

**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

**Facultad de Ingeniería en Electricidad y Computación**

Sistema de predicción de datos ambientales de una red de sensores  
inalámbrica en cultivos agrícolas

**PROYECTO INTEGRADOR**

Previo la obtención del Título de:

**Ingeniero en Computación**

Presentado por:