APLICATIVO WEB PARA AGRICULTURA DE PRECISIÓN EN EMPRESAS FLORÍCOLAS UTILIZANDO DATOS DE CULTIVOS DE FLORES DE LA EMPRESA “AGROSCAN” mismo que ha sido desarrollado para optar el título de INGENIEROS EN CIENCIAS DE LA COMPUTACIÓN, en la Universidad Politécnica Salesiana quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente. En aplicación a lo determinado en la Ley de Propiedad Intelectual en nuestra condición de autores nos reservamos los derechos morales de la obra antes citada. En concordancia, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.



.....
Misael Eduardo Cabascango Garcia
1755043328



.....
Diego Gonzalo Osorio Guaminga
1723677702

Quito, octubre de 2021

DECLARATORIA DE COAUTORÍA DEL DOCENTE TUTOR

Yo declaro que bajo mi dirección y asesoría fue desarrollado el proyecto técnico con el tema: **DESARROLLO DE APLICATIVO WEB PARA AGRICULTURA DE PRECISIÓN EN EMPRESAS FLORÍCOLAS UTILIZANDO DATOS DE CULTIVOS DE FLORES DE LA EMPRESA “AGROSCAN”** realizado por Misael Eduardo Cabascango Garcia y Diego Gonzalo Osorio Guaminga, obteniendo un producto que cumple todos los requisitos estipulados por la Universidad Politécnica Salesiana para ser considerado como trabajo final de titulación.

Quito, octubre 2021

A handwritten signature in blue ink, consisting of a vertical line on the left, a horizontal line, and a large, stylized loop on the right.

Holger Raúl Ortega Martínez, MSc.
CI: 1708182728

ÍNDICE GENERAL

INTRODUCCIÓN.....	1
ANTECEDENTES	1
PROBLEMA.....	6
JUSTIFICACIÓN.....	7
OBJETIVOS	7
OBJETIVO GENERAL.....	7
OBJETIVOS ESPECÍFICOS	7
METODOLOGÍA.....	8
CAPÍTULO 1: ANÁLISIS EXPLORATORIO DE LOS DATOS.....	9
1.1 VARIABLES	9
1.1.1 VARIABLES INDEPENDIENTES	9
1.1.2 VARIABLE DEPENDIENTE.....	10
1.2 NÚMERO DE EJEMPLOS	10
1.3 RANGO DE FECHAS	10
1.4 RANGOS Y ESTADÍSTICOS BÁSICOS.....	10
CAPÍTULO 2: MODELO.....	12
2.1 CONSTRUCCIÓN DE LA MATRIZ DE DATOS.....	12
2.1.1 VENTANA DESLIZANTE AJUSTADA	12
2.2 PREPROCESAMIENTO DE DATOS.....	13
2.2.1 LIMPIEZA DE DATOS.....	13
2.2.2 NORMALIZACIÓN.....	13
2.2.3 OVERSAMPLING	13
2.2.4 UNDERSAMPLING	13
2.3 DEFINICIÓN DE CLASES	13
2.4 MODELO DE CLASIFICACIÓN	15

2.4.1 CLASIFICACIÓN BINARIA	15
2.4.2 CLASIFICACIÓN MULTINOMIAL CON SUBCLASIFICACIÓN	16
CAPÍTULO 3: DESARROLLO DE LA APLICACIÓN.....	17
3.1 ANÁLISIS DE REQUERIMIENTOS.....	17
3.1.1 ALCANCE.....	17
3.1.2 USUARIOS DE LA APLICACIÓN	18
3.1.3 HISTORIAS DE USUARIO	18
3.3.3 PRODUCT BACKLOG	25
3.3.4 PLANIFICACIÓN DE LOS SPRINTS.....	26
3.2 DISEÑO	26
3.2.1 CASOS DE USO	26
3.2.1.1 Casos de uso módulo de ingreso de datos.....	28
3.2.1.2 Casos de uso módulo de recomendaciones	29
3.2.1.3 Módulo de Reportes	30
3.2.1.4 Módulo de gestión de usuarios.....	32
3.2.2 DIAGRAMA DE CLASES	34
3.2.3 DIAGRAMA CONCEPTUAL DE BASE.....	35
3.2.4 DIAGRAMA NAVEGACIONAL.....	37
3.3 DESARROLLO.....	38
3.3.1 HERRAMIENTAS DE DESARROLLO	38
3.3.1.1 Lenguaje Python	38
3.3.1.2 Flask.....	39
3.3.1.3 Keras	39
3.3.1.4 SQLAlchemy.....	39
3.3.1.5 MySQL	39

3.3.1.6 PyCharm IDE	40
3.3.1.7 Jira	40
3.3.2 ARQUITECTURA DE LA APLICACIÓN	40
3.3.2.1 Modelo	40
3.3.2.2 Vista.....	41
3.3.2.3 Controlador.....	41
3.3.3 EJECUCIÓN DE LOS SPRINTS	43
3.3.3.1 PRIMER SPRINT.....	43
3.3.3.2 SEGUNDO SPRINT.....	46
3.3.3.3 TERCER SPRINT	47
Historia de usuario Calificar Recomendación	48
Historia de usuario Crear Usuario	48
3.3.3.4 CUARTO SPRINT	49
3.3.3.5 QUINTO SPRINT.....	52
3.3.3.6 SEXTO SPRINT.....	54
3.4 IMPLEMENTACIÓN.....	57
3.4.1 Configuración del servidor en Linode	57
3.4.2 Configuración de la aplicación	61
3.5 PRUEBAS.....	71
3.5.1 PRUEBA DE FUNCIONALIDAD.....	71
3.5.2 PRUEBA DE EFICIENCIA	76
CONCLUSIONES.....	80
RECOMENDACIONES.....	81
REFERENCIAS	82

INDÍCE DE TABLAS

Tabla 1 <i>Resumen de variables recolectadas</i>	11
Tabla 2 <i>Actividades Agrícolas</i>	14
Tabla 3 <i>Agrupación Actividades Agrícolas</i>	14
Tabla 4 <i>Formato de historia de usuario</i>	19
Tabla 5 <i>Historia de usuario ingresar datos de una medición</i>	19
Tabla 6 <i>Historia de usuario visualizar recomendación</i>	20
Tabla 7 <i>Historia de usuario calificar recomendación</i>	20
Tabla 8 <i>Historia de usuario reporte de Recomendaciones</i>	21
Tabla 9 <i>Historia de usuario Reporte de Mediciones</i>	21
Tabla 10 <i>Historia de usuario iniciar sesión</i>	22
Tabla 11 <i>Historia de usuario recuperar contraseña</i>	22
Tabla 12 <i>Historia de usuario crear usuario</i>	23
Tabla 13 <i>Historia de usuario modificar usuario</i>	23
Tabla 14 <i>Historia de usuario visualizar usuarios</i>	24
Tabla 15 <i>Historia de usuario eliminar usuario</i>	24
Tabla 16 <i>Descripción de Caso de Uso Ingresar medición</i>	28
Tabla 17 <i>Descripción de Caso de Uso Visualizar recomendación</i>	29
Tabla 18 <i>Descripción de Caso de Uso Retroalimentar recomendación</i>	29
Tabla 19 <i>Descripción de Caso de Uso Consultar recomendaciones</i>	30
Tabla 20 <i>Descripción de Caso de Uso Consultar recomendaciones</i>	31
Tabla 21 <i>Descripción de Caso de Uso Imprimir reporte de mediciones</i>	32
Tabla 22 <i>Descripción de Caso de Uso Crear Usuario</i>	32
Tabla 23 <i>Descripción de Caso de Uso Editar Usuario</i>	33
Tabla 24 <i>Descripción de Caso de Uso Eliminar Usuario</i>	33
Tabla 25 <i>Prueba de funcionalidad 1</i>	71
Tabla 26 <i>Prueba de funcionalidad 2</i>	72

Tabla 27 Prueba de Funcionalidad 3	72
Tabla 28 Prueba de Funcionalidad 4	73
Tabla 29 Prueba de Funcionalidad 5	73
Tabla 30 Prueba de Funcionalidad 6	74
Tabla 31 Prueba de Funcionalidad 7	74
Tabla 32 Prueba de Funcionalidad 8	75
Tabla 33 Prueba de Funcionalidad 9	75
Tabla 34 Prueba de Funcionalidad 10	76

ÍNDICE DE FIGURAS

Figura 1 <i>Ventana deslizante</i>	12
Figura 2 <i>Product Backlog</i>	25
Figura 3 <i>Diagrama de caso de uso de la aplicación Asistente Florícola</i>	27
Figura 4 <i>Diagrama de clases de la aplicación</i>	35
Figura 5 <i>Diagrama conceptual de base de datos</i>	37
Figura 6 <i>Diagrama navegacional de la aplicación</i>	38
Figura 7 <i>Diagrama de componentes</i>	42
Figura 8 <i>Scrum board para el Sprint 1</i>	43
Figura 9 <i>Formulario 1 para ingreso de una medición</i>	44
Figura 10 <i>Formulario 2 para ingreso de una medición</i>	45
Figura 11 <i>Mensaje al ingresar medición</i>	46
Figura 12 <i>Scrum Board para el Sprint 2</i>	46
Figura 13 <i>Recomendación obtenida del modelo</i>	47
Figura 14 <i>Tablero Sprint 3</i>	47
Figura 15 <i>Calificación de una recomendación</i>	48
Figura 16 <i>Formulario para la creación de un nuevo usuario</i>	49
Figura 17 <i>Scrum Board para el Sprint 4</i>	50
Figura 18 <i>Interfaz para la búsqueda e impresión de reporte de recomendaciones</i>	50
Figura 19 <i>Interfaz para visualizar los usuarios de la aplicación</i>	51
Figura 20 <i>Eliminación de un usuario</i>	52
Figura 21 <i>Scrum board para el Sprint 5</i>	52
Figura 22 <i>Interfaz para imprimir reporte de mediciones</i>	53
Figura 23 <i>Interfaz de inicio de sesión</i>	54
Figura 24 <i>Scrum board para el Sprint 6</i>	54
Figura 25 <i>Formulario para Editar un usuario</i>	55
Figura 26 <i>Interfaz de recuperación de contraseña</i>	56
Figura 27 <i>Correo electrónico enviado desde la aplicación</i>	57

Figura 28	<i>Acceso al servidor mediante SSH.....</i>	58
Figura 29	<i>Archivo de configuración de host del servidor.....</i>	59
Figura 30	<i>Generación de llave par privada pública.....</i>	60
Figura 31	<i>Asignación de permisos a directorio .ssh y sus archivos.....</i>	60
Figura 32	<i>Configuración de firewall del servidor.....</i>	61
Figura 33	<i>Lista de paquetes de Miniconda.....</i>	62
Figura 34	<i>Ambiente virtual creado en el servidor.....</i>	62
Figura 35	<i>Archivo de configuración modificado.....</i>	63
Figura 36	<i>Configuración de MySQL Server en el servidor</i>	64
Figura 37	<i>Archivo de configuración de NGINX.....</i>	65
Figura 38	<i>Activación del tráfico de http y TCP.....</i>	65
Figura 39	<i>Archivo de configuración de Gunicorn.....</i>	66
Figura 40	<i>Pantalla de administración de Namecheap.....</i>	67
Figura 41	<i>Creación de dominio en el servidor en Linode.....</i>	67
Figura 42	<i>Creación de un registro AAAA.....</i>	68
Figura 43	<i>Tiempo de carga.....</i>	77
Figura 44	<i>Prueba de carga con 200 peticiones.....</i>	78
Figura 45	<i>Prueba de carga con 1000 peticiones.....</i>	78

RESUMEN

En este proyecto se describe el desarrollo e implementación de una aplicación web para agricultura de precisión en empresas florícolas, con datos de cultivos de flores proporcionados por la empresa Agroscan. Esta aplicación tiene embebido un modelo de clasificación de series de tiempo generado en un estudio anterior, el cual utiliza mediciones diarias de tres variables tomadas de una parcela de cultivo y devuelve como salida una actividad agrícola recomendada para aplicarse en la parcela de la que se tomó los datos. La aplicación tendrá dos tipos de usuarios: técnico agrícola y administrador. El técnico agrícola será quien ingrese los datos de las mediciones en el aplicativo web, mientras que el administrador realizará la gestión de usuarios.

La metodología de desarrollo de software empleada es Scrum. El aplicativo se desarrolló en lenguaje Python versión 3.7 con el framework Flask, versión 2.1.1, el cual está diseñado para el desarrollo de aplicaciones web. El modelo de clasificación fue construido utilizando la librería Keras versión 2.3.1.

Como resultado se obtuvo una aplicación web con un modelo de Deep Learning capaz de facilitar la implementación de agricultura de precisión en cultivos de rosas.

ABSTRACT

This project describes the development and implementation of a web application for precision agriculture in flower companies, with data from rose crops provided by the Agrosan company. This application will have embedded a time series classification model generated in a previous study, which uses daily measurements of three variables taken from a crop plot and returns as an output a recommended agricultural activity to be applied in the plot from which the data was taken. The application will have two types of users: agricultural technician and administrator. The agricultural technician will be the one who enters the data of the measurements in the web application, while the administrator will carry out the user management.

The software development methodology used is Scrum. The application will be developed in Python language version 3.7 with the Flask framework, version 2.1.1, which is designed for the development of web applications. The classification model was built using the Keras library version 2.3.1.

As a result, a web application with a Deep Learning model is obtained, which could facilitate the implementation of precision agriculture in rose plantations.

INTRODUCCIÓN

ANTECEDENTES

El sector florícola es uno de los sectores agrícolas más importantes en la economía del Ecuador. Las flores producidas en el país se encuentran dentro de los 5 principales productos de exportación (Asociación de Productores y Exportadores de Flores 2020, [Expoflores], p. 2). De las especies de flores exportadas, las rosas generan la mayor cantidad de ingresos de este sector (Expoflores, 2020, p. 5).

Como en cualquier actividad económica, en el sector florícola se desea lograr una mayor producción con pérdidas mínimas, para lo cual es necesario emplear métodos y técnicas. Una de las técnicas actualmente empleadas es la conocida como “agricultura de precisión”.

La agricultura de precisión es una estrategia de gestión que recoge, procesa y analiza datos temporales, espaciales e individuales y los combina con otras informaciones para respaldar las decisiones de manejo de acuerdo con la variabilidad estimada, y así mejorar la eficiencia en el uso de recursos, la productividad, la calidad, la rentabilidad y la sostenibilidad de la producción agrícola. (International Society of Precision Agriculture, s.f.)

Por otro lado, según Zofío Jiménez (2013), “se denomina aplicación web al software que reside en un ordenador, denominado servidor web, que los usuarios pueden utilizar a través de Internet o de una intranet, con un navegador web, para obtener los servicios que ofrezca” (p. 7).

Meza, Zambrano, & Ortega (2021) en su estudio tuvieron como objetivo general crear un modelo que permita simular las decisiones de un experto en producción florícola utilizando mediciones del suelo y el riego. La metodología utilizada es cuantitativa. En los resultados se pudo observar una dependencia positiva entre la conductividad eléctrica y el pH. Por otro lado, se tienen dependencias negativas entre las variables densidad/pH y densidad/conductividad eléctrica. Como resultado de la agrupación en dos clases (hacer algo, no hacer nada) se obtuvo un conjunto de datos con una clase dominante y aplicando la técnica oversampling se logró

equiparar la cantidad de registros por clase. La subagrupación de los resultados de la clasificación binaria también entregó un conjunto de datos con 4 clases desbalanceadas. Finalmente se observó que el modelo de clasificación multiclase entrenado y evaluado con los datos balanceados tiene un 77.23% de exactitud. Se concluyó que: i) La técnica de la ventana deslizante resulta conveniente y efectiva al tratar este problema, ii) Clasificar, agrupar y manejar las actividades agrícolas como etiquetas del conjunto de características permitió visibilizar un problema de aprendizaje supervisado y favoreció la precisión de procesos predictivos posteriores, iii) el desbalance de datos afecta negativamente al proceso de clasificación dando como resultado porcentajes muy altos de error. Al recolectar más datos, se resuelve dicho problema a la vez que la capacidad de aprendizaje de la red neuronal aumenta, y mejora la precisión de ésta, y iv) el modelo desarrollado obtuvo mejores resultados al tener una configuración donde las capas intermedias de la red neuronal tienen funciones de activación intercaladas entre ReLu y sigmoideal, y la capa de salida siempre es configurada con función de activación softmax.

Rodríguez, Gualotuña, & Grilo (2017) en su estudio tuvieron como objetivo general desarrollar un sistema de monitoreo de condiciones ambientales aplicado a la agricultura de alta precisión dentro de un ambiente controlado para un invernadero de rosas en Ecuador, mediante la implementación de una Red de Sensores Inalámbrica, en inglés Wireless Sensor Network (WSN), dentro del invernadero. La metodología utilizada es cuantitativa y se presenta como un caso de estudio dentro de los invernaderos de rosas disponibles para la Universidad de las Fuerzas Armadas - ESPE, Ecuador. En los resultados se demostró que la regresión lineal es un algoritmo de predicción aceptable para el propósito del sistema de monitoreo, ya que obtuvo un error absoluto relativo del 13,18% que corresponde a un error absoluto medio de 0,807°C. Con redes neuronales se obtuvo el peor resultado con un error absoluto medio de 0.998 °C y con máquinas de soporte vectorial se obtuvo un error absoluto relativo de 11,33%,

correspondiente a un error absoluto medio de 0,698 ° C, el cual fue el mejor resultado y que por consecuencia proporcionará el mejor modelo de predicción para el sistema. Se concluyó que:

- i) el sistema propuesto otorgará a la floricultura una herramienta de seguimiento permanente de tamaño, duración, estado sanitario y un tallo floral proporcional y bien formado de los cultivos, permitiendo a los agricultores tomar acciones preventivas o correctivas según sea el caso, ii) la aplicación cliente constituye una herramienta útil para los agricultores, ya que no los obliga a estar físicamente presentes en el invernadero para ver qué sucede en tiempo real y iii) a partir de este trabajo, uno de los enfoques futuros será el diseño de una solución que permita el control automático de datos sobre estados fenológicos, mediante el uso de cámaras de alta definición y procesamiento de imágenes.

Bonilla Segovia, Dávila Rojas, & Villa Quishpe (2021) en su estudio tuvieron como objetivo general presentar una revisión de las aplicaciones de la IA en el análisis del suelo, manejo de cultivos, manejo de malezas y enfermedades por administración. Los tipos de investigación fueron: de campo, debido a que los datos e información se obtuvieron mediante encuestas realizadas a los empleados de los invernaderos y su experiencia con la tecnología de la información exploratoria, ya que al hacer el análisis del uso de técnicas de inteligencia artificial para el análisis de suelos para el sector agrícola pueden surgir nuevos temas de investigación a futuro; y descriptiva, ya que depende de la interpretación. Como técnica de investigación se utilizó la encuesta.

Entre los resultados más relevantes se puede mencionar que: i) La aplicación de los materiales orgánicos son esenciales para mejorar la calidad del suelo, por ejemplo la aplicación de abono y estiércol mejora la porosidad y agregación, ii) para la temperatura del suelo el rango óptimo según esta investigación es de 17 a 23, para la humedad es de 5 a 50 y para pH el valor óptimo es 7, iii) las malezas siguen causando problemas en los cultivos de floricultura debido a las prácticas culturales deficientes. El control de malezas más rentable es la prevención y el uso de

químicos y métodos no químicos para un programa de manejo de plagas verdaderamente integrado (MIP) y iv) las principales limitaciones de la implementación de la IA en el sector agrícola son: a) Tiempo de respuesta y precisión, b) Big Data Requerido, c) Método de Implementación, d) Alto costo de datos y e) Flexibilidad.

Se concluyó que i) el rendimiento de la industria agrícola puede aumentar con la ayuda de la tecnología logrando minimizar problemas como la falta de sistemas de riegos efectivos, malezas, problemas con el monitoreo de plantas, etc, ii) Las diferentes técnicas impulsadas por la IA brindan mejoras en el manejo de técnicas aplicadas en la agricultura, aunque se pueden dar limitantes en cuanto al manejo e implementación de los equipos y al conocimiento de los mismos por parte del personal que labora en el sector agrícola, iii) se necesita de un entrenamiento por parte de los dueños de fincas e invernaderos a los agricultores e ingenieros en sistemas con el fin de conocer el uso de tecnologías que brindan mejorías a los suelos y que logran predecir situaciones a futuro y iv) el empleo de la IA para obtener características del cultivo como altura de la planta, textura y contenido del suelo ofrece mayores ventajas en el sector agrícola tanto en campo abierto como cerrado y un acceso más controlado a la información.

(Islam, y otros, 2018) en su estudio tuvieron como objetivo general el desarrollo de un sistema de monitoreo inteligente basado en Internet de las cosas, en inglés IoT, para la práctica agrícola, que incluye una aplicación web cliente y una aplicación en Android; y el diseño de un sistema de control para riego energéticamente eficiente. La metodología utilizada es cuantitativa. El sistema incorpora 3 tipos de nodo: nodo sensor, nodo de puerta de enlace, nodo de riego; base de datos, aplicación web y aplicación de Android. En los resultados se mostró que: i) el éxito de envío de datos del nodo sensor y almacenamiento de estos datos en la base de datos fue del 99%; ii) hay 5.164 datos almacenados en la base de datos, que incluye todos los datos del nivel de prueba del sistema; iii) se logró la recolección de datos de los sensores, en tiempo real, y su

visualización tanto en la aplicación web como en la aplicación para Android; y, iv) el sistema de irrigación propuesto plantea ser automático, basado en las condiciones del campo para así cuidar a los cultivos de problemas de sobreriego, así como también para ahorrar energía. Como conclusiones se obtuvo que: i) las WSN's son una tecnología importante a la hora de aplicar agricultura de precisión con datos recolectados en tiempo real; ii) se espera que este sistema puede ser adoptado en todos los campos donde se requiera monitores y seguimiento; y, iii) el sistema desarrollado demuestra que tiene su relevancia y eficiencia tecnológica para implementarse en agricultura de precisión.

Triantafyllou, Tsouros, Sarigiannidis, & Bibi (2019) en su estudio tuvieron como objetivos: i) proponer un modelo de arquitectura de referencia genérico; y, ii) presentar y discutir las tecnologías que incorporan las siete capas del modelo de arquitectura: la capa de sensores, la capa de enlace, la capa de encapsulación, la capa de middleware, la capa de configuración, la capa de administración y la capa de aplicación. La metodología utilizada es cuantitativa con enfoque de meta síntesis cualitativa. En los resultados se encontraron desafíos para las capas de sensores y capa de red del modelo propuesto. Para la capa de sensores el desafío principal es lograr un funcionamiento eficiente y continuo durante mucho tiempo en un entorno natural, teniendo en cuenta el cambio climático y las intervenciones de la vida silvestre. En la capa de red, los desafíos de utilizar tecnologías WSN e IoT incluyen las capacidades computacionales limitadas de los nodos sensores. La memoria restringida de los nodos inhabilita a éstos para manejar grandes cantidades de datos de comunicación y procedimientos de interconexión basados en clústeres. También se encontró que, en muchos casos, la falla de un sistema de monitoreo avanzado puede deberse a la distancia geográfica, cultural o socioeconómica entre los diseñadores del sistema y la comunidad de usuarios a la que se destina. Se concluyó que: i) las aplicaciones agrícolas de IoT permiten a los agricultores recopilar y analizar datos significativos; ii) los grandes propietarios de tierras y los pequeños agricultores deberían

aprovechar el potencial del mercado de IoT para la agricultura mediante la instalación de tecnologías inteligentes para aumentar la competitividad y la sostenibilidad de sus producciones; y, iii) el rápido crecimiento de la población obliga a los agricultores a satisfacer la demanda mediante la implementación de soluciones de IoT agrícolas de manera apropiada.

PROBLEMA

En la empresa florícola de la cual se obtuvo los datos para construir el modelo de clasificación se ha constatado, a través de reuniones con el técnico agrícola que trabaja ahí, que los datos de los cultivos se obtienen de manera manual, sin la ayuda de sensores, y que dichos datos se registran en hojas de cálculo de Excel, en archivos diferentes para diferentes variables y mediciones. Esto hace que la manipulación de los datos sea difícil y deba hacerse de forma manual, especialmente cuando se necesita integrar la información.

Los datos son imprescindibles para poder aplicar agricultura de precisión, por ello el presente proyecto pretende crear un aplicativo web que permita gestionar la información de manera más ágil y automatizada y que brinde recomendaciones de las actividades agrícolas a realizarse diariamente en los cultivos.

JUSTIFICACIÓN

Entre los motivos que nos llevaron a la creación de un asistente agrícola está que la contratación de un experto externo es un gasto alto para una florícola que no tiene uno de planta. En ese sentido, la aplicación puede constituirse en una solución atractiva para plantaciones de mediana envergadura, cuyo presupuesto no alcance para costear un experto humano. Por otra parte, la aplicación permitirá tener una mejor gestión de la información, que ayudará a los técnicos agrícolas a conocer mejor sus cultivos; esto gracias a que la aplicación trabajará con una base de datos relacional. Adicionalmente, tiene un modelo de Deep Learning que arroja predicciones de las actividades agrícolas a partir de los datos.

Un directivo de Agrosca y el técnico agrícola mencionado anteriormente nos han manifestado, en entrevistas llevadas a efecto con motivo de este trabajo, que un aplicativo web como el propuesto es una necesidad en la industria agrícola. Nuestro proyecto permitiría que las industrias florícolas tengan acceso a una nueva alternativa de bajo costo y al alcance de todos.

OBJETIVOS

OBJETIVO GENERAL

Desarrollar un aplicativo web para agricultura de precisión en empresas florícolas utilizando datos de cultivos de flores de la empresa “Agrosca”.

OBJETIVOS ESPECÍFICOS

- Crear un módulo de ingreso de datos para el aplicativo web.
- Crear un módulo que presente recomendaciones al agricultor en base a las variables censadas: pH, densidad y conductividad eléctrica, de tal forma que él pueda aceptarlas o modificarlas de acuerdo con su experiencia, para lo cual este podrá emitir observaciones aceptando o modificando la recomendación.

- Crear un módulo de seguridad para controlar el acceso al aplicativo web, únicamente para los usuarios autorizados.
- Crear un módulo para generar reportes de las actividades sugeridas por el aplicativo web, incluyendo las observaciones efectuadas por el agricultor.
- Evaluar el rendimiento del aplicativo web aplicando métricas de funcionalidad, fiabilidad y eficiencia.

METODOLOGÍA

La metodología de desarrollo de software a emplear será Scrum. Las entregas en Scrum son progresivas y se realizan en iteraciones llamadas Sprints (Flewelling, 2018). Esta metodología constituye un marco de trabajo ligero lo cual permite a diferentes usuarios (personas, equipos y organizaciones) encontrar soluciones adaptables a sus problemas sin importar lo complejo que sean (Schwaber & Sutherland, 2020).

Scrum es una metodología de desarrollo de software ágil y define 3 mecanismos: Product Backlog, Sprint Backlog e Incremento.

Existen 3 responsabilidades dentro del equipo Scrum que son: desarrolladores, el propietario del producto y el Scrum Master (Schwaber & Sutherland, 2020).

Para la obtención de los requerimientos de usuario se utilizó como técnica de investigación, la entrevista.

CAPÍTULO 1: ANÁLISIS EXPLORATORIO DE LOS DATOS

En este capítulo se resume el proceso exploratorio de los datos, realizado previamente a la construcción del modelo de clasificación, que se incorporará en el aplicativo.

1.1 VARIABLES

1.1.1 VARIABLES INDEPENDIENTES

Se consideraron 3 variables de todo el conjunto de datos como variables independientes para construir el modelo de clasificación: pH, densidad y conductividad eléctrica. A continuación, se ofrece una breve descripción de cada una.

pH: El pH es una variable química que nos permite medir el grado de acidez de una sustancia (García, 2019). Según (Anthura, 2018), el pH juega un papel muy importante en los cultivos. Un agua demasiado ácida ejerce un impacto negativo sobre la planta y su entorno radicular.

Densidad: (Fundación Aquae, s.f.) señala que la densidad del agua en gramos es 1000 kg/m³, y que depende de factores como la temperatura, la presión (su valor equivale a 1 atmósfera) y salinidad (en el caso de agua destilada que no contiene sal). A 20° C, la densidad del agua sería aproximadamente 998 kg/m³ o 0,998 g/cm³.

Conductividad eléctrica: Capacidad de una sustancia de conducir la corriente eléctrica y es lo contrario de la resistencia. La conductividad es una variable que se controla en muchos sectores, desde la industria química a la agricultura. Esta variable depende de la cantidad de sales disueltas presentes en un líquido y es inversamente proporcional a la resistividad del mismo (Infoagro, s.f.).

1.1.2 VARIABLE DEPENDIENTE

La variable dependiente del modelo se denomina Actividad Agrícola. Los datos de esta variable provienen de un experto agrícola empleado en la plantación donde se tomó la data. Dicho profesional registró la actividad agrícola que, en su criterio, debía realizarse cada día, a partir del análisis de los valores de las variables independientes en las plantas en los días previos.

1.2 NÚMERO DE EJEMPLOS

Para el proyecto se cuenta con los datos proporcionados por una empresa floricultora ecuatoriana que trabaja con distintas variedades de rosas como Akito, Cream & Cream, Explorer, Freedom2, Freedom15 y Vendela. Los datos fueron recolectados diariamente de varias plantaciones y durante todas sus etapas fenológicas, obteniendo un total de 1558 ejemplos.

1.3 RANGO DE FECHAS

Los 1558 datos obtenidos han sido recolectados en 1558 días, desde el 11/05/2015 hasta el 15/08/2019, es decir, cubren un rango de aproximadamente 4 años.

1.4 RANGOS Y ESTADÍSTICOS BÁSICOS

El resumen de las variables recolectadas, por variedad de planta, y sus siglas, se presenta en la Tabla 1. En este trabajo, el proceso de aprendizaje automático utiliza las mediciones de pH, CE y D para la variedad de rosa Akito, tomadas del sustrato líquido suministrado diariamente durante 1558 días.

Tabla 1*Resumen de variables recolectadas*

Variable	Descripción	Unidad	Máximo	Mínimo
CE	Conductividad eléctrica (planta y suelo)	Siemens	6.78	0.56
pH	Grado de acidez y alcalinidad	Hidrones	8.0	5.5
D	Densidad de agua	Kilogramo	1.85	0.26
Cap.	Capacidad de retención del agua	Humedad	10.0	6.0
Luminosidad	Luminosidad	Lumen	10.0	2.5
Temp	Temperatura grados	Centígrados	38.7	16.4

Elaborado por Misael Cabascango y Diego Osorio.

CAPÍTULO 2: MODELO

2.1 CONSTRUCCIÓN DE LA MATRIZ DE DATOS

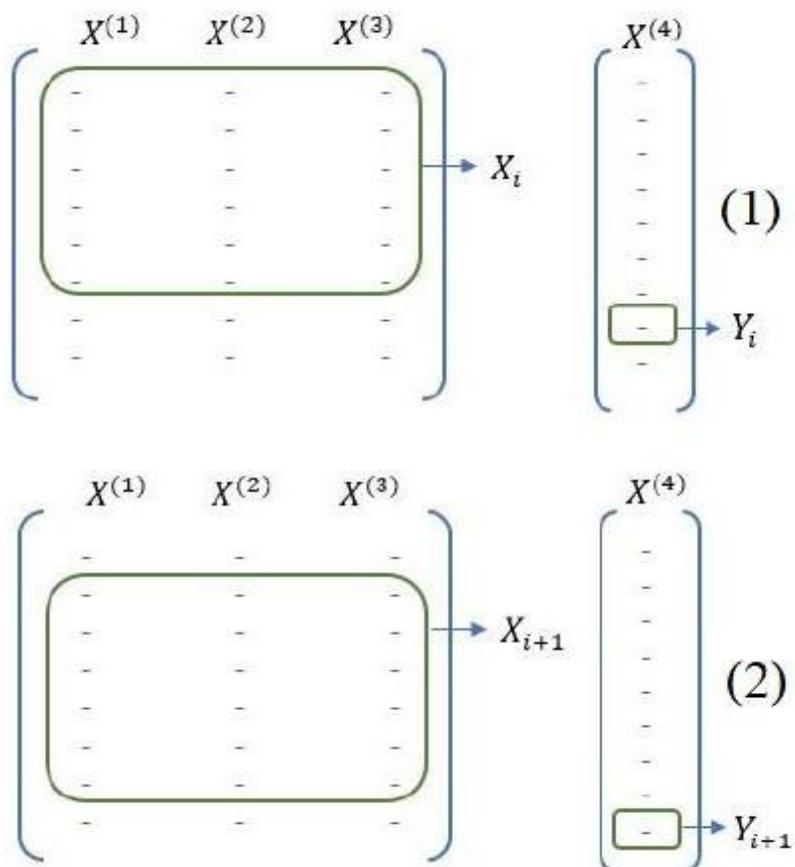
2.1.1 VENTANA DESLIZANTE AJUSTADA

El modelo desarrollado por (Meza, Zambrano, & Ortega, 2021), que es el que se encuentra embebido en nuestra aplicación web, utiliza el método de la ventana deslizante. La dimensión de la ventana utilizada en dicho trabajo se determina empíricamente, basándose en el método heurístico y la distribución de los datos.

La ventana deslizante empleada en el proyecto comprende un total de $n=10$ registros consecutivos, de los cuales se toma la media para formar el correspondiente vector de características X_i y cada onceavo registro corresponde a la etiqueta de clase Y_i , como se muestra en la Figura 1.

Figura 1

Ventana deslizante



Adaptada de Ventana Deslizante, (Meza, Zambrano, & Ortega, 2021)

2.2 PREPROCESAMIENTO DE DATOS

El preprocesamiento de datos es un conjunto de técnicas que limpia, normaliza, busca, filtra y selecciona los datos susceptibles de análisis para lograr el máximo provecho.

2.2.1 LIMPIEZA DE DATOS

Para la limpieza de los datos se ha optado en incluir operaciones como corrección de datos, reducción de detalles que son innecesarios, detección de discrepancias e identificación de datos sucios. (Meza, Zambrano, & Ortega, 2021).

2.2.2 NORMALIZACIÓN

La normalización tiene como objetivo la comparación entre diferentes variables mediante la realización de cálculos que extienden o comprimen valores hasta obtener una escala y un rango estándar (Meza, Zambrano, & Ortega, 2021).

2.2.3 OVERSAMPLING

Según (Brownlee, 2020), oversampling es un enfoque que implica la duplicación aleatoria de ejemplos de la clase minoritaria.

2.2.4 UNDERSAMPLING

Según (Brownlee, 2020), undersampling implica borrar aleatoriamente ejemplos de la clase mayoritaria.

2.3 DEFINICIÓN DE CLASES

Para la implementación del modelo de clasificación se etiquetan las actividades agrícolas registradas en días específicos, utilizando las etiquetas de clase que se muestran en la Tabla 2.

Tabla 2*Actividades agrícolas*

N°	Actividad agrícola	Cardinalidad
1	Incorporar compost	2
2	Cambio de lugar	7
3	Drenchado	35
4	Trinchar camas	4
5	Levantar camas	8
6	Aplicar sulfato de calcio	7
7	Aplicar sulfato de amonio	8
8	Aplicar nitrato de amonio	5
9	Aplicar nitrato de calcio	2

Elaborado por Misael Cabascango y Diego Osorio.

Posteriormente se disminuyó el número de clases, agrupando las variables que presentan cierta similitud, según se muestra en la Tabla 3. La intención de este paso fue resolver, al menos en parte, el problema de las clases desbalanceadas.

Tabla 3*Agrupación de actividades agrícolas*

Etiqueta	Actividad agrícola
1	Incorporar Compost o Cambio de lugar
2	Drenchado, Trinchar camas o Levantar camas
3	Aplicar sulfato de calcio o Aplicar nitrato de calcio

4	Aplicar sulfato de amonio o Aplicar nitrato de amonio
---	---

Elaborado por Misael Cabascango y Diego Osorio.

2.4 MODELO DE CLASIFICACIÓN

Con el análisis del conjunto de datos se definió el propósito del modelo a utilizar. (Meza, Zambrano, & Ortega, 2021). Nuestra tarea consiste en determinar una etiqueta para un conjunto de datos, por lo tanto, se trata de un problema de clasificación con series de tiempo.

2.4.1 CLASIFICACIÓN BINARIA

Antes de realizar una clasificación multiclase y para reducir la cantidad de clases de nuestra variable dependiente se ejecuta una clasificación binaria que separa el conjunto de datos en dos clases: “hacer algo” o “no hacer nada”, donde “hacer algo” engloba todas las actividades agrícolas.

Para evaluar el modelo de clasificación binario, se divide el conjunto de datos en 70% para entrenamiento y 30% para la evaluación.

Al tratarse de una clasificación binaria habitual y puesto que entre las topologías comunes está la Radial Basis Network (RBF), nuestro modelo de red neuronal para esta clasificación binaria consta de 3 capas; una primera de entrada con 3 neuronas, la segunda con 30 neuronas y función de activación ReLu y por último una tercera capa de salida con función de activación sigmoide. El entrenamiento se realizó en 1000 épocas utilizando entropía cruzada binaria, sensibilidad y especificidad como métricas de precisión validado con el 30% de los datos. El resultado de la precisión de nuestro modelo fue 89.60% (+/- 13.71%).

2.4.2 CLASIFICACIÓN MULTINOMIAL CON SUBCLASIFICACIÓN

En esta clasificación multinomial se utilizan los resultados de la clasificación binaria que fueron etiquetados como clase 1 (hacer algo), las agrupaciones y etiquetas expuestas en la Tabla 2 y la técnica *undersampling* para mejorar el balance de los datos.

Para esta clasificación también se divide el conjunto de datos en entrenamiento y evaluación con la misma distribución utilizada anteriormente. La transformación del vector de etiquetas con one-hot encoding genera esta vez una matriz de 4 columnas como se lo puede ver en la Tabla 3.

La topología RBF fue seleccionada para la creación del modelo de red neuronal que tiene 5 capas: la primera capa, de entrada, que tiene 3 neuronas; una segunda con 10 neuronas y funciones de activación de ReLu; una tercera contiene 20 neuronas y la función de activación sigmoide; la cuarta con 12 neuronas y ReLu en función de activación; y, por último, la quinta capa, de salida, con 5 neuronas y con softmax como función de activación. Para esta ocasión el entrenamiento tiene 5000 épocas, se utiliza entropía cruzada categórica, sensibilidad y especificidad como métricas de precisión. (Meza, Zambrano, & Ortega, 2021)

CAPÍTULO 3: DESARROLLO DE LA APLICACIÓN

3.1 ANÁLISIS DE REQUERIMIENTOS

3.1.1 ALCANCE

La aplicación estará constituida por 4 módulos: Módulo de Ingreso de Datos, Módulo de Recomendaciones, Módulo de Reportes y Módulo de Administración de Usuarios; y tendrá embebido un modelo de clasificación por series de tiempo, desarrollado en un trabajo previo a éste.

El módulo de Ingreso de Datos permitirá que un usuario registrado en la aplicación con el rol de técnico agrícola ingrese y guarde los datos de la medición de una muestra de una parcela de cultivo en una base de datos. Los datos a ingresar son: pH, densidad y conductividad eléctrica de compost, tanque, goteo y programación, fecha de la medición, bloque, cama, número de muestreo, la cantidad de botones de cada estadio y el promedio en centímetros de las medidas de los botones. Las medidas individuales no serán almacenadas.

A continuación, se describe la funcionalidad del módulo de Recomendaciones. Como ya se mencionó anteriormente, la aplicación se servirá de un modelo de clasificación para poder brindar una recomendación automática de la actividad agrícola al usuario, utilizando los datos de pH, densidad y conductividad eléctrica de suelo y la mostrará al usuario en pantalla. Adicionalmente el usuario tendrá la opción de calificar esta recomendación automática en un formulario respondiendo dos preguntas: “¿Está de acuerdo con esta recomendación?” y “¿Qué recomendación daría usted?”, en esta última pregunta el usuario tendrá a disposición todas las recomendaciones posibles que brinda el modelo de clasificación. Un futuro trabajo puede crear un sistema que tome esta nueva información para reentrenar el modelo.

Con el módulo de Reportes el usuario podrá imprimir reportes en archivos de Excel de las mediciones guardadas. También podrá buscar recomendaciones obtenidas previamente, en un rango de fechas, e imprimir un reporte en formato PDF con el resultado de su consulta.

El módulo de administración de usuarios estará destinado a la gestión de usuarios. Un usuario con rol de administrador podrá crear, modificar, consultar y eliminar a usuarios de la aplicación. Por motivos de seguridad, cada usuario podrá recuperar, mediante su correo electrónico, su contraseña en caso de haberla perdido u olvidado.

El aplicativo se desarrollará en lenguaje Python versión 3.7 con el framework Flask versión 2.1.1, el cual está diseñado para el desarrollo de aplicaciones web. En la fase de desarrollo se utilizará SQLAlchemy ORM, mientras que para la fase de implementación se utilizará MySQL. El modelo de clasificación está construido con la librería Keras versión 2.3.1 y su código está alojado en un repositorio en GitHub.

3.1.2 USUARIOS DE LA APLICACIÓN

Los usuarios que tendrán acceso a la aplicación son:

Técnico Agrícola: será el usuario que ingresará los datos de las mediciones diarias y que recibirá de la aplicación una recomendación de actividades agrícolas cada vez que ingrese nuevas mediciones.

Administrador: este usuario tendrá todos los permisos del usuario Técnico Agrícola y además estará encargado de administrar los usuarios que acceden a la aplicación. Puede crear, modificar, consultar y eliminar usuarios.

3.1.3 HISTORIAS DE USUARIO

Las historias de usuario son descripciones breves, pero bien formadas de alguna característica o funcionalidad nueva contadas desde la perspectiva de la persona que la necesita (Flewelling, 2018). Se utilizó historias de usuario para describir los requerimientos de la aplicación. Éstas fueron elaboradas con el técnico agrícola de la florícola de la cual “Agroscan” obtuvo los datos de las mediciones. La Tabla 4 muestra el formato utilizado para las historias de usuario. Los campos a llenarse en estas tablas serán: Número, que será un identificador de la historia de usuario; Título; Descripción, que incluirá un detalle de la historia de usuario; Criterios de

aceptación, es lo que el Product Owner busca en una historia para considerar que está terminada; está compuesta de los aspectos de funcionalidad que el Product Owner espera ver y probar para verificar que la historia funcione como se espera (Flewelling, 2018).

Tabla 4

Formato de historia de usuario

Número:
Título:
Descripción:
Criterios de aceptación:

Elaborado por Misael Cabascango y Diego Osorio.

En la Tabla 5 se puede apreciar la historia de usuario Ingresar Datos de una Medición, que concierne a los dos tipos de usuario del aplicativo.

Tabla 5

Historia de usuario Ingresar Datos de una Medición

Número: 1
Título: Ingresar Datos de una Medición.
Descripción: Como usuario deseo guardar los datos de la medición que se realiza a diario para tener un registro de mediciones.
Criterios de aceptación: Los datos deben ser validados antes de guardarlos. Pueden existir campos o variables para los cuales no se haya realizado una medición. En ese caso se guardan como NaN. No se debe ingresar una fecha superior a la fecha del día en que se realice la medición.

No se puede ingresar más de una medición el mismo día.
--

Elaborado por Misael Cabascango y Diego Osorio.

En la Tabla 6 se puede apreciar la historia de usuario Visualizar Recomendación.

Tabla 6

Historia de usuario Visualizar Recomendación

Número: 2
Título: Visualizar Recomendación.
Descripción: Como usuario deseo visualizar una recomendación de una actividad agrícola.
Criterios de aceptación: Ninguno.

Elaborado por Misael Cabascango y Diego Osorio.

En la Tabla 7 se puede apreciar la historia de usuario Calificar Recomendación.

Tabla 7

Historia de usuario Calificar Recomendación

Número: 3
Título: Calificar Recomendación.
Descripción: Como usuario deseo calificar y guardar la recomendación dada por el aplicativo para tener un historial que se pueda consultar e imprimir un reporte del mismo.
Criterios de aceptación: Si no se está de acuerdo con la recomendación, se debe poder escoger otra opción de todas las recomendaciones posibles que pueda brindar la aplicación. La recomendación se guardará con la fecha en que se hizo la medición.

Elaborado por Misael Cabascango y Diego Osorio.

En la Tabla 8 se puede apreciar la historia de usuario Reporte de Recomendaciones.

Tabla 8*Historia de usuario Reporte de Recomendaciones*

Número: 4
Título: Reporte de Recomendaciones.
Descripción: Como usuario deseo buscar una recomendación (o recomendaciones) en un rango de fechas e imprimir un reporte PDF para tener informes físicos de las recomendaciones que ha brindado la aplicación.
Criterios de aceptación: Se debe validar el rango de fechas. Los datos mostrados en el archivo deben estar acordes a los filtros seleccionados.

Elaborado por Misael Cabascango y Diego Osorio.

En la Tabla 9 se puede apreciar la historia de usuario Reporte de Mediciones.

Tabla 9*Historia de usuario Reporte de Mediciones*

Número: 5
Título: Reporte de Mediciones.
Descripción: Como usuario deseo imprimir en un reporte tipo archivo de Excel las mediciones filtradas en un rango de número de bloque, de número de cama, de número de muestreo y de fechas; y también por variedad, para poder manipular los datos.
Criterios de aceptación: Se debe poder escoger cualquiera de los filtros mencionados. Los datos mostrados en el archivo deben estar acordes a los filtros seleccionados. Se debe mostrar todos los datos de la medición y también los datos de lote, cama, y variedad.

Elaborado por Misael Cabascango y Diego Osorio.

En la Tabla 10 se puede apreciar la historia de usuario Iniciar Sesión.

Tabla 10

Historia de usuario Iniciar Sesión

Número: 6
Título: Iniciar Sesión.
Descripción: Como usuario deseo iniciar sesión en la aplicación con credenciales personales (correo personal y contraseña) para acceder a las funcionalidades que ofrece el aplicativo.
Criterios de aceptación: Se debe acceder a la aplicación con el correo y contraseña proporcionados al Administrador para el registro. Se debe poder cerrar sesión en el aplicativo.

Elaborado por Misael Cabascango y Diego Osorio.

En la Tabla 11 se puede apreciar la historia de usuario Recuperar Contraseña.

Tabla 11

Historia de usuario Recuperar Contraseña

Número: 7
Título: Recuperar Contraseña.
Descripción: Como usuario deseo tener un método para poder reestablecer mi contraseña en caso de pérdida u olvido de la misma
Criterios de aceptación: El usuario debe recibir un correo enviado desde el aplicativo con un link para realizar el proceso de recuperación de la contraseña

Una vez reestablecida la contraseña, el usuario debe poder acceder a la aplicación sin ningún problema.

Elaborado por Misael Cabascango y Diego Osorio.

Las siguientes historias de usuario pertenecen exclusivamente al usuario administrador. En la Tabla 12 se puede apreciar la historia de usuario Crear Usuario.

Tabla 12

Historia de usuario Crear Usuario

Número: 8
Título: Crear Usuario.
Descripción: Como usuario administrador deseo poder crear nuevos usuarios con un determinado rol para que puedan acceder a la aplicación.
Criterios de aceptación: Se debe poder ingresar el nombre, apellido, correo electrónico, contraseña y rol del usuario a crearse. No se debe poder crear usuarios duplicados. El criterio de validación será el correo electrónico.

Elaborado por Misael Cabascango y Diego Osorio.

En la Tabla 13 se puede apreciar la Historia de usuario Modificar Usuario.

Tabla 13

Historia de usuario Modificar Usuario

Número: 9
Título: Modificar Usuario.
Descripción: Como usuario administrador deseo modificar los datos de registro de cualquier usuario registrado con el fin de que éste pueda actualizar sus datos si así lo requiere.

Criterios de aceptación:

No se debe poder modificar la contraseña de ningún usuario.

Una vez modificados los datos de un usuario, éste debe poder iniciar sesión sin ningún problema.

Elaborado por Misael Cabascango y Diego Osorio.

En la Tabla 14 se puede apreciar la historia de usuario Visualizar Usuarios.

Tabla 14

Historia de usuario Visualizar Usuarios

Número: 10
Título: Visualizar Usuarios.
Descripción: Como usuario administrador deseo visualizar todos los usuarios registrados en la aplicación y los datos de registro de un solo usuario para realizar tareas de gestión (crear, modificar, o eliminar usuarios).
Criterios de aceptación: Se deben mostrar todos los usuarios registrados en la aplicación

Elaborado por Misael Cabascango y Diego Osorio.

En la Tabla 15 se puede apreciar la historia de usuario Eliminar Usuario.

Tabla 15

Historia de usuario Eliminar Usuario

Número: 11
Título: Eliminar Usuario.
Descripción: Como usuario administrador deseo eliminar uno o varios usuarios registrados en la aplicación para descartar usuarios que por varias razones ya no deban iniciar sesión.
Criterios de aceptación:

Una vez eliminado un usuario del registro, éste no debe poder más iniciar sesión en el aplicativo.

No se puede eliminar todos los usuarios de la aplicación, debe quedar al menos un usuario administrador.

Elaborado por Misael Cabascango y Diego Osorio.

3.3.3 PRODUCT BACKLOG

El product backlog fue definido por las historias de usuario descritas anteriormente. Cada historia de usuario está puntuada con una medida subjetiva denominada puntos de historia. Para puntuar cada historia de usuario se tomó como pivote la historia de usuario Iniciar Sesión, a la cual se la puntuó con 10 puntos de historia. La puntuación de las demás historias de usuario se realizó mediante estimaciones hechas por parte del equipo.

En la Figura 2 se puede apreciar la creación del product backlog con las historias de usuario y la puntuación de cada una, en Jira.

Figura 2

Product Backlog

Proyectos / Asistente Florícola

Backlog

85 0 0 Crear sprint

ID	Descripción	Puntos	Estado	Asignado a
AF-17	Ingresar datos de una medición	10	TAREAS POR HACER	
AF-18	Visualizar recomendación	15	TAREAS POR HACER	
AF-19	Calificar recomendación	5	TAREAS POR HACER	
AF-20	Reporte de recomendaciones	5	TAREAS POR HACER	
AF-21	Reporte de mediciones	5	TAREAS POR HACER	
AF-22	Iniciar sesión	10	TAREAS POR HACER	
AF-23	Recuperar contraseña	5	TAREAS POR HACER	
AF-24	Crear usuario	10	TAREAS POR HACER	

Estás en un proyecto gestionado por el equipo. Más información

Quickstart

Elaborado por Misael Cabascango y Diego Osorio.

3.3.4 PLANIFICACIÓN DE LOS SPRINTS

En la primera reunión se definió los roles del equipo Scrum: Desarrolladores, Scrum Master y Product Owner. El Product Owner fue un técnico agrícola. Por el tamaño del equipo se decidió que una persona desempeñe 2 roles a la vez, sin embargo, en equipos más grandes cada persona tiene un único rol dentro del equipo Scrum.

Para cada sprint el equipo estableció un límite de cumplimiento de 15 puntos de historia. El proyecto se completó en 6 sprints con un tiempo de duración de 2 semanas cada uno.

Cada sprint backlog posee un tablero en el que se realiza el seguimiento de las tareas a realizarse, llamado Scrum board. Este tablero se conforma de varias secciones en las que se van ubicando las historias de usuario según el avance del sprint.

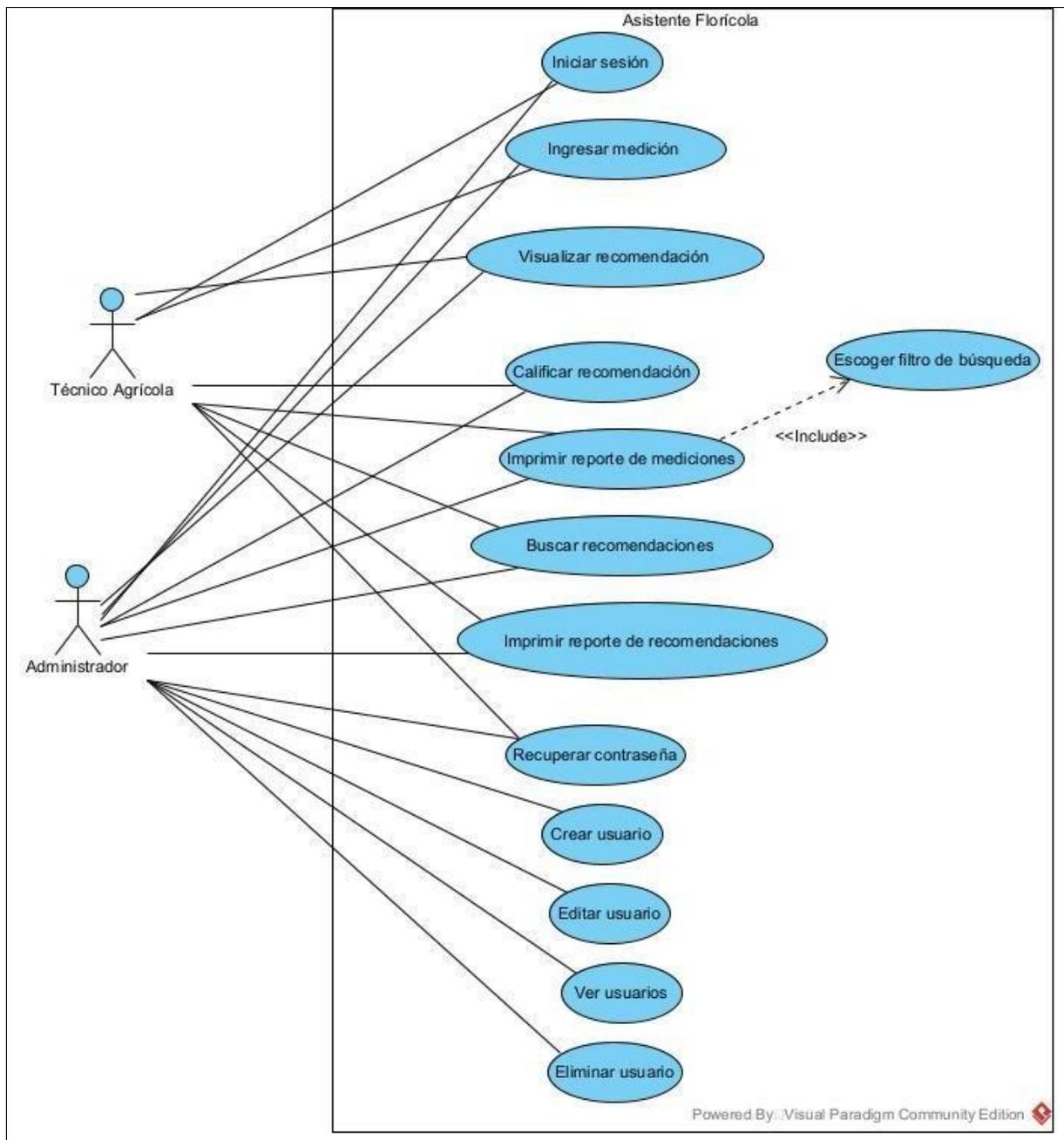
3.2 DISEÑO

3.2.1 CASOS DE USO

A través de los requerimientos de la aplicación, los cuales se encuentran descritos en forma de historias de usuario, se realizó el diagrama de casos de uso. Para entender cada caso de uso a detalle se colocaron a continuación descripciones de casos de uso por cada módulo de la aplicación. En la Figura 3 se muestra el diagrama de casos de uso completo de la aplicación.

Figura 3

Diagrama de caso de uso de la aplicación Asistente Florícola



Elaborado por Misael Cabascango y Diego Osorio.

A continuación, se presenta por módulos la descripción de los casos de uso más relevantes. Cada tabla contiene los siguientes campos: Nombre del caso de uso, Actores, Descripción, Precondiciones, Pasos realizados (ruta principal), Información para los pasos y Escenarios alternos (si existiesen).

3.2.1.1 Casos de uso módulo de ingreso de datos

En la Tabla 16 se muestra la descripción del caso de uso Ingresar Medición.

Tabla 16

Descripción de caso de uso Ingresar Medición

Nombre del caso de uso:	Ingresar Medición.
Actores:	Técnico Agrícola, Administrador.
Descripción:	El Técnico Agrícola ingresa en la aplicación los siguientes datos obtenidos de la parcela de cultivo: pH, densidad y conductividad eléctrica (planta y suelo), pH, densidad y conductividad eléctrica de compost, pH y conductividad eléctrica de tanque, pH y conductividad eléctrica de goteo, pH y conductividad eléctrica de programación, número de muestreo, cantidad en estado arroz, cantidad en estado arveja, cantidad en estado garbanzo, cantidad en estado lenteja, cantidad en estado pintando color, cantidad en estado rayando color, cantidad en estado color definido y fecha de la medición.
Precondiciones:	El usuario inició sesión en la aplicación con el rol de Técnico Agrícola o de Administrador.
Pasos realizados (ruta principal)	Información para los pasos
1. El usuario accede al formulario de Ingreso de Medición.	
2. El usuario ingresa los datos de la medición en el formulario.	pH, densidad y conductividad eléctrica (planta y suelo), pH, densidad y conductividad eléctrica de compost, pH y conductividad eléctrica de tanque, pH y conductividad eléctrica de goteo, pH y conductividad eléctrica de programación, número de muestreo, cantidad en estado arroz, cantidad en estado arveja, cantidad en estado garbanzo, cantidad en estado lenteja, cantidad en estado pintando color, cantidad en estado rayando color, cantidad en estado color definido y fecha de la medición.
3. La aplicación valida los datos ingresados.	

4. El usuario guarda los datos ingresados en la aplicación.	
5. La aplicación informa al usuario que los datos han sido ingresados con éxito.	
Escenarios Alternos Paso 2: Si la fecha de la toma de datos coincide con la fecha actual, no es necesario ingresarla, porque estará ingresada por defecto.	

Elaborado por: Misael Cabascango y Diego Osorio

3.2.1.2 Casos de uso módulo de recomendaciones

En la Tabla 17 se muestra la descripción del caso de uso Visualizar Recomendación.

Tabla 17

Descripción de caso de uso Visualizar Recomendación

Nombre del caso de uso:	Visualizar Recomendación.
Actores:	Técnico Agrícola, Administrador.
Descripción:	El usuario visualiza una recomendación de una actividad agrícola para su parcela de cultivo, brindada por la aplicación.
Precondiciones:	El usuario guardó los datos de la medición, desde la aplicación. El modelo de clasificación ha sido entrenado con datos de anteriores mediciones.
Pasos realizados (ruta principal)	Información para los pasos
1. Los datos de la medición son enviados al modelo de clasificación.	
2. El modelo de clasificación utiliza los datos y genera una recomendación.	
3. La recomendación dada por el modelo se muestra al usuario.	

Elaborado por Misael Cabascango y Diego Osorio.

En la Tabla 18 se muestra la descripción del caso de uso Retroalimentar Recomendación.

Tabla 18

Descripción de caso de uso Retroalimentar Recomendación

Nombre del caso de uso:	Retroalimentar Recomendación.
Actores:	Técnico Agrícola, Administrador.
Descripción:	El usuario retroalimenta al modelo de clasificación informando si está o no de

	acuerdo con la recomendación dada por el sistema. En caso negativo escoge la recomendación que él considera adecuada.
Precondiciones:	Se muestra una recomendación y un formulario para que el usuario pueda ingresar los parámetros de retroalimentación.
Pasos realizados (ruta principal)	Información para los pasos
1. Se muestra en una interfaz la recomendación de una actividad agrícola.	
2. El usuario escoge la opción Sí en la pregunta: ¿Está de acuerdo con la recomendación?	
3. La información del formulario se guarda en la base de datos	Respuesta a la pregunta ¿Está de acuerdo con la recomendación?, recomendación a criterio del usuario.
4. La aplicación informa al usuario que los datos han sido ingresados con éxito.	
Escenarios Alternos	
Paso 2: Si el usuario no está de acuerdo con la opinión, se muestra un menú desplegable donde deberá escoger una opción, a su criterio	

Elaborado por Misael Cabascango y Diego Osorio.

3.2.1.3 Módulo de Reportes

En la Tabla 19 se muestra la descripción del caso de uso Consultar Recomendaciones.

Tabla 19

Descripción de caso de uso Consultar Recomendaciones

Nombre de caso de uso:	Consultar Recomendaciones.
Actores:	Técnico Agrícola, Administrador.
Descripción:	El usuario ingresa una fecha de inicio y una fecha de fin y obtiene las recomendaciones generadas en ese lapso de tiempo.
Precondiciones:	El historial de recomendaciones se muestra correctamente en la aplicación.
Pasos realizados (ruta principal)	Información para los pasos
1. El usuario ingresa un rango de fechas para la búsqueda.	Fecha de inicio, Fecha de fin.

2. Los parámetros de búsqueda son validados por la aplicación.	
3. Se realiza la búsqueda en la base de datos.	
4. Se muestra los resultados de la búsqueda en la aplicación, en una tabla.	
5. El usuario puede imprimir un reporte PDF con la información de la búsqueda si desea.	

Elaborado por Misael Cabascango y Diego Osorio.

En la Tabla 20 se muestra la descripción del caso de uso Consultar Recomendaciones.

Tabla 20

Descripción de caso de uso Consultar Recomendaciones

Nombre de caso de uso:	Consultar Recomendaciones.
Actores:	Técnico Agrícola, Administrador.
Descripción:	El usuario ingresa una fecha de inicio y una fecha de fin y obtiene las recomendaciones generadas en ese lapso de tiempo.
Precondiciones:	El historial de recomendaciones se muestra correctamente en la aplicación.
Pasos realizados (ruta principal)	Información para los pasos
1. El usuario ingresa un rango de fechas para la búsqueda.	Fecha de inicio, fecha de fin.
2. Los parámetros de búsqueda son validados por la aplicación.	
3. Se realiza la búsqueda en la base de datos.	
4. Se muestra los resultados de la búsqueda en la aplicación, en una tabla.	
5. El usuario puede imprimir un reporte PDF con la información de la búsqueda si desea.	

Elaborado por Misael Cabascango y Diego Osorio.

En la Tabla 21 se muestra la descripción del caso de uso Imprimir Reporte de Mediciones.

Tabla 21

Descripción de caso de uso Imprimir Reporte de Mediciones

Nombre de caso de uso:	Imprimir Reporte de Mediciones.
Actores:	Técnico Agrícola, Administrador.
Descripción:	El usuario selecciona un filtro de búsqueda y obtiene un reporte de Excel de las mediciones encontradas.
Precondiciones:	Ingresar al menos una medición.
Pasos realizados (ruta principal)	Información para los pasos
1. El Técnico Agrícola selecciona un filtro de búsqueda.	
2. El usuario ingresa los parámetros solicitados para la búsqueda.	Número de inicio de rango, número de fin de rango, variedad(es), fecha de inicio, fecha de fin.
3. Los parámetros de búsqueda son validados por la aplicación.	
4. Se realiza la búsqueda en la base de datos.	
5. La aplicación genera un reporte en Excel para el usuario con la información de la búsqueda.	

Elaborado por Misael Cabascango y Diego Osorio.

3.2.1.4 Módulo de gestión de usuarios

En la Tabla 22 se muestra la descripción del caso de uso Crear Usuario.

Tabla 22

Descripción de caso de uso Crear Usuario

Nombre de caso de uso:	Crear Usuario.
Actores:	Administrador.
Descripción:	El Administrador registra a un usuario nuevo en la aplicación.
Precondiciones:	El usuario inició sesión en la aplicación con el rol de Administrador. El usuario a crear no se encuentra registrado en la aplicación.
Pasos realizados (ruta principal)	Información para los pasos
1. El Administrador se dirige al formulario de creación de nuevo usuario.	

2. El Administrador ingresa los datos del nuevo usuario.	Nombre, apellido, correo electrónico, contraseña, rol de usuario.
3. La aplicación valida los datos ingresados.	
4. El Administrador guarda los datos del nuevo usuario, en la aplicación.	
5. La aplicación informa al Administrador que el usuario ha sido creado con éxito.	

Elaborado por Misael Cabascango y Diego Osorio.

En la Tabla 23 se muestra la descripción del caso de uso Editar Usuario.

Tabla 23

Descripción de caso de uso Editar Usuario

Nombre de caso de uso:	Editar Usuario.
Actores:	Administrador.
Descripción:	El Administrador modifica los datos de un usuario que ya se encuentre registrado en aplicación.
Precondiciones:	El Administrador accedió al formulario del usuario que desea editar.
Pasos realizados (ruta principal)	Información para los pasos
1. El Administrador selecciona al usuario del que desea editar sus datos.	
2. El Administrador ingresa los datos que necesite editar.	Nombre, apellido, contraseña, correo electrónico, rol(es) de usuario.
3. La aplicación valida los datos ingresados.	
4. El Administrador guarda los datos editados, en la aplicación.	
5. La aplicación informa al Administrador que el usuario ha sido editado con éxito.	

Elaborado por Misael Cabascango y Diego Osorio.

En la Tabla 24 se muestra la descripción del caso de uso Eliminar Usuario.

Tabla 24

Descripción de caso de uso Eliminar Usuario

Nombre de caso de uso:	Eliminar Usuario.
Actores:	Administrador.

Descripción:	El Administrador elimina de la aplicación a un usuario que ya se encuentre registrado.
Precondiciones:	El usuario localizó en el registro de usuarios al usuario que desea eliminar.
Pasos realizados (ruta principal)	Información para los pasos
1. El Administrador selecciona al usuario que desea eliminar.	
2. El Administrador elimina al usuario.	
3. La aplicación pide la confirmación del Administrador para eliminar al usuario seleccionado.	
4. El Administrador confirma la eliminación del usuario.	
5. La aplicación informa al Administrador que el usuario ha sido eliminado con éxito.	

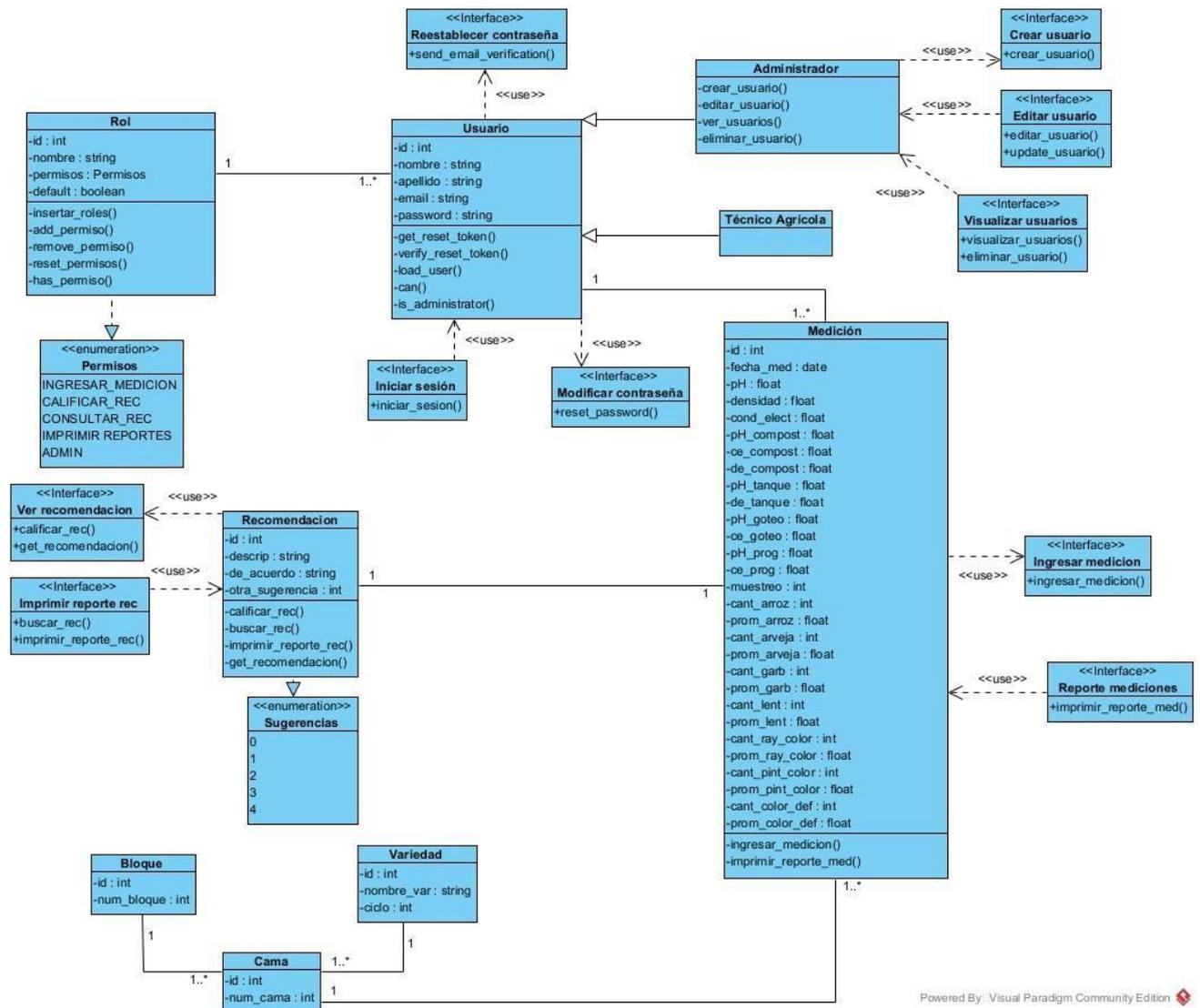
Elaborado por Misael Cabascango y Diego Osorio.

3.2.2 DIAGRAMA DE CLASES

En la Figura 4 se puede apreciar el diagrama de clases de la aplicación. Se consideró utilizar las siguientes clases: Usuario, Rol, Lote, Cama, Variedad, Medición y Recomendación, en las cuales se puede apreciar sus atributos, métodos y relaciones existentes entre cada clase.

Figura 4

Diagrama de clases de la aplicación



Elaborado por Misael Cabascango y Diego Osorio.

3.2.3 DIAGRAMA CONCEPTUAL DE BASE

El diagrama conceptual de la base de datos se presenta en la Figura 5. Las tablas que forman parte del diagrama se describen a continuación:

Usuario. Almacenará toda la información de los usuarios de la aplicación. Está formada por los campos: id, nombre, apellido, email y password. Los 2 últimos son obligatorios. El campo email es un campo único. Tiene una relación de uno a varios con la tabla “Medición”.

Rol: Almacenará la información de los roles de un usuario en la aplicación. Contiene los campos: id, nombre y permisos. El campo nombre es obligatorio y único. El campo permisos es de tipo entero y define la lista de permisos para el rol. Para ello se sirve de la clase de enumeración permisos, presente en el diagrama de clases. Tiene una relación de uno a varios con la tabla “Usuario”.

Variedad: Almacenará la información de las variedades de rosas. Sus campos son: id, nombre_var un campo obligatorio y ciclo, un campo no obligatorio. Tiene una relación de uno a varios con la tabla “Cama”.

Bloque: Almacenará la información de los bloques o lotes de una florícola. El campo num_bloque es un campo entero y obligatorio. Tiene una relación de uno a varios con la tabla “Cama”.

Cama: Almacenará la información de las camas de los diferentes bloques. Está formada por los campos: id, num_cama, las llaves foráneas procedentes de las tablas “Variedad” y “Bloque”.

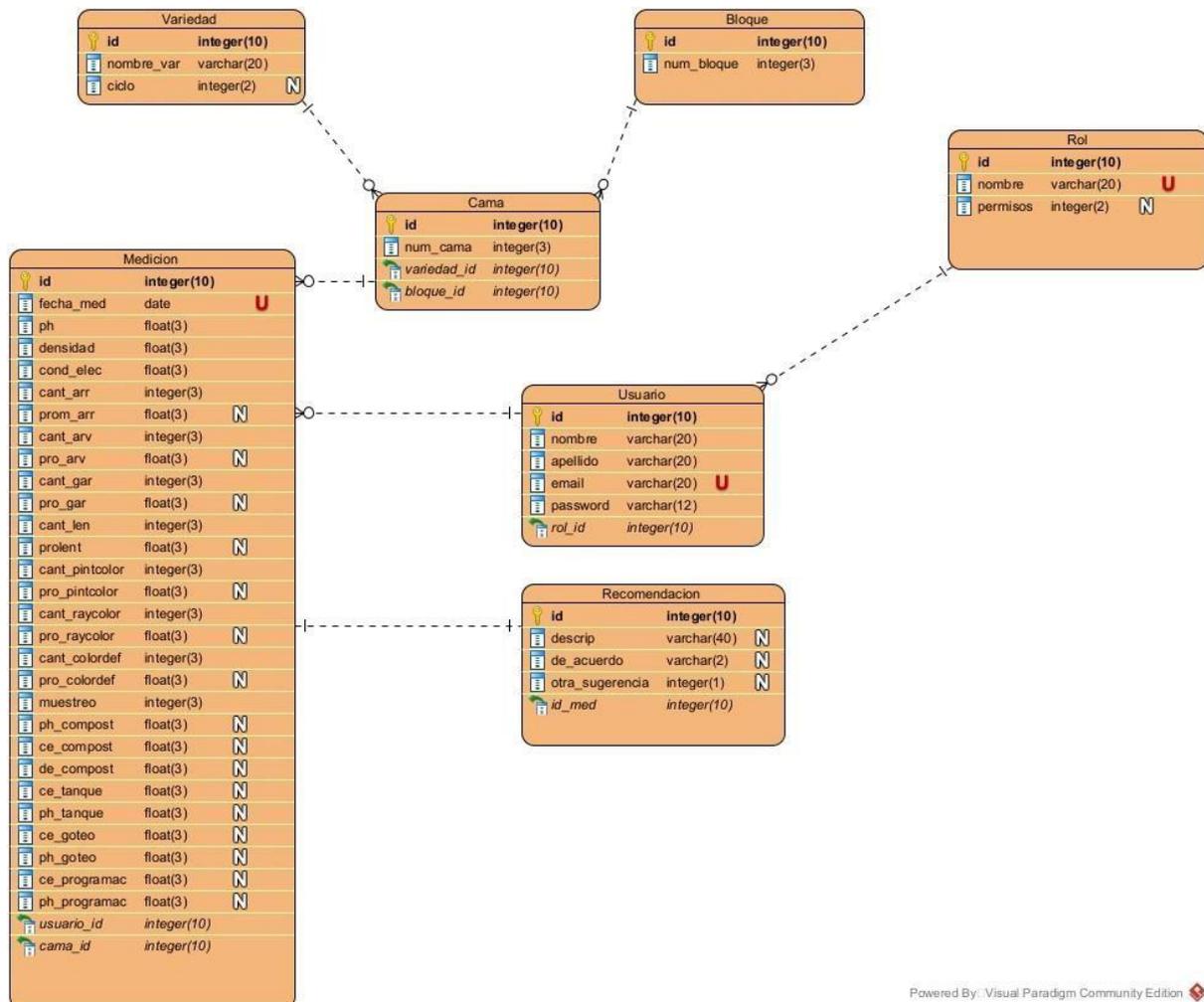
Medición: Almacenará la información de las mediciones realizadas a diario. Los campos fecha_med, ph, densidad, cond_elec, cant_arr, cant_arv, cant_garb, cant_len, cant_pintcolor, cant_raycolor, cant_colordef y muestreo son campos obligatorios y el resto son campos opcionales. La tabla también contiene las llaves foráneas de la tabla “Usuario” y “Cama”. Tiene una relación de uno a uno con la tabla “Recomendación”.

Recomendación: Almacenará la información de las recomendaciones obtenidas a partir de las mediciones ingresadas. Está compuesta por los campos: id, descrip, que es un campo en el que se guardará exclusivamente la recomendación arrojada por el modelo de clasificación, de_acuerdo, que es un campo obligatorio, y el campo otra_sugerencia que es un campo entero que guardará un valor numérico que se corresponde con todas las recomendaciones posibles

que puede ofrecer el modelo de clasificación y que se detallan a través de la clase de enumeración sugerencias, del diagrama de clases.

Figura 5

Diagrama conceptual de base de datos



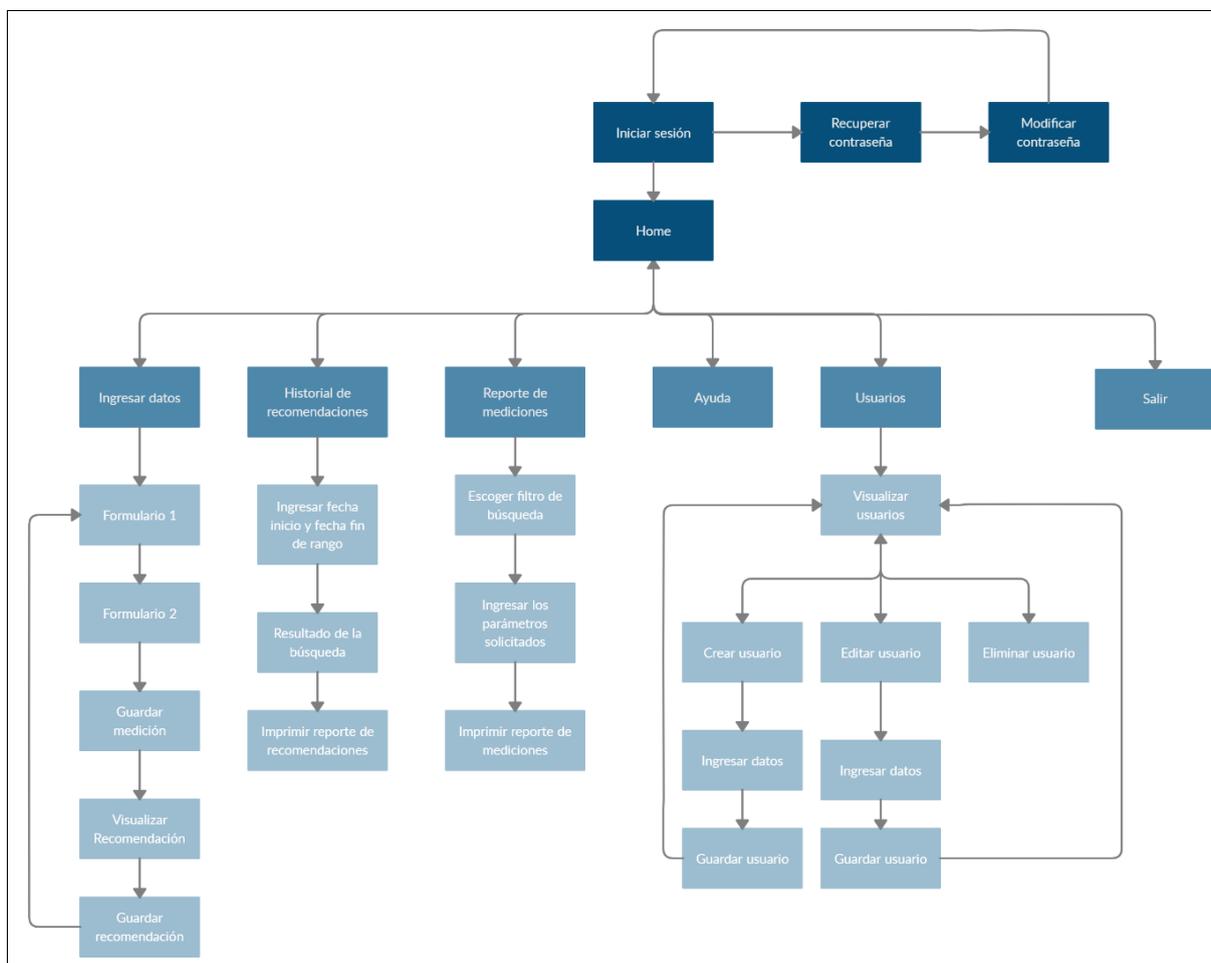
Elaborado por Misael Cabascango y Diego Osorio.

3.2.4 DIAGRAMA NAVEGACIONAL

En la Figura 6 se muestra el diagrama de navegación de la aplicación, en el cual se puede apreciar el flujo de navegación tanto de un usuario Administrador como de un usuario con rol de Técnico Agrícola. El usuario Administrador tiene acceso al menú Usuario mientras que el usuario Técnico Agrícola no. Cada nodo representa una página o un proceso que el usuario realiza.

Figura 6

Diagrama navegacional de la aplicación



Elaborado por Misael Cabascango y Diego Osorio.

3.3 DESARROLLO

3.3.1 HERRAMIENTAS DE DESARROLLO

3.3.1.1 Lenguaje Python

Python, es un lenguaje de programación de propósito general muy poderoso y flexible, a la vez que sencillo y fácil de aprender. Es un lenguaje de alto nivel, que permite procesar fácilmente todo tipo de estructuras de datos, tanto numéricos como de texto.

3.3.1.2 Flask

Flask es un “micro” framework escrito en Python y concebido para facilitar el desarrollo de Aplicaciones Web bajo el patrón MVC.

3.3.1.3 Keras

Keras es una biblioteca que funciona a nivel de modelo, proporcionando bloques modulares con los que se pueden desarrollar modelos complejos de aprendizaje (Ionos, 2020). A diferencia de otros frameworks, este software de código abierto no se puede utilizar para operaciones sencillas de bajo nivel, sino que utiliza bibliotecas de los frameworks, que en cierto punto actúan como un motor de backend para Keras.

3.3.1.4 SQLAlchemy

SQLAlchemy es una librería para Python que facilita el acceso a una base de datos relacional, así como las operaciones a realizar sobre la misma.

Es independiente del motor de base de datos a utilizar, es decir, en principio, es compatible con la mayoría de las bases de datos relacionales conocidas: PostgreSQL, MySQL, Oracle, Microsoft SQL Server, SQLite.

3.3.1.5 MySQL

MySQL es el sistema de gestión de bases de datos relacional más extendido en la actualidad al estar basada en código abierto. Desarrollado originalmente por MySQL AB, fue adquirida por Sun Microsystems en 2008 y esta su vez comprada por Oracle Corporation en 2010, la cual ya era dueña de un motor propio InnoDB para MySQL.

MySQL es un sistema de gestión de bases de datos que cuenta con una doble licencia. Por una parte, es de código abierto, pero por otra, cuenta con una versión comercial gestionada por la compañía Oracle. (Robledano, 2019)

3.3.1.6 PyCharm IDE

PyCharm es un entorno de desarrollo integrado (IDE) utilizado en programación de computadoras , específicamente para el lenguaje Python . Está desarrollado por la empresa checa JetBrains. Proporciona análisis de código, un depurador gráfico, un probador de unidades integrado, integración con sistemas de control de versiones (VCSes) y admite el desarrollo web con Django , así como la ciencia de datos con Anaconda . (Wikipedia, 2021)

3.3.1.7 Jira

Para gestionar el proyecto se utilizó el software Jira. Jira es una herramienta utilizada para gestionar proyectos con metodologías ágiles como Scrum y Kanban. Es una herramienta que lleva varios años en el mercado y una de las más populares para gestionar proyectos de software. Es accesible a través de la web desde cualquier navegador.

3.3.2 ARQUITECTURA DE LA APLICACIÓN

La arquitectura seleccionada para la construcción de la solución es la planteada por el modelo MVC que se muestra en la Figura 7 y se describe de la siguiente forma:

3.3.2.1 Modelo

El modelo de la aplicación se encuentra descrito en el archivo models.py. Para facilitar la interacción con la base de datos, se empleó ORM. Un ORM es un modelo de programación que permite mapear las estructuras de una base de datos relacional (SQL Server, Oracle, MySQL, etc.), o también llamados RDBMS (Relational Database Management System), sobre una estructura lógica de entidades y tiene como finalidad simplificar y acelerar el

desarrollo de aplicaciones (Muro, s.f.). El software utilizado para emplear ORM en el desarrollo de esta aplicación fue SQLAlchemy mientras que el RDBMS fue MySQL.

3.3.2.2 Vista

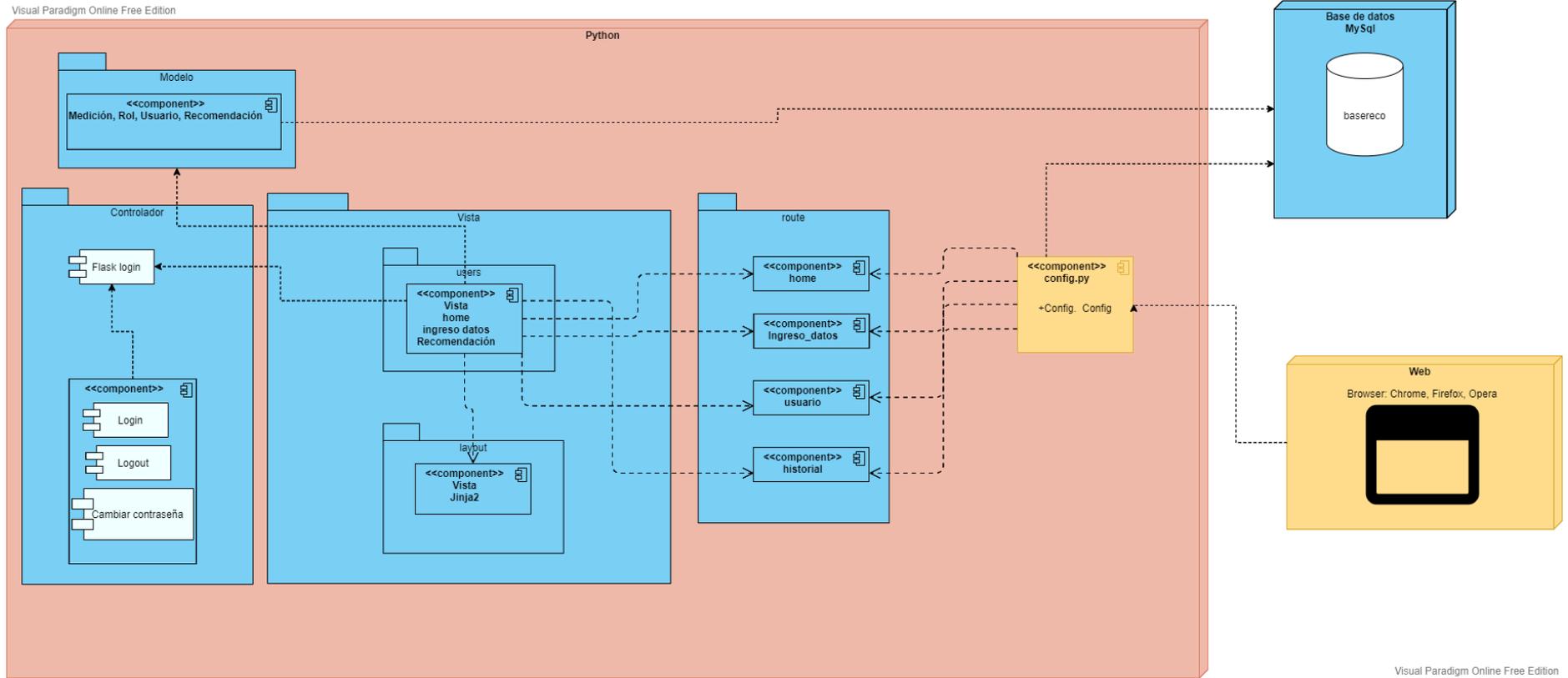
Comprende los archivos ubicados en la carpeta templates. Se optó por el uso del motor de plantillas Jinja2, ya que Flask utiliza este motor para generar documentos HTML válidos de una manera muy sencilla y eficiente.

3.3.2.3 Controlador

Son aquellos bloques encargados de enlazar el modelo con la vista y actúan como un intérprete de datos para que estos puedan ser comprendidos por los componentes como flask -login (inicio de sesión) y routes.py, que tendrá el direccionamiento y las funciones que se van a utilizar

Figura 7

Diagrama de componentes



Elaborado por Misael Cabascango y Diego Osorio

3.3.3 EJECUCIÓN DE LOS SPRINTS

En cada Sprint se mostrará el Sprint backlog resultante. El proceso seguido en cada Sprint fue el siguiente:

- 1) El primer día de cada Sprint se realizó el Sprint planning, en el que se definieron las historias de usuario a completar en el Sprint.
- 2) Cada día se mantenía reuniones virtuales de 10 minutos para llevar un control del progreso del Sprint.
- 3) El día final de cada Sprint se realizó una retroalimentación con el Product Owner, conocida como Sprint review.

3.3.3.1 PRIMER SPRINT

En la Figura 8 se muestra el Scrum board del Sprint 1 donde se muestran las historias de usuario realizadas en este Sprint.

Figura 8

Scrum board para el Sprint 1



Elaborado por Misael Cabascango y Diego Osorio.

Primero se ingresaron datos en las tablas Bloque, Variedad y Cama, a través de la consola de Python y con la sintaxis de SQLAlchemy, para poder llevar a cabo el desarrollo de este Sprint. Debido a la cantidad de campos que tiene una medición, se decidió dejar que el usuario ingrese datos en 2 formularios separados. A continuación, se muestran los resultados del Sprint 1. La

Figura 9 muestra el primer formulario y la Figura 11 el segundo. La historia de usuario para este Sprint se encuentra funcional.

Figura 9

Formulario 1 para ingreso de una medición

Estadio	N° Botones	Medidas Botones	Promedio
Arroz			

Elaborado por Misael Cabascango y Diego Osorio

Figura 10

Formulario 2 para ingreso de una medición

The screenshot shows a mobile application interface for data entry. On the left is a navigation menu with icons and text for 'Ingreso de datos', 'Historial Recomendaciones', 'Reportes Mediciones', 'Ayuda', and 'Salir'. The main content area has a dark green header with the title 'Ingreso de Datos' and a date '08/08/2021'. Below the header, there are three colored panels: a blue panel for 'Sustrato Líquido', an olive green panel for 'Compost', and a grey panel for 'Tanque'. Each panel contains three input fields labeled 'Conductividad Eléctrica', 'pH', and 'Densidad'.

Elaborado por Misael Cabascango y Diego Osorio.

No existen datos de mediciones guardadas previamente, por lo que una vez que el usuario guarde la medición, si el número de registros de la tabla “Medición” es menor o igual a 10, la aplicación redirigirá al usuario al formulario 1 y mostrará un mensaje que se puede apreciar en la Figura 11.

Figura 11*Mensaje al ingresar medición*
Elaborado por Misael Cabascango y Diego Osorio.

3.3.3.2 SEGUNDO SPRINT

En la Figura 12 se muestra el Scrum board del Sprint 2 donde se pueden apreciar las historias de usuario realizadas en este Sprint.

Figura 12*Scrum Board para el Sprint 2**Elaborado por Misael Cabascango y Diego Osorio.*

La recomendación a visualizarse se obtiene mediante los datos de pH, conductividad eléctrica y densidad, enviados desde el formulario 2 de ingreso de medición al modelo de clasificación

una vez existan 11 registros en la tabla “Medición”. El usuario es automáticamente dirigido al formulario que se muestra en la Figura 13. La recomendación no se puede modificar, por razones obvias, pero su calificación se realiza en un próximo Sprint. La historia de usuario para este Sprint se encuentra funcional.

Figura 13

Recomendación obtenida del modelo



Elaborado por Misael Cabascango y Diego Osorio.

3.3.3.3 TERCER SPRINT

En la Figura 14 se muestra el Scrum board del Sprint 3 donde se visualizan las historias de usuario realizadas en este Sprint.

Figura 14

Tablero Sprint 3



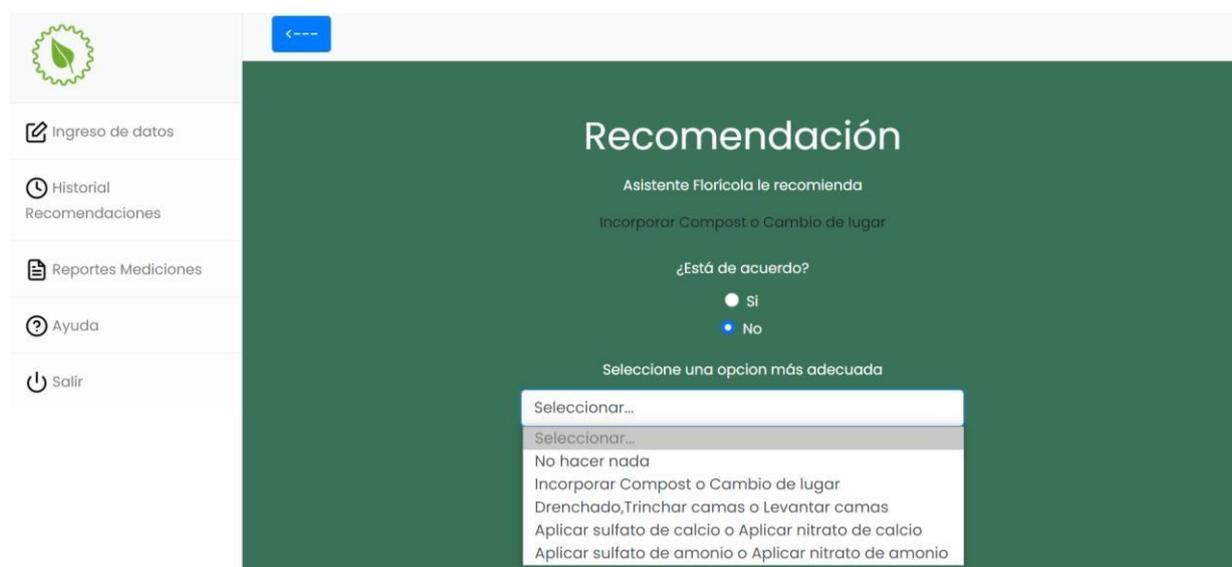
Elaborado por Misael Cabascango y Diego Osorio

Historia de usuario Calificar Recomendación

La recomendación se califica escogiendo si está de acuerdo o no, Si escoge la opción “No”, el combobox siguiente se llenará con las sugerencias posibles que ofrece el modelo, como se puede apreciar en la Figura 15. El campo está de acuerdo, tiene por defecto el valor “No”. Después de llenar el formulario, la recomendación se guarda en la base de datos y el usuario es redirigido al primer formulario para el ingreso de una nueva medición. La historia de usuario Calificar Recomendación se encuentra funcional.

Figura 15

Calificación de una recomendación



Elaborado por Misael Cabascango y Diego Osorio.

Historia de usuario Crear Usuario

La Figura 16 muestra la interfaz para que el usuario Administrador registre un usuario en la aplicación. Para poder registrar un nuevo usuario fue necesario ejecutar desde la consola de Python el método `insert_role()` de la clase `Rol` para asignar los permisos respectivos a cada rol de usuario, luego, mediante un combobox en el formulario, se puede escoger el rol del usuario a registrarse. Un usuario no puede registrarse con el mismo correo varias veces. La contraseña del usuario a registrar debe tener entre 8 y 12 caracteres de longitud. El Administrador es

quien asigna el rol a cualquier usuario. La historia de usuario Crear Usuario se encuentra funcional.

Figura 16

Formulario para la creación de un nuevo usuario



The image shows a web application interface for creating a new user. On the left is a vertical sidebar with a light gray background and a green logo at the top. The sidebar contains the following menu items: 'Ingreso de datos' (with a pencil icon), 'Historial Recomendaciones' (with a clock icon), 'Reportes Mediciones' (with a document icon), 'Ayuda' (with a question mark icon), 'Usuarios' (with a person icon), and 'Salir' (with a power icon). The main content area has a dark green background and is titled 'Nuevo Usuario' in white. Below the title are five white input fields: 'Nombre', 'Apellido', 'Email', 'Contraseña', and 'Rol'. The 'Rol' field is a dropdown menu with 'Seleccionar...' as the placeholder text. At the bottom of the form are two buttons: a green 'Guardar' button and a red 'Cancelar' button.

Elaborado por Misael Cabascango y Diego Osorio

3.3.3.4 CUARTO SPRINT

En la Figura 17 se muestra el Scrum board del Sprint 4 donde se pueden apreciar las historias de usuario realizadas en este Sprint.

Figura 17*Scrum Board para el Sprint 4*

Elaborado por Misael Cabascango y Diego Osorio

El reporte de recomendaciones está constituido por una interfaz donde además de poder imprimir los resultados de las búsquedas de recomendaciones por rango de fechas, se podrá visualizar con anticipación los resultados en una tabla. La interfaz para la búsqueda e impresión del reporte de recomendaciones se puede apreciar en la Figura 18.

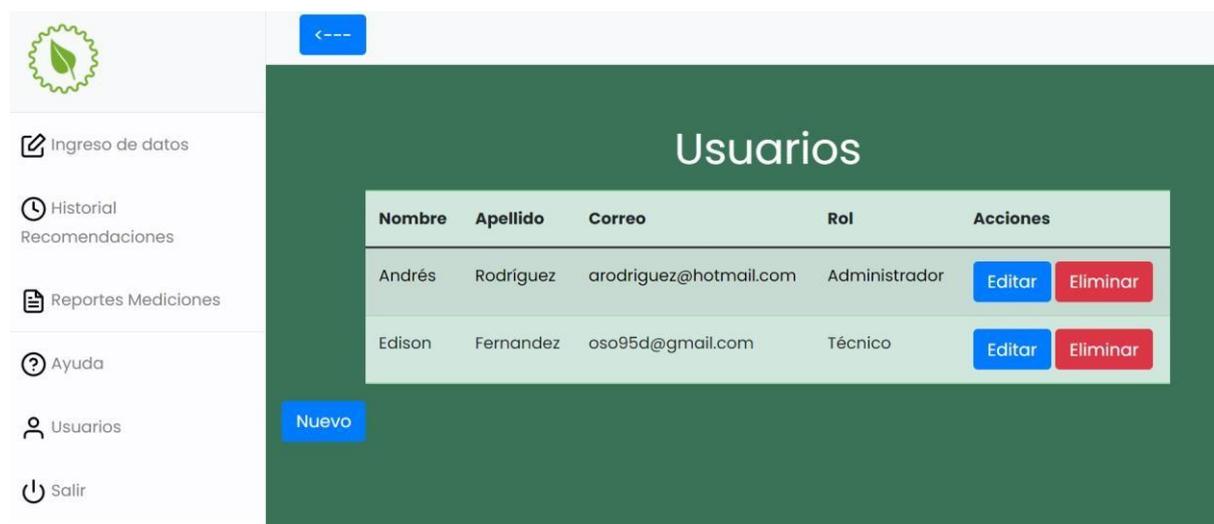
Figura 18*Interfaz para la búsqueda e impresión de reporte de recomendaciones*

Elaborado por Misael Cabascango y Diego Osorio.

El ingreso de fechas se encuentra validado de modo que el usuario no puede ingresar una fecha de inicio mayor a una fecha de fin. Al dar click en el botón Imprimir la aplicación automáticamente empezará a descargar un archivo en formato PDF. El reporte descargado contiene los rangos de fecha que el usuario ingresó y los resultados mostrados en la interfaz. En la Figura 19 se puede apreciar la interfaz donde se visualiza en una tabla todos los usuarios de la aplicación registrados. Desde esta interfaz el administrador puede realizar las operaciones de Editar, Eliminar y Crear un nuevo usuario dando clic en el botón Nuevo que se encuentra en la parte inferior izquierda de la tabla.

Figura 19

Interfaz para visualizar los usuarios de la aplicación



The screenshot shows a web application interface for managing users. On the left is a sidebar menu with icons and labels: 'Ingreso de datos', 'Historial Recomendaciones', 'Reportes Mediciones', 'Ayuda', 'Usuarios', and 'Salir'. The main content area has a dark green header with the title 'Usuarios'. Below the header is a table with the following data:

Nombre	Apellido	Correo	Rol	Acciones
Andrés	Rodríguez	arodriguez@hotmail.com	Administrador	Editar Eliminar
Edison	Fernandez	oso95d@gmail.com	Técnico	Editar Eliminar

At the bottom left of the table area, there is a blue button labeled 'Nuevo'.

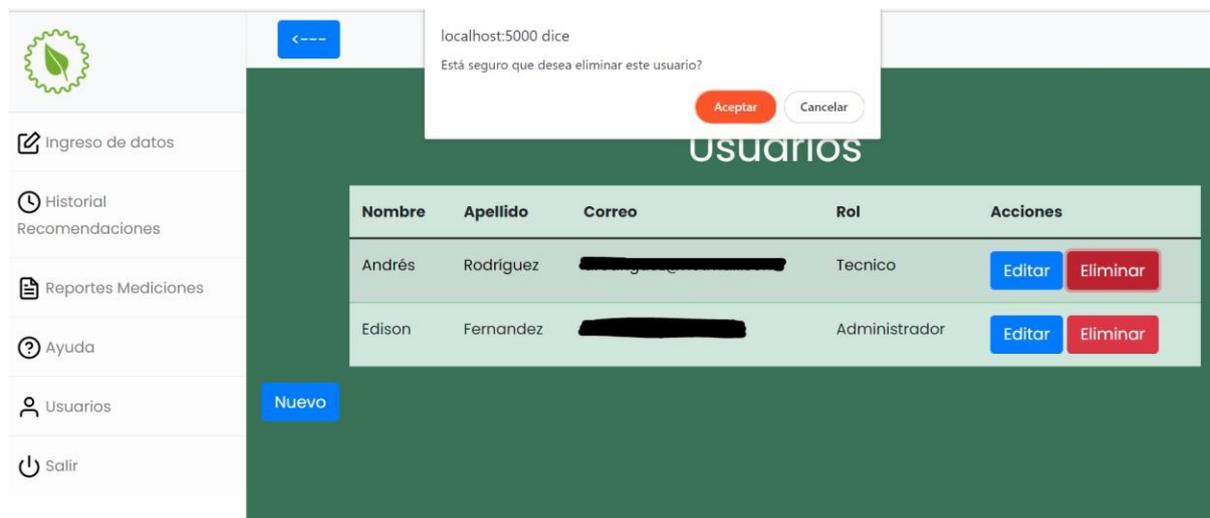
Elaborado por Misael Cabascango y Diego Osorio.

Como ya se mencionó en el párrafo anterior, desde la interfaz donde se visualiza a todos los usuarios se puede eliminar a cualquier usuario que se desee. Al eliminar a un usuario, el Administrador debe tomar en cuenta que la aplicación no permite eliminar todos los usuarios de la aplicación, siempre verificará que quede al menos un usuario Administrador. La

aplicación pide la confirmación para eliminar a un usuario. En la Figura 20 se puede ver dicha solicitud de confirmación.

Figura 20

Eliminación de un usuario



Elaborado por Misael Cabascango y Diego Osorio

Todas las historias de usuario de este Sprint se encuentran funcionales.

3.3.3.5 QUINTO SPRINT

En la Figura 21 se muestra el Scrum board del Sprint 5 donde se muestran las historias de usuario realizadas en este Sprint.

Figura 21

Scrum board para el Sprint 5



Elaborado por Misael Cabascango y Diego Osorio.

El reporte de mediciones se descarga a través de la interfaz mostrada en la Figura 22, en donde el usuario escoge un filtro de búsqueda. Los resultados se mostrarán en un archivo con extensión .xls, de acuerdo al filtro seleccionado por el usuario. Para los 3 primeros filtros: Bloque, Cama y Muestreo se debe ingresar un rango de números. Este ingreso está validado de modo que el usuario no puede ingresar un número de inicio mayor a un número de fin. En el filtro Variedad el usuario puede escoger una o varias variedades de rosas. Debe escoger al menos una para poder imprimir el reporte. Para el filtro de Fecha debe ingresar una fecha de inicio y una fecha de fin. El reporte de mediciones está funcional.

Figura 22

Interfaz para imprimir reporte de mediciones

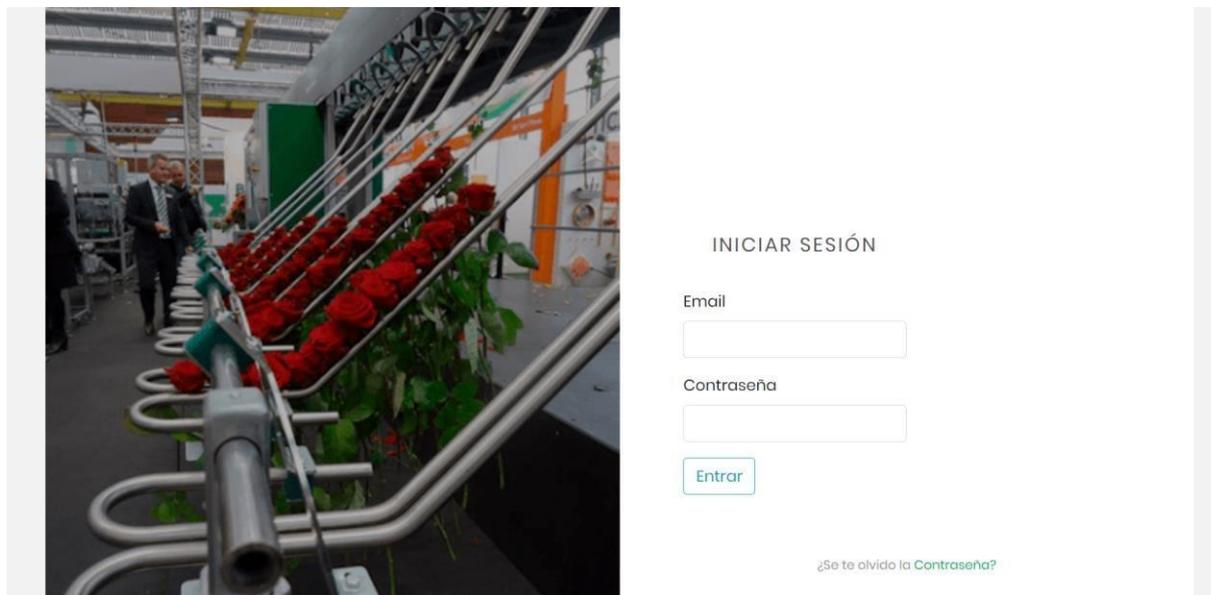


Elaborado por Misael Cabascango y Diego Osorio.

Para que un usuario pueda iniciar sesión en la aplicación debe acceder a la página inicial de la aplicación, la cual se muestra en la Figura 23. Ahí ingresará las respectivas credenciales con las que fue registrado y dando click en el botón Entrar podrá acceder a la página principal de la aplicación o Home. Si el usuario ingresa credenciales incorrectas, la aplicación le dará un aviso. Las sesiones que inicie el usuario tienen un tiempo límite de 5 minutos de inactividad, tras lo cual, si el usuario quiere acceder a cualquier funcionalidad de la aplicación, deberá volver a iniciar sesión.

Figura 23

Interfaz de inicio de sesión



Elaborado por Misael Cabascango y Diego Osorio.

3.3.3.6 SEXTO SPRINT

En la Figura 24 se muestra el Scrum board del Sprint 6 donde se muestran las historias de usuario realizadas en este Sprint.

Figura 24

Scrum board para el Sprint 6

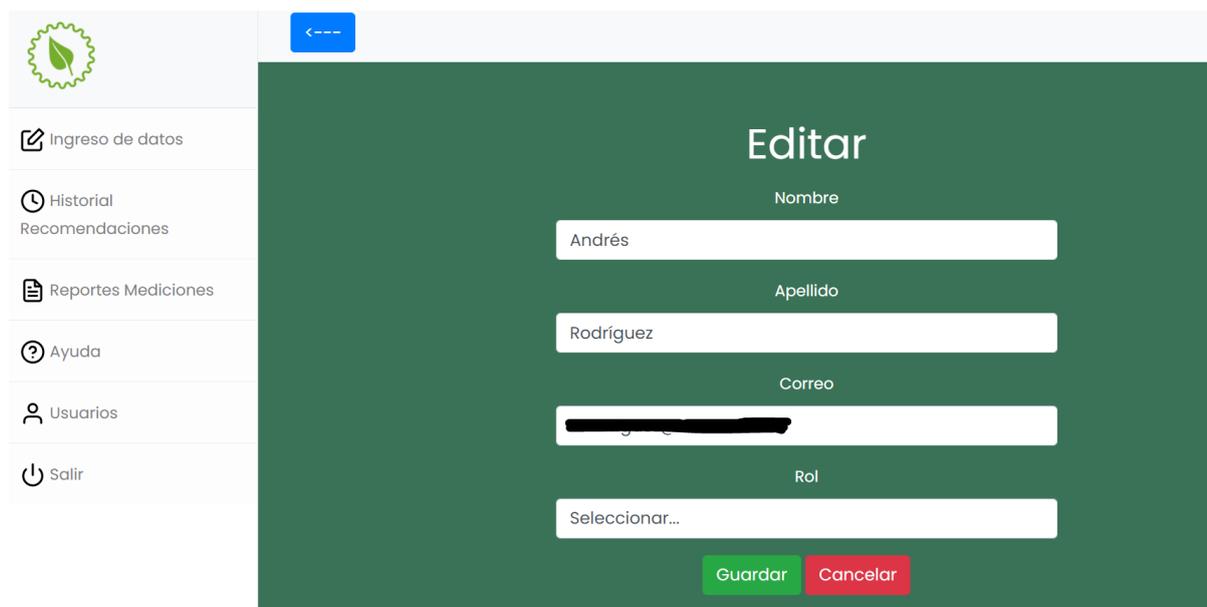


Elaborado por Misael Cabascango y Diego Osorio.

Para modificar un usuario se accede desde la interfaz de visualización de usuarios mediante el botón Editar que se encuentra al lado derecho del rol del usuario. Cuando se da clic se accede al formulario mostrado en la Figura 25.

Figura 25

Formulario para Editar un usuario

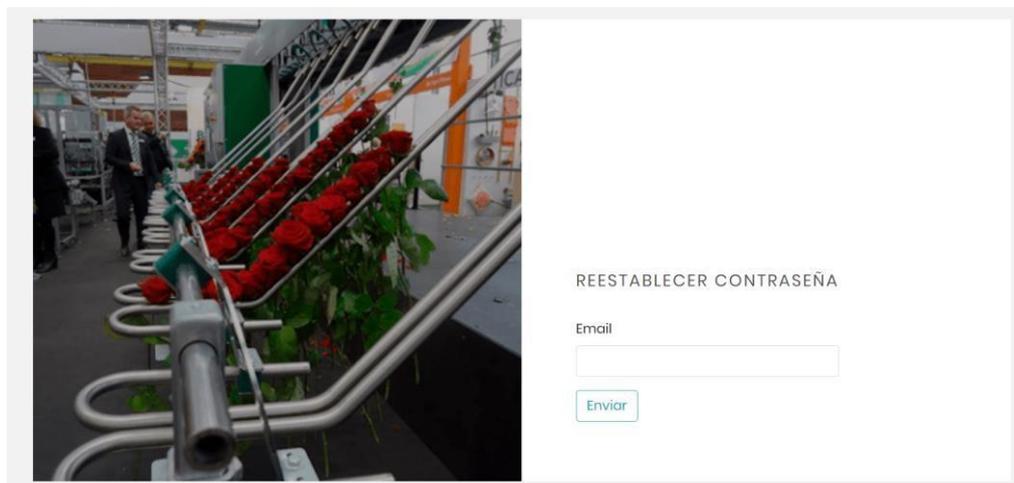


The screenshot shows a web application interface for editing a user. On the left is a sidebar with a green leaf logo and menu items: 'Ingreso de datos', 'Historial Recomendaciones', 'Reportes Mediciones', 'Ayuda', 'Usuarios', and 'Salir'. The main content area has a dark green background and a white header with a blue back arrow and the title 'Editar'. Below the title are four input fields: 'Nombre' with the value 'Andrés', 'Apellido' with 'Rodríguez', 'Correo' which is redacted with a black bar, and 'Rol' with a dropdown menu showing 'Seleccionar...'. At the bottom are two buttons: a green 'Guardar' button and a red 'Cancelar' button.

Elaborado por Misael Cabascango y Diego Osorio.

Por motivos de seguridad, el administrador no puede editar la contraseña de ningún usuario. Al terminar de editar, se da clic en el botón Guardar y la aplicación dirige al administrador a la pantalla de visualización de usuarios, donde puede constatar que efectivamente se realizaron los cambios. La historia de usuario Modificar Usuario está funcional.

Para que un usuario puede recuperar su contraseña en caso de pérdida u olvido deberá acceder desde la pantalla de inicio de sesión a la interfaz de recuperación de contraseña, la cual se muestra en la Figura 26.

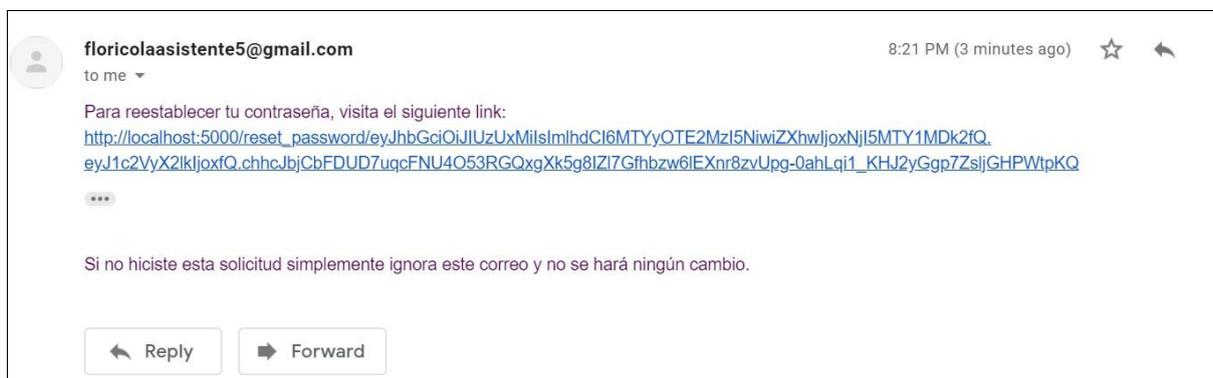
Figura 26*Interfaz de recuperación de contraseña*

Elaborado por Misael Cabascango y Diego Osorio.

En dicha pantalla el usuario ingresa el correo electrónico con el que se encuentra registrado en la aplicación. La aplicación busca si el correo electrónico se encuentra en los registros de la tabla Usuario. Si es así, envía automáticamente un correo electrónico al usuario, dándole instrucciones para establecer una nueva contraseña. El usuario es redirigido a la página de inicio de sesión y se muestra un mensaje indicando que el correo ha sido enviado exitosamente. El correo electrónico que se envía a los usuarios tiene un link con vigencia de 30 minutos, el cual se muestra en la Figura 27.

Figura 27

Correo electrónico enviado desde la aplicación



Elaborado por Misael Cabascango y Diego Osorio.

Al visitar el link el usuario tiene un formulario en el que debe digitar una nueva contraseña 2 veces. Al dar clic en el botón Aceptar, será dirigido a la interfaz de inicio de sesión donde puede iniciar sesión con su nueva contraseña. La historia de usuario Recuperar Contraseña está funcional.

3.4 IMPLEMENTACIÓN

3.4.1 Configuración del servidor en Linode

Para el despliegue de la aplicación se eligió alquilar un servidor dedicado en el sitio web Linode. En la Tabla 24 se puede apreciar las características del servidor.

Tabla 24

Características del servidor donde se alojará la aplicación

Característica	Descripción
Sistema Operativo	Ubuntu 18.04 LTS
Memoria RAM	4 GB
Espacio en disco	80 GB
Procesadores	2

Elaborado por Misael Cabascango y Diego Osorio

A continuación, se muestran las configuraciones más importantes realizadas en el servidor. Primero se estableció una conexión al servidor a través de SSH y se ingresó la contraseña de root. En la Figura 28 se puede apreciar que se pudo acceder al servidor exitosamente.

Figura 28

Acceso al servidor mediante SSH.

```
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.15.0-151-generic x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

System information as of Mon Aug 30 01:02:05 UTC 2021

System load:  0.0                Processes:            99
Usage of /:   1.9% of 78.19GB    Users logged in:     0
Memory usage: 3%                IP address for eth0: 50.116.24.155
Swap usage:  0%

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

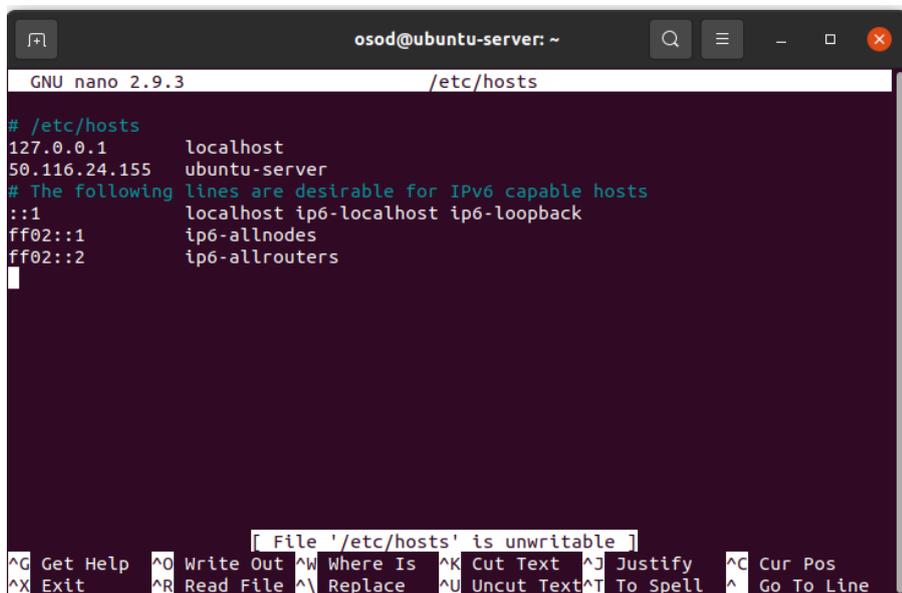
root@localhost:~#
```

Elaborado por Misael Cabascango y Diego Osorio

Posteriormente en el archivo de configuración de host se asignó un nombre para el servidor y la dirección IP proporcionada por Linode al servidor. En la Figura 29 se muestra la configuración realizada.

Figura 29

Archivo de configuración de host del servidor

A screenshot of a terminal window on an Ubuntu server. The window title is 'osod@ubuntu-server: ~'. The terminal shows the GNU nano 2.9.3 editor editing the /etc/hosts file. The content of the file is as follows:

```
# /etc/hosts
127.0.0.1    localhost
50.116.24.155  ubuntu-server
# The following lines are desirable for IPv6 capable hosts
::1         localhost ip6-localhost ip6-loopback
ff02::1    ip6-allnodes
ff02::2    ip6-allrouters
```

At the bottom of the terminal, a message reads 'File '/etc/hosts' is unwritable'. Below the message is a row of keyboard shortcuts: ^G Get Help, ^O Write Out, ^W Where Is, ^K Cut Text, ^J Justify, ^C Cur Pos, ^X Exit, ^R Read File, ^_ Replace, ^U Uncut Text, ^T To Spell, ^_ Go To Line.

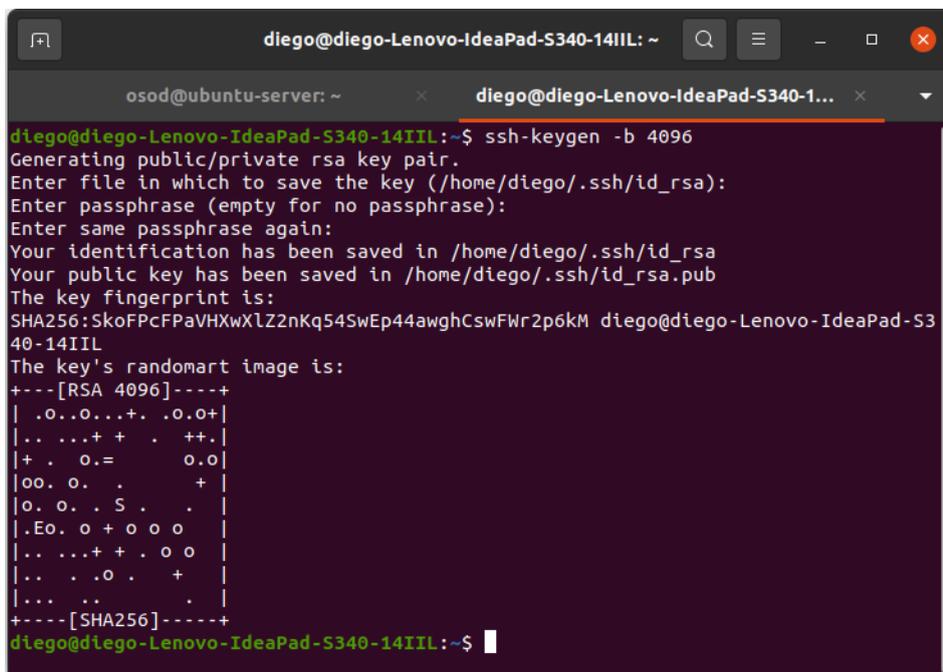
Elaborado por Misael Cabascango y Diego Osorio

Por motivos de seguridad y practicidad se creó un usuario no root en el sistema operativo y se accedió con este usuario para realizar las configuraciones que vienen a continuación.

Para acceder al servidor mediante autenticación basada en claves en lugar de usar una contraseña se generó una llave par privada pública desde una máquina local. En la Figura 30 se puede apreciar el comando utilizado para la generación de la llave. El archivo id_rsa.pub es la llave pública. En el directorio home del servidor se creó un directorio llamado .ssh y dentro de éste un archivo llamado authorized_keys, luego desde la máquina local se almacenó la llave en dicho archivo. Para concluir esta parte fue necesario configurar los permisos de lectura, escritura y ejecución para el propietario del directorio .ssh y de lectura y escritura para el propietario de los archivos que se encuentran en dicho directorio. En la Figura 31 se puede apreciar la asignación de permisos.

Figura 30

Generación de llave par privada pública



```

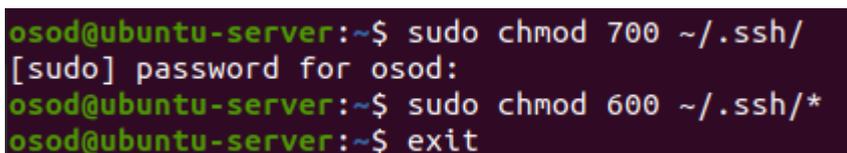
diego@diego-Lenovo-IdeaPad-S340-14IIL: ~
osod@ubuntu-server: ~ x diego@diego-Lenovo-IdeaPad-S340-1... x
diego@diego-Lenovo-IdeaPad-S340-14IIL:~$ ssh-keygen -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/diego/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/diego/.ssh/id_rsa
Your public key has been saved in /home/diego/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:SkofPcFPaVHXwXLZ2nKq54SwEp44awghCswFwr2p6kM diego@diego-Lenovo-IdeaPad-S3
40-14IIL
The key's randomart image is:
+---[RSA 4096]-----+
| .o..o...+. .o.o+|
|.. ...+ + . ++.|
|+ . o.=      o.o|
|oo. o. .    + |
|o. o. . S . . |
|.Eo. o + o o o |
|.. ...+ + . o o |
|.. . .o . + |
|... .. . |
+-----[SHA256]-----+
diego@diego-Lenovo-IdeaPad-S340-14IIL:~$

```

Elaborado por Misael Cabascango y Diego Osorio

Figura 31

Asignación de permisos a directorio .ssh y sus archivos



```

osod@ubuntu-server:~$ sudo chmod 700 ~/.ssh/
[sudo] password for osod:
osod@ubuntu-server:~$ sudo chmod 600 ~/.ssh/*
osod@ubuntu-server:~$ exit

```

Elaborado por Misael Cabascango y Diego Osorio

Se deshabilitó el inicio de sesión de usuarios root mediante SSH cambiando el valor de PermitRootLogin a No en el archivo de configuración llamado sshd_config. También se cambió el valor de Password Authentication a “No”, es decir, el acceso al servidor mediante contraseña quedó desactivado. Para aplicar estos cambios se reinició el servicio ssh con el comando:

```
sudo systemctl restart sshd
```

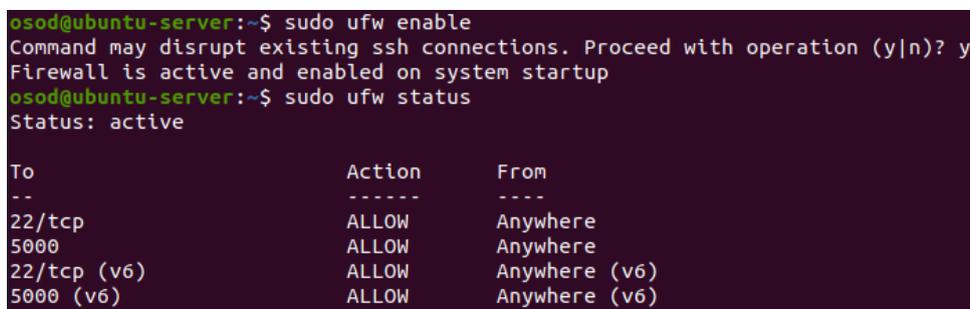
A continuación, se configuró el firewall con la ayuda de la herramienta UFW (Uncomplicated Firewall). Primero se habilitó el tráfico saliente y se deshabilitó el tráfico entrante con el comando, luego se activó el acceso por SSH y el puerto del servidor de desarrollo de Flask, el cual es el puerto 5000 con ayuda del comando:

```
sudo ufw allow
```

En la Figura 32 se muestra la activación del firewall y las configuraciones realizadas.

Figura 32

Configuración de firewall del servidor



```
osod@ubuntu-server:~$ sudo ufw enable
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y
Firewall is active and enabled on system startup
osod@ubuntu-server:~$ sudo ufw status
Status: active

To Action From
--
22/tcp ALLOW Anywhere
5000 ALLOW Anywhere
22/tcp (v6) ALLOW Anywhere (v6)
5000 (v6) ALLOW Anywhere (v6)
```

Elaborado por Misael Cabascango y Diego Osorio

3.4.2 Configuración de la aplicación

Se procedió a instalar git en el servidor para poder clonar el repositorio del proyecto con el comando git clone. Se consideró a conda como gestor de paquetes y para crear un entorno virtual con el cual poner en marcha la aplicación, es así que se instaló la última versión de Miniconda. En la Figura 33 se puede observar los paquetes disponibles al instalar Miniconda.

Figura 33*Lista de paquetes de Miniconda*

```
osod@ubuntu-server:~/miniconda3/envs$ conda list
# packages in environment at /home/osod/miniconda3:
#
# Name                    Version            Build Channel
libgcc_mutex              0.1                main
openmp_mutex              4.5                1_gnu
BrotliPy                  0.7.0             py39h27cfd23_1003
ca-certificates           2021.7.5          h06a4308_1
certifi                   2021.5.30         py39h06a4308_0
cffi                       1.14.6            py39h400218f_0
charset                    4.0.0             py39h06a4308_1003
conda                     4.10.3            py39h06a4308_0
conda-package-handling    1.7.3             py39h27cfd23_1
cryptography              3.4.7             py39hd23ed53_0
idna                      2.10              pyhd3eb1b0_0
ld_impl_linux-64         2.35.1            h7274673_9
libffi                    3.3               he6710b0_2
libgcc-ng                 9.3.0             h5101ec6_17
libgomp                   9.3.0             h5101ec6_17
libstdcxx-ng              9.3.0             hd4cf53a_17
ncurses                   6.2               he6710b0_1
openssl                   1.1.1k            h27cfd23_0
pip                       21.1.3            py39h06a4308_0
pycosat                   0.6.3             py39h27cfd23_0
pyparser                  2.20              py_2
pyopenssl                 20.0.1            pyhd3eb1b0_1
pysocks                   1.7.1             py39h06a4308_0
python                    3.9.5             h12deb9_4
readline                  8.1               h27cfd23_0
requests                  2.25.1            pyhd3eb1b0_0
ruamel_yaml               0.15.100          py39h27cfd23_0
setuptools                52.0.0            py39h06a4308_0
six                       1.16.0            pyhd3eb1b0_0
sqlite                    3.36.0            hc218d9a_0
tk                         8.6.10            hbc83047_0
tqdm                      4.61.2            pyhd3eb1b0_1
tzdata                    2021a             h52ac0ba_0
urllib3                   1.26.6            pyhd3eb1b0_1
wheel                     0.36.2            pyhd3eb1b0_0
xz                        5.2.5             h7b6447c_0
yaml                      0.2.5             h7b6447c_0
zlib                      1.2.11            h7b6447c_3
```

Elaborado por Misael Cabascango y Diego Osorio

Una vez instalado Miniconda se procedió a crear el ambiente virtual. Para ello se utilizó el archivo environment.yml que se encuentra en el repositorio de la aplicación. Dentro del directorio envs de Miniconda se ejecutó el comando:

```
conda env create -f /home/osod/recomendador/environment.yml.
```

En la Figura 34 se puede constatar la creación del ambiente virtual llamado asistente_floricola.

Figura 34*Ambiente virtual creado en el servidor*

```
osod@ubuntu-server:~/miniconda3/envs$ conda env list
# conda environments:
#
base * /home/osod/miniconda3
asistente_floricola /home/osod/miniconda3/envs/asistente_floricola
```

Elaborado por Misael Cabascango y Diego Osorio

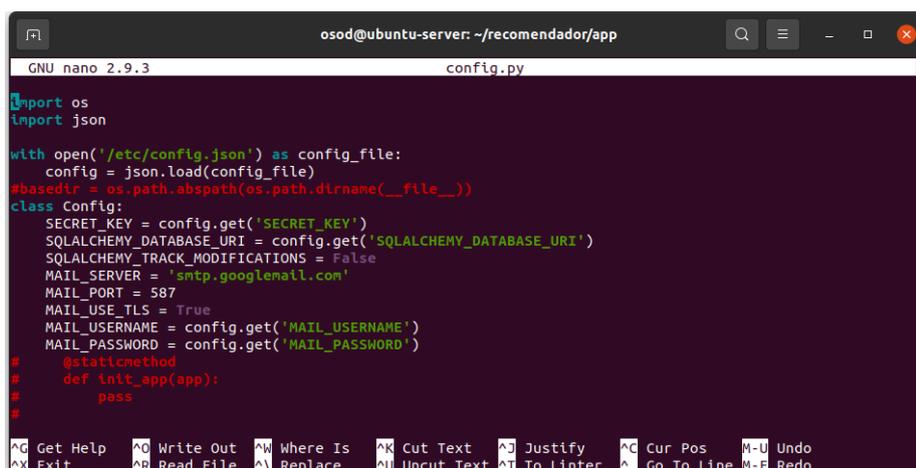
Se procede a activar el ambiente con el comando:

```
conda activate
```

A continuación, se creó un archivo de configuración en el directorio etc en formato JSON con las variables de entorno establecidas en el proceso de desarrollo y que se encuentran en el archivo config.py. Después de esto se llamó desde el archivo config.py de la aplicación al archivo de configuración recién creado como muestra la Figura 35.

Figura 35

Archivo de configuración modificado



```

osod@ubuntu-server: ~/recomendador/app
GNU nano 2.9.3 config.py
import os
import json

with open('/etc/config.json') as config_file:
    config = json.load(config_file)
#basedir = os.path.abspath(os.path.dirname(__file__))
class Config:
    SECRET_KEY = config.get('SECRET_KEY')
    SQLALCHEMY_DATABASE_URI = config.get('SQLALCHEMY_DATABASE_URI')
    SQLALCHEMY_TRACK_MODIFICATIONS = False
    MAIL_SERVER = 'smtp.googlemail.com'
    MAIL_PORT = 587
    MAIL_USE_TLS = True
    MAIL_USERNAME = config.get('MAIL_USERNAME')
    MAIL_PASSWORD = config.get('MAIL_PASSWORD')
    @staticmethod
    # def init_app(app):
    #     pass
    #
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos M-U Undo
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Linter ^_ Go To Line M-E Redo

```

Elaborado por Misael Cabascango y Diego Osorio

Posteriormente se instaló MySQL Server con el comando:

```
sudo apt-get install mysql-server
```

Una vez instalado se configuró las credenciales de acceso como se muestra en la Figura 36.

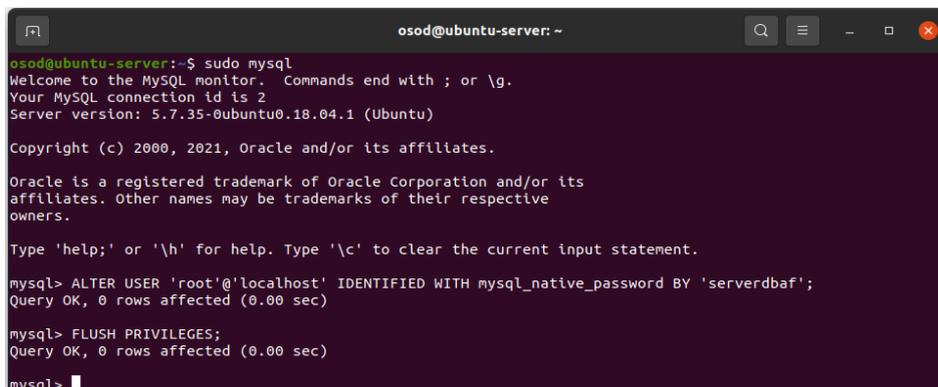
Luego se creó la base de datos con la que trabajará la aplicación, con el comando:

```
create database
```

El nombre de la base de datos es dbasistente.

Figura 36

Configuración de MySQL Server en el servidor

A terminal window titled 'osod@ubuntu-server: ~' with a dark background and light text. The terminal shows the execution of 'sudo mysql' and the MySQL command-line interface. The output includes a welcome message, connection details, and the execution of two SQL commands: 'ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'serverdbaf';' and 'FLUSH PRIVILEGES;'. Both queries return 'Query OK, 0 rows affected (0.00 sec)'.

```
osod@ubuntu-server:~$ sudo mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.7.35-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'serverdbaf';
Query OK, 0 rows affected (0.00 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

mysql>
```

Elaborado por Misael Cabascango y Diego Osorio

Antes de correr la aplicación fue necesario migrar el modelo creado con SQLAlchemy durante el desarrollo de la aplicación, a la base de datos creada en MySQL. Para ello, teniendo activado el ambiente virtual se accedió a la consola de Python. Una vez allí se ejecutó el comando:

```
db.create_all()
```

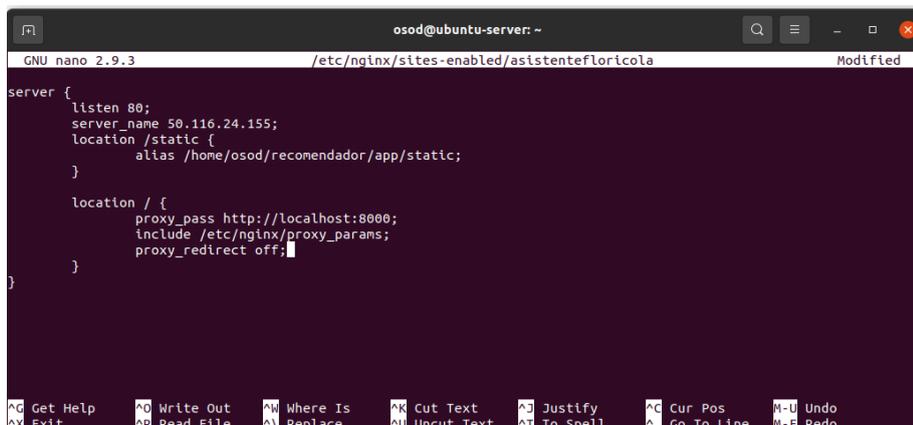
Luego se procedió a crear los roles de usuario y los permisos para cada uno en la tabla Rol utilizando el método `insert_roles` de la clase Rol. También se creó el primer usuario con rol de administrador en la tabla Usuario.

Para poder arrancar el servidor de Flask se realizó la exportación de la variable temporal `FLASK_APP`. También se configuró el parámetro `host` con la dirección IP `0.0.0.0` para poder acceder a la aplicación desde un navegador en cualquier computador.

Como servidor de producción se escogió Nginx. Para configurarlo se instaló la librería `nginx`. También se utilizó Gunicorn como WSGI, para lo cual se instaló la librería `gunicorn`. Se removió el archivo de configuración por defecto de `nginx` para crear uno nuevo. En el nuevo archivo se configuró el puerto `80` como puerto de escucha, el nombre del servidor en el cual se colocó la dirección IP proporcionada por Linode, la ubicación del directorio de los archivos estáticos de la aplicación, en `proxy_pass` se colocó el puerto `8000` que es el puerto con que trabaja Gunicorn. En la Figura 37 se puede observar el contenido del archivo de configuración.

Figura 37

Archivo de configuración de NGINX



```

GNU nano 2.9.3 /etc/nginx/sites-enabled/asistentefloricola Modified
server {
    listen 80;
    server_name 50.116.24.155;
    location /static {
        alias /home/osod/recomendador/app/static;
    }

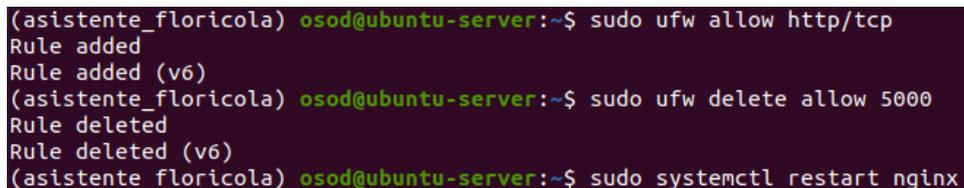
    location / {
        proxy_pass http://localhost:8000;
        include /etc/nginx/proxy_params;
        proxy_redirect off;
    }
}
  
```

Elaborado por Misael Cabascango y Diego Osorio

Luego, se activó el tráfico http y TCP en el firewall y se desactivó el puerto 5000 que era el puerto del servidor de desarrollo. Finalmente se reinició el servidor NGINX como se muestra en la Figura 38.

Figura 38

Activación del tráfico de http y TCP



```

(asistente_floricola) osod@ubuntu-server:~$ sudo ufw allow http/tcp
Rule added
Rule added (v6)
(asistente_floricola) osod@ubuntu-server:~$ sudo ufw delete allow 5000
Rule deleted
Rule deleted (v6)
(asistente_floricola) osod@ubuntu-server:~$ sudo systemctl restart nginx
  
```

Elaborado por Misael Cabascango y Diego Osorio

Posteriormente se configuró Gunicorn. El número de trabajos se configura siguiendo la fórmula:

$$2 \times \text{número de cores} + 1$$

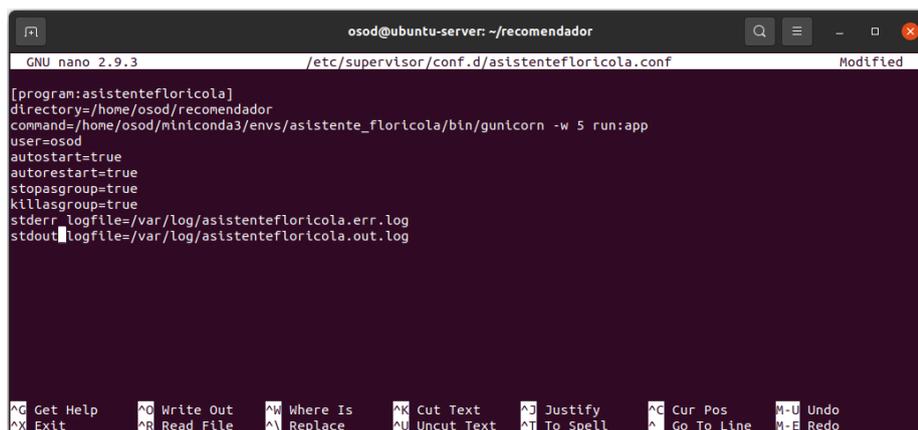
Como el procesador tiene 2 cores el parámetro obtenido fue 5. Dentro del directorio del servidor ejecutamos el siguiente comando:

```
gunicorn -w 5 run:app
```

Para gestionar GUnicorn en caso de errores o cuando este deba iniciarse o reiniciarse automáticamente se utilizó la herramienta Supervisor. En la Figura 39 se puede observar el archivo de configuración.

Figura 39

Archivo de configuración de GUnicorn



```
osod@ubuntu-server: ~/recomendador
GNU nano 2.9.3 /etc/supervisor/conf.d/asistentefloricola.conf Modified
[program:asistentefloricola]
directory=/home/osod/recomendador
command=/home/osod/miniconda3/envs/asistente_floricola/bin/gunicorn -w 5 run:app
user=osod
autostart=true
autorestart=true
stopasgroup=true
killasgroup=true
stderr_logfile=/var/log/asistentefloricola.err.log
stdout_logfile=/var/log/asistentefloricola.out.log
Get Help  Write Out  Where Is  Cut Text  Justify  Cur Pos  Undo
Exit      Read File  Replace  Uncut Text  To Spell  Go To Line  Redo
```

Elaborado por Misael Cabascango y Diego Osorio

Se creó el directorio y el archivo donde se guardarán los logs en caso de surgir errores y después se reinició Supervisor con el comando:

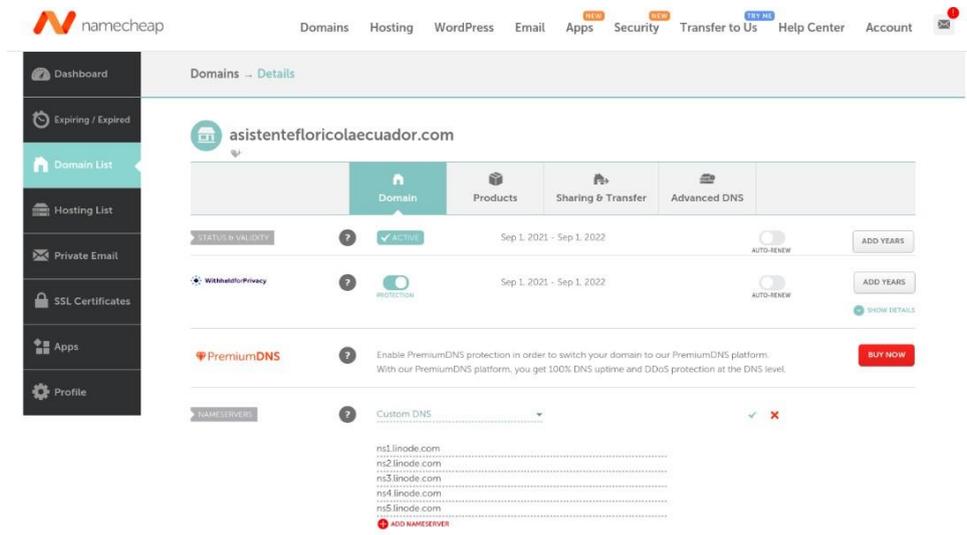
```
sudo supervisorctl reload
```

Ahora la aplicación se encuentra ejecutándose en el servidor de producción.

Para adquirir un dominio para la aplicación se consideró el sitio namecheap.com. El nombre del dominio es asistentefloricolaecuador.com y se adquirió por el plazo de 1 año. En la Figura 40 se puede apreciar la pantalla de administración de Namecheap. De acuerdo a la página de ayuda de Linode, lo siguiente que se debe hacer es añadir los nombres de dominio proporcionados por Linode en el dominio que se adquirió. En la misma pantalla se escoge la opción CustomDNS y luego se ingresan los nombres de dominio. El tiempo que el servidor DNS toma en actualizarse está entre 24 y 48 horas.

Figura 40

Pantalla de administración de Namecheap

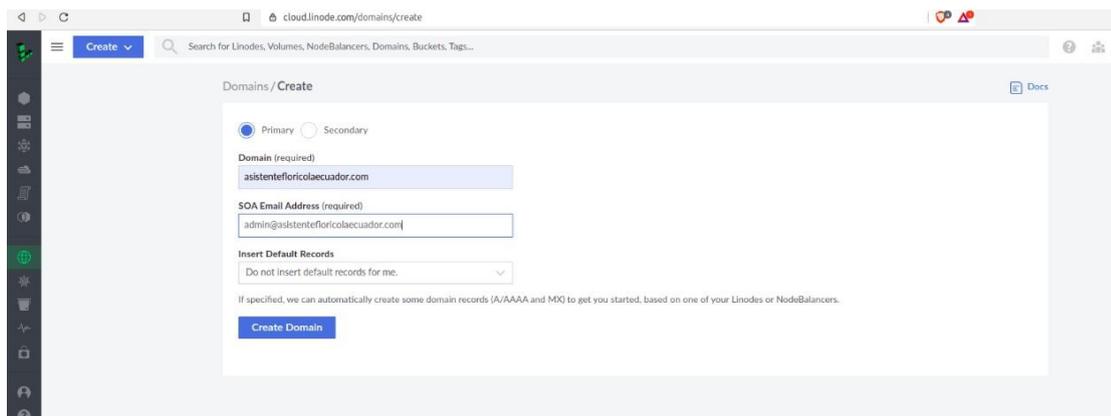


Elaborado por Misael Cabascango y Diego Osorio

Lo siguiente que se hizo fue añadir un nuevo dominio en el servidor en Linode con el nombre del dominio adquirido en Namecheap, como se observa en la Figura 41.

Figura 41

Creación de dominio en el servidor en Linode



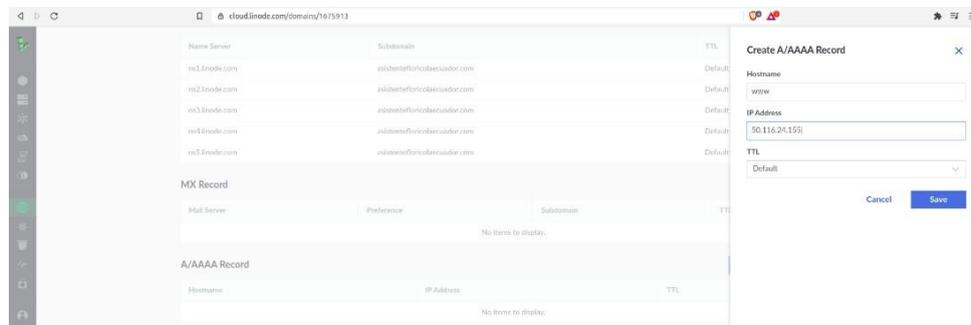
Elaborado por Misael Cabascango y Diego Osorio

Luego, fue necesario añadir registros AAAA en el dominio creado, desde el panel de administración de dominios. En el campo hostname se coloca www y en el campo IPAddress,

la dirección IP del servidor. En la Figura 42 se puede observar la creación del registro mencionado.

Figura 42

Creación de un registro AAAA



Elaborado por Misael Cabascango y Diego Osorio

El siguiente paso realizado fue establecer DNS inverso. “Las computadoras usan DNS para determinar la dirección IP asociada con un nombre de dominio. La búsqueda de DNS inversa hace lo contrario al resolver una dirección IP en un nombre de dominio designado” (Linode, 2021). Para realizar esto se ingresó al panel de administración del servidor, en la pestaña Networking. En la sección de direcciones IPv4 existe un registro con la IP del servidor. Allí se abrió un menú y se escogió la opción Edit RDNS. En el campo de texto se colocó nuestro nombre de dominio, antecedido de www. Una vez realizado este proceso, fue posible acceder al sitio, desde cualquier navegador, escribiendo en la barra de direcciones: www.asistentefloricolaecuador.com

Finalmente se activó https para hacer el sitio seguro y se obtuvo un certificado SSL/TLS gratuito utilizando el sitio web letsencrypt.org. Para ello se instaló la herramienta Certbot, a través de los siguientes comandos:

- `sudo apt-get update`
- `sudo apt-get install software-properties-common`
- `sudo add-apt-repository universe`
- `sudo add-apt-repository ppa:certbot/certbot`
- `sudo apt-get update`

- `sudo apt-get install python-certbot-nginx`

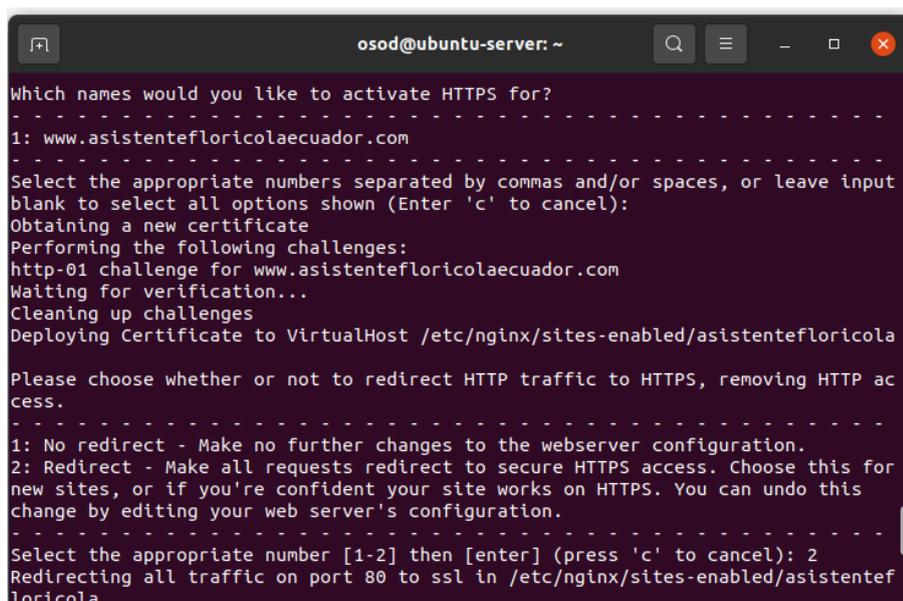
En el archivo de configuración de nginx se cambió el parámetro `name_server` por el nombre de dominio. En la Figura 44 se muestra el proceso de activación del certificado ejecutando el comando:

```
sudo certbot -nginx
```

El nombre de dominio para activar https es `asistentefloricolaecuador.com`, y se eligió la opción Hacer que todas las solicitudes se redirijan a un acceso HTTPS seguro. Una vez terminado el proceso, se obtiene el certificado de activación y el archivo de configuración de nginx se muestra como en la Figura 45.

Figura 44

Activación de certificado con Certbot



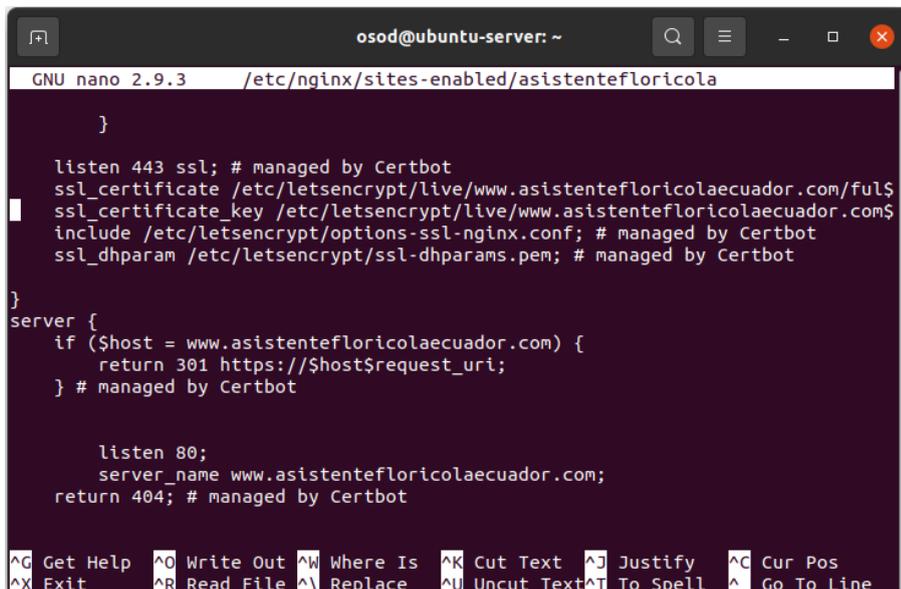
```
osod@ubuntu-server: ~
Which names would you like to activate HTTPS for?
-----
1: www.asistentefloricolaecuador.com
-----
Select the appropriate numbers separated by commas and/or spaces, or leave input
blank to select all options shown (Enter 'c' to cancel):
Obtaining a new certificate
Performing the following challenges:
http-01 challenge for www.asistentefloricolaecuador.com
Waiting for verification...
Cleaning up challenges
Deploying Certificate to VirtualHost /etc/nginx/sites-enabled/asistentefloricola

Please choose whether or not to redirect HTTP traffic to HTTPS, removing HTTP ac
cess.
-----
1: No redirect - Make no further changes to the webserver configuration.
2: Redirect - Make all requests redirect to secure HTTPS access. Choose this for
new sites, or if you're confident your site works on HTTPS. You can undo this
change by editing your web server's configuration.
-----
Select the appropriate number [1-2] then [enter] (press 'c' to cancel): 2
Redirecting all traffic on port 80 to ssl in /etc/nginx/sites-enabled/asistentef
loricola
```

Elaborado por Misael Cabascango y Diego Osorio

Figura 45

Archivo de configuración de NGINX con certificado SSL/TLS activado



```

GNU nano 2.9.3 /etc/nginx/sites-enabled/asistentefloricola

}

listen 443 ssl; # managed by Certbot
ssl_certificate /etc/letsencrypt/live/www.asistentefloricolaecuador.com/ful$
ssl_certificate_key /etc/letsencrypt/live/www.asistentefloricolaecuador.com$
include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot
}
server {
    if ($host = www.asistentefloricolaecuador.com) {
        return 301 https://$host$request_uri;
    } # managed by Certbot

    listen 80;
    server_name www.asistentefloricolaecuador.com;
    return 404; # managed by Certbot
}

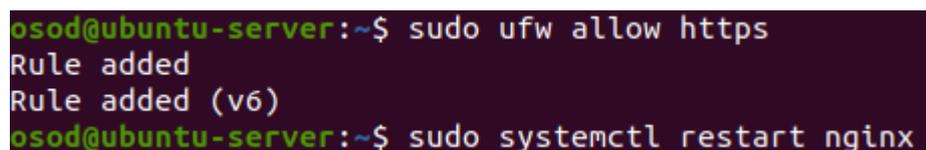
```

Elaborado por Misael Cabascango y Diego Osorio

Ahora el servidor tiene activado el puerto 443 como puerto de escucha, el cual es SSL. El tráfico http ahora es redirigido a https. Finalmente se activó el tráfico https en el firewall y se reinició el servidor nginx como se muestra en la Figura 46.

Figura 46

Activación de tráfico https en el firewall del servidor



```

osod@ubuntu-server:~$ sudo ufw allow https
Rule added
Rule added (v6)
osod@ubuntu-server:~$ sudo systemctl restart nginx

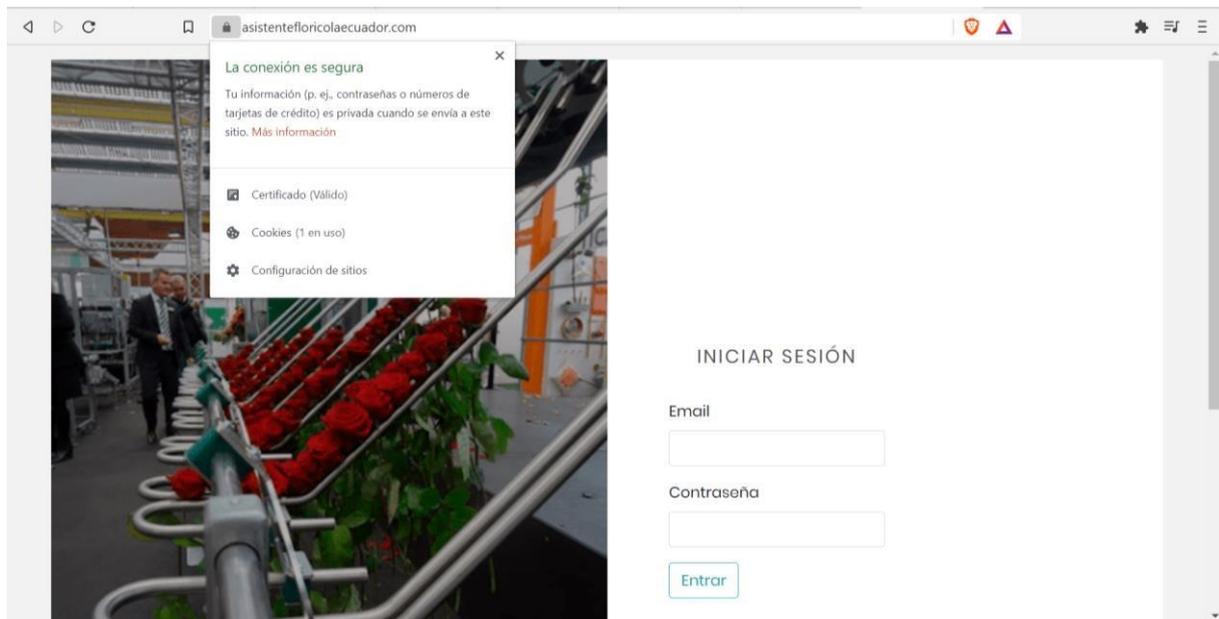
```

Elaborado por Misael Cabascango y Diego Osorio

Como se puede observar en la Figura 47 el sitio web tiene un certificado SSL/TSL y el acceso a éste es seguro.

Figura 47

Aplicación con certificado SSL/TSL activado



Elaborado por Misael Cabascango y Diego Osorio

3.5 PRUEBAS

3.5.1 PRUEBA DE FUNCIONALIDAD

Tabla 25

Prueba de funcionalidad 1

Prueba funcional	1	Módulo	Usuario
Descripción	Ingreso a la página	Prioridad	Alta
Caso prueba			
N°	Descripción	Resultado esperado	Resultado obtenido
1	Enviar con campos vacíos.	Mensaje “Debe llenar los campos”	Mensaje “Debe llenar los campos”.

2	Ingresar texto en el correo.	Mensaje “Debe ingresar un correo válido.”	Mensaje “Ingrese un correo válido”.
---	------------------------------	---	-------------------------------------

Elaborado por Misael Cabascango y Diego Osorio.

Tabla 26

Prueba de funcionalidad 2

Prueba funcional	2	Módulo	Usuario
Descripción	Comprobar el registro de datos	Prioridad	Alta
Caso prueba			
N°	Descripción	Resultado esperado	Resultado obtenido
1	Enviar con campos vacíos.	Mensaje “llene este campo”	Mensaje “Complete este campo”
2	Ingreso de fecha superior a la actual	Mensaje “la fecha no debe ser mayor a la actual”	Mensaje” La fecha de la medición no puede ser mayor a la fecha de hoy”
3	Ingreso de igual muestreo en la misma cama	Mensaje “ya existe este dato”	Mensaje “Muestre ya registrado”

Elaborado por Misael Cabascango y Diego Osorio.

Tabla 27

Prueba de Funcionalidad 3

Prueba funcional	3	Módulo	Usuario
Descripción	Comprobar el registro de nuevos usuarios	Prioridad	Alta
Caso prueba			
N°	Descripción	Resultado esperado	Resultado obtenido
1	Ingresar datos personales	Mensaje “ingrese este campo”	Mensaje “Complete este campo”

2	Ingresar texto en el correo.	Mensaje “Debe ingresar un correo válido.”	Mensaje “Ingrese un correo válido”.
3	Ingresar correo repetido	Mensaje “el correo ya se ha utilizado”	Mensaje “este correo ya está registrado”
4	Ingreso de contraseña	Mensaje “la contraseña no tiene 8 caracteres”	Mensaje “La contraseña debe tener como mínimo 8 caracteres”

Elaborado por Misael Cabascango y Diego Osorio.

Tabla 28

Prueba de Funcionalidad 4

Prueba funcional	4	Módulo	Usuario
Descripción	Recuperar contraseña	Prioridad	Alta
Caso prueba			
N°	Descripción	Resultado esperado	Resultado obtenido
1	Correo no registrado	Mensaje de advertencia que el correo no está registrado	Mensaje “Correo no registrado”.
2	Ingreso de contraseña	Mensaje la contraseña no debe ser menor a 8 caracteres	Mensaje “La contraseña debe tener como mínimo 8 caracteres”

Elaborado por Misael Cabascango y Diego Osorio.

Tabla 29

Prueba de Funcionalidad 5

Prueba funcional	5	Módulo	Usuario
Descripción	Historial recomendaciones	Prioridad	Alta
Caso prueba			
N°	Descripción	Resultado esperado	Resultado obtenido
1	Seleccionar fecha	Mensaje “debe llenar campos”	Mensaje “Complete este campo”.

2	Imprimir historial	Mensaje “haga una búsqueda”	Mensaje “Realice primero una búsqueda”
3	Búsqueda con fecha superior a la actual	Mensaje “la fecha no debe ser superior a la actual”	Mensaje “No existen resultados”

Elaborado por Misael Cabascango y Diego Osorio.

Tabla 30

Prueba de Funcionalidad 6

Prueba funcional	6	Módulo	Usuario
Descripción	Historial mediciones	Prioridad	Alta
Caso prueba			
N°	Descripción	Resultado esperado	Resultado obtenido
1	Imprimir historial sin seleccionar	Mensaje “debe llenar campos”	Mensaje “Seleccione un elemento de la lista”.
2	Imprimir por bloque, cama o muestreo	Mensaje “llenar campos”	Mensaje “complete este campo”
3	Imprimir por variedad	Mensaje “seleccione la variedad”	Mensaje “selecciona un campo”
4	Imprimir por fecha	Mensaje “la fecha no debe ser superior a la actual”	Mensaje “No existen resultados”

Elaborado por Misael Cabascango y Diego Osorio.

Tabla 31

Prueba de Funcionalidad 7

Prueba funcional	7	Módulo	Usuario
Descripción	Eliminar usuario	Prioridad	Alta
Caso prueba			
N°	Descripción	Resultado esperado	Resultado obtenido
1	Eliminar usuario	Mensaje “Se ha eliminado el usuario”	Mensaje “Usuario eliminado”.

2	Eliminar ultimo usuario administrador	Mensaje “no se puede eliminar”	Mensaje “No se puede eliminar más usuarios”

Elaborado por Misael Cabascango y Diego Osorio.

Tabla 32

Prueba de Funcionalidad 8

Prueba funcional	8	Módulo	Usuario
Descripción	Actualizar usuario	Prioridad	Alta
Caso prueba			
N°	Descripción	Resultado esperado	Resultado obtenido
1	Ingresar nuevos datos personales	Mensaje “llene este campo”	Mensaje “Complete este campo”.
2	Ingreso nuevo correo	Mensaje “este correo ya está registrado”	Mensaje “Correo ya registrado”

Elaborado por Misael Cabascango y Diego Osorio.

Tabla 33

Prueba de Funcionalidad 9

Prueba funcional	9	Módulo	Usuario
Descripción	Ingreso de datos de variables	Prioridad	Alta
Caso prueba			
N°	Descripción	Resultado esperado	Resultado obtenido
1	Ingresar de datos de sustrato liquido (pH, densidad y conductividad eléctrica)	Mensaje “llene este campo”	Mensaje “Complete este campo”.
2	Ingreso de datos compost, tanque y goteo	Guardar datos vacíos como null.	Guardar datos en la base como 0

Elaborado por Misael Cabascango y Diego Osorio.

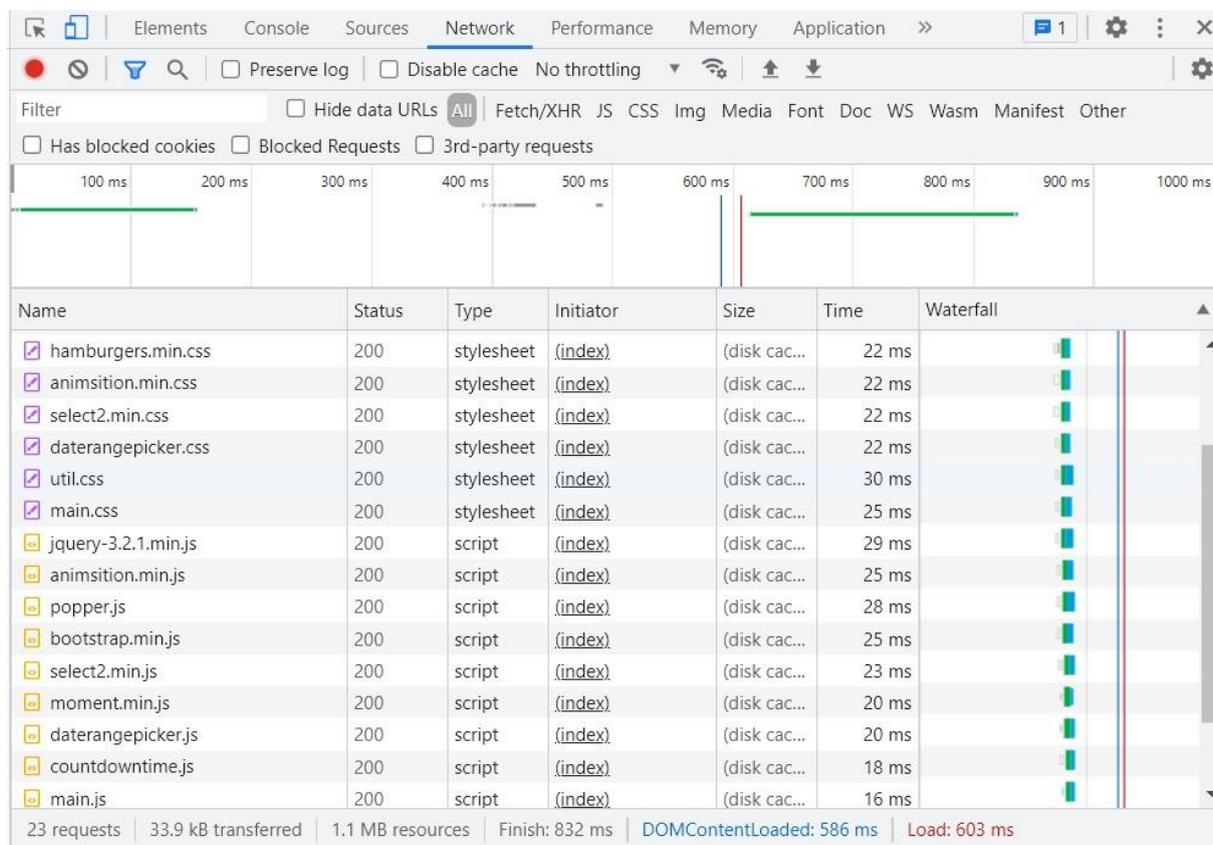
Tabla 34*Prueba de Funcionalidad 10*

Prueba funcional	10	Módulo	Usuario
Descripción	Guardar recomendación	Prioridad	Alta
Caso prueba			
N°	Descripción	Resultado esperado	Resultado obtenido
1	Si no está de acuerdo con la recomendación	Mensaje “seleccione un elemento”	Mensaje “Seleccione un elemento de la lista”.
2	Si está de acuerdo con la recomendación	Guardar recomendación	Guardar recomendación

Elaborado por Misael Cabascango y Diego Osorio.

3.5.2 PRUEBA DE EFICIENCIA

Para evaluar la eficiencia de nuestro software se han probado el tiempo de carga de nuestra aplicación obteniendo los datos que se pueden ver en la Figura 43.

Figura 43*Tiempo de carga*

Elaborado por Misael Cabascango y Diego Osorio.

Como se puede observar en la Figura 43, se ha demorado 832ms (es decir 0.832 segundos) en cargar la página home que se usó en este ejemplo.

A continuación, se evalúa el tiempo de cargar con 200 peticiones en un segundo, para realizar las pruebas de carga del asistente agrícola se ha utilizado la herramienta de Apache JMETER, dando como resultado la respuesta de ok en 200/200 de las peticiones requeridas como se muestra en la Figura 44.

Figura 44

Prueba de carga con 200 peticiones.

Muestra #	Tiempo de comi...	Nombre del hilo	Etiqueta	Tiempo de Mue...	Estado	Bytes	Sent Bytes	Latency	Connect Time(...
175	16:00:24.187	Grupo de Hilos ...	Peticion HTTP	5529	✓	5457	138	5529	4555
176	16:00:24.590	Grupo de Hilos ...	Peticion HTTP	5128	✓	5457	138	5128	4134
177	16:00:24.856	Grupo de Hilos ...	Peticion HTTP	4864	✓	5457	138	4864	3879
178	16:00:24.337	Grupo de Hilos ...	Peticion HTTP	5386	✓	5457	138	5386	4386
179	16:00:24.401	Grupo de Hilos ...	Peticion HTTP	5325	✓	5457	138	5325	4348
180	16:00:24.357	Grupo de Hilos ...	Peticion HTTP	5371	✓	5457	138	5371	4358
181	16:00:24.598	Grupo de Hilos ...	Peticion HTTP	5132	✓	5457	138	5132	4145
182	16:00:24.594	Grupo de Hilos ...	Peticion HTTP	5136	✓	5457	138	5136	4149
183	16:00:24.244	Grupo de Hilos ...	Peticion HTTP	5490	✓	5457	138	5490	4471
184	16:00:24.265	Grupo de Hilos ...	Peticion HTTP	5469	✓	5457	138	5469	4465
185	16:00:24.755	Grupo de Hilos ...	Peticion HTTP	4980	✓	5457	138	4979	3987
186	16:00:24.430	Grupo de Hilos ...	Peticion HTTP	5305	✓	5457	138	5305	4319
187	16:00:24.888	Grupo de Hilos ...	Peticion HTTP	4847	✓	5457	138	4847	3838
188	16:00:24.187	Grupo de Hilos ...	Peticion HTTP	5548	✓	5457	138	5548	4550
189	16:00:24.097	Grupo de Hilos ...	Peticion HTTP	5639	✓	5457	138	5639	4630
190	16:00:24.452	Grupo de Hilos ...	Peticion HTTP	5295	✓	5457	138	5295	4283
191	16:00:24.713	Grupo de Hilos ...	Peticion HTTP	5035	✓	5457	138	5035	4024
192	16:00:24.698	Grupo de Hilos ...	Peticion HTTP	5054	✓	5457	138	5053	4035
193	16:00:24.366	Grupo de Hilos ...	Peticion HTTP	5388	✓	5457	138	5388	4372
194	16:00:24.099	Grupo de Hilos ...	Peticion HTTP	5657	✓	5457	138	5657	4626
195	16:00:24.587	Grupo de Hilos ...	Peticion HTTP	5172	✓	5457	138	5172	4153
196	16:00:24.120	Grupo de Hilos ...	Peticion HTTP	5642	✓	5457	138	5642	4605
197	16:00:24.661	Grupo de Hilos ...	Peticion HTTP	5104	✓	5457	138	5104	4081
198	16:00:24.892	Grupo de Hilos ...	Peticion HTTP	4879	✓	5457	138	4879	3844
199	16:00:24.099	Grupo de Hilos ...	Peticion HTTP	5674	✓	5457	138	5674	4645
200	16:00:24.757	Grupo de Hilos ...	Peticion HTTP	5017	✓	5457	138	5017	3981

Scroll automatically?
 Child samples?
 No. de Muestras 200
 Última Muestra 5017
 Media 4940
 Desviación 423

Elaborado por Misael Cabascango y Diego Osorio.

Para finalizar la prueba de carga se ha realizado 1000 peticiones en un segundo.

Figura 45

Prueba de carga con 1000 peticiones.

Muestra #	Tiempo de comi...	Nombre del hilo	Etiqueta	Tiempo de Mue...	Estado ↓	Bytes	Sent Bytes	Latency	Connect Time(...
720	16:07:16.342	Grupo de Hilos ...	Peticion HTTP	9072	✓	6191	293	8856	8761
721	16:07:16.928	Grupo de Hilos ...	Peticion HTTP	8603	✓	6191	293	8398	8294
722	16:07:16.937	Grupo de Hilos ...	Peticion HTTP	8594	✓	6191	293	8390	8285
723	16:07:16.356	Grupo de Hilos ...	Peticion HTTP	9201	✓	6191	293	8986	8877
724	16:07:17.686	Grupo de Hilos ...	Peticion HTTP	7879	✓	6191	293	7665	7556
725	16:07:16.293	Grupo de Hilos ...	Peticion HTTP	9371	✓	6191	293	9166	9063
726	16:07:17.830	Grupo de Hilos ...	Peticion HTTP	7970	✓	6191	293	7756	7647
727	16:07:16.935	Grupo de Hilos ...	Peticion HTTP	8880	✓	6191	293	8776	8673
728	16:07:16.935	Grupo de Hilos ...	Peticion HTTP	8885	✓	6191	293	8781	8677
729	16:07:16.348	Grupo de Hilos ...	Peticion HTTP	9574	✓	6191	293	9367	9263
730	16:07:16.297	Grupo de Hilos ...	Peticion HTTP	9639	✓	6191	293	9428	9321
731	16:07:16.218	Grupo de Hilos ...	Peticion HTTP	9751	✓	6191	293	6431	3814
732	16:07:16.183	Grupo de Hilos ...	Peticion HTTP	9827	✓	6191	293	3553	3242
733	16:07:16.221	Grupo de Hilos ...	Peticion HTTP	9852	✓	6191	293	9647	3910
734	16:07:16.228	Grupo de Hilos ...	Peticion HTTP	9850	✓	6191	293	9745	9641
735	16:07:16.221	Grupo de Hilos ...	Peticion HTTP	9863	✓	6191	293	9650	3962
736	16:07:16.214	Grupo de Hilos ...	Peticion HTTP	9896	✓	6191	293	6380	3732
737	16:07:16.354	Grupo de Hilos ...	Peticion HTTP	9839	✓	6191	293	9625	9516
738	16:07:16.193	Grupo de Hilos ...	Peticion HTTP	10072	✓	6191	293	3831	3396
739	16:07:16.174	Grupo de Hilos ...	Peticion HTTP	10102	✓	6191	293	3550	3229
740	16:07:16.344	Grupo de Hilos ...	Peticion HTTP	9963	✓	6191	293	9748	9641
741	16:07:16.220	Grupo de Hilos ...	Peticion HTTP	10146	✓	6191	293	4350	3884
742	16:07:16.203	Grupo de Hilos ...	Peticion HTTP	10164	✓	6191	293	6416	3579
743	16:07:16.194	Grupo de Hilos ...	Peticion HTTP	10326	✓	6191	293	3870	3415
744	16:07:16.227	Grupo de Hilos ...	Peticion HTTP	10881	✓	6191	293	10669	4136
745	16:07:16.174	Grupo de Hilos ...	Peticion HTTP	11122	✓	6191	293	3462	3139

Scroll automatically?
 Child samples?
 No. de Muestras 745
 Última Muestra 11122
 Media 6193
 Desviación 1745

Elaborado por Misael Cabascango y Diego Osorio.

Como se puede apreciar en la Figura 45 la aplicación llevo hasta 745 peticiones, es decir que a partir del 746 la aplicación cae.

CONCLUSIONES

En el presente trabajo se realizó un aplicativo web para agricultura de precisión aplicando inteligencia artificial para generar una recomendación automática acerca de actividades agrícolas a partir de datos históricos de pH, densidad y conductividad eléctrica. Se trabajó en lenguaje de programación Python enlazado a una base de datos MySQL. Se utilizó una red neuronal Deep Learning en Keras para la creación del modelo predictivo. Jira fue una herramienta de gran utilidad al momento de aplicar Scrum en este proyecto.

El módulo de ingreso de datos contemplaba inicialmente las variables pH, densidad y conductividad eléctrica. Un análisis más exhaustivo de la data disponible evidenció la necesidad de tomar en cuenta más variables como fecha, muestreo, cama, variedad, ciclo, bloque, longitud de los botones y la población en cada estadio. El módulo fue ampliado para el ingreso de estas variables. Además, permitirá el ingreso de datos de pH, densidad y conductividad eléctrica en varias instancias de la plantación: compost, tanque, goteo y programación.

Mediante la inteligencia artificial que se usó en el aplicativo web, se pudo brindar una recomendación en base a las variables registradas. El modelo predictivo logró una precisión de 89.60% (+/- 13.71%).

Se implementó el módulo de seguridad con dos posibles usuarios: Administrador y Técnico Agrícola, con las restricciones pertinentes.

Se implementó un módulo de reportes el cual genera un archivo xls para las mediciones y un archivo pdf para las recomendaciones mediante criterios de búsqueda y rangos de fechas.

Se realizaron pruebas de funcionalidad y eficiencia obteniéndose resultados favorables, en el sentido que la aplicación cumplió los requerimientos de usuario. No se pudo realizar pruebas de fiabilidad debido a la premura del tiempo.

RECOMENDACIONES

Realizar una capacitación a nuevos usuarios sobre el funcionamiento del asistente agrícola, para que no exista inconvenientes al momento de usarlo.

Revisar detenidamente cada campo del asistente agrícola para así evitar errores de ingreso y guardado.

Si se desea realizar un cambio al código, se recomienda descargarlo de GitHub y realizar un respaldo e instalar todos los paquetes en el archivo requirements.txt.

Elaborar un modelo de clasificación que tome en cuenta más variables de una medición que de las que se tomaron en este trabajo, para conseguir un modelo más preciso.

Separar el modelo de clasificación de la aplicación en sí, de modo que se pueda consumir el modelo de clasificación como un servicio (API).

Investigar sobre otras librerías de Flask que pueden ayudar a mejorar la eficiencia, el rendimiento y la seguridad de la aplicación.

REFERENCIAS

- Anthura. (08 de noviembre de 2018). Obtenido de <https://www.anthura.nl/growing-advise/la-influencia-del-ph-en-el-cultivo/?lang=es>
- Asociación de Productores y Exportadores de Flores. (2020). *Reporte Estadístico Anual 2020*.
- Bongiovanni, R., Chartuni Mantovani, E., Best, S., & Roel, Á. (2006). *AGRICULTURA DE PRECISIÓN: Integrando conocimientos*. Montevideo, Uruguay: PROCISUR.
- Bonilla Segovia, J. S., Dávila Rojas, F. A., & Villa Quishpe, M. (2021). Estudio del uso de técnicas de inteligencia artificial aplicadas para análisis de suelos para el sector agrícola. *Revista Científica Mundo de la Investigación y el Conocimiento*.
- Brownlee, J. (15 de Enero de 2020). *machine learning mastery*. Obtenido de machine learning mastery: <https://machinelearningmastery.com/random-oversampling-and-undersampling-for-imbalanced-classification/>
- Del Mar, M. (29 de Noviembre de 2020). *espaciociencia.com*. Obtenido de espaciociencia.com: <https://espaciociencia.com/densidad/>
- Flewelling, P. (2018). *The Agile Developer's Handbook : Get More Value from Your Software Development: Get the Best Out of the Agile Methodology*.
- Fundación Aquae. (s.f.). Recuperado el 17 de 03 de 2021, de <https://www.fundacionaquae.org/densidad-agua-aceite-hielo/>
- García Bello, D. (28 de 11 de 2019). Obtenido de <https://culturacientifica.com/2019/11/28/que-es-el-ph/>
- García, B. (28 de Noviembre de 2019). *cultura científica*. Obtenido de cultura científica: <https://culturacientifica.com/2019/11/28/que-es-el-ph/>
- Infoagro. (s.f.). Obtenido de https://www.infoagro.com/instrumentos_medida/doc_conductividad_electrica.asp?k=

- International Society of Precision Agriculture. (s.f.). *Precision Agriculture Definition*.
Recuperado el 14 de marzo de 2021, de <https://www.ispag.org/about/definition>
- Ionos*. (8 de Octubre de 2020). Obtenido de Ionos: <https://www.ionos.es/digitalguide/online-marketing/marketing-para-motores-de-busqueda/que-es-keras/>
- Islam, A., Akter, K., Jahan Nipu, N., Das, A., Rahman, M., & Rahman, M. (2018). IoT Based Power Efficient Agro Field Monitoring and Irrigation Control System : An Empirical Implementation in Precision Agriculture.
- Linode. (6 de agosto de 2021). *Configure Your Linode for Reverse DNS (rDNS)*. Obtenido de <https://www.linode.com/docs/guides/configure-your-linode-for-reverse-dns/>
- Meza, J., Zambrano, G., & Ortega, H. (2021). DEPURACIÓN DE LA DATA PROVENIENTE DE CULTIVOS DE FLORES (HUMEDAD, CONDUCTIVIDAD ELÉCTRICA, TEMPERATURA, ETC.) Y GENERACIÓN DE UN MODELO PREDICTIVO BÁSICO CON MACHINE LEARNING.
- Muro, J. (s.f.). Obtenido de <https://www2.deloitte.com/es/es/pages/technology/articles/que-es-orm.html>
- Robledano, A. (24 de Septiembre de 2019). *openwebinars*. Obtenido de openwebinars: <https://openwebinars.net/blog/que-es-mysql/>
- Rodríguez, S., Gualotuña, T., & Grilo, C. (2017). A System for the Monitoring and Predicting of Data in Precision Agriculture in a Rose Greenhouse Based on Wireless Sensor. *Procedia Computer Science*, 306-313.
- Schwaber, K., & Sutherland, J. (2020). *The 2020 Scrum Guide*. Obtenido de <https://scrumguides.org/scrum-guide.html#daily-scrum>
- Triantafyllou, A., Tsouros, D., Sarigiannidis, P., & Bibi, S. (2019). An Architecture model for Smart Farming.

Wikipedia. (7 de Agosto de 2020). Obtenido de Wikipedia:

<https://es.wikipedia.org/wiki/Agricultura>

Wikipedia. (8 de Agosto de 2021). Obtenido de Wikipedia:

<https://en.wikipedia.org/wiki/PyCharm>

Zofío Jiménez, J. (2013). *Aplicaciones Web*. Macmillan Iberia, S.A.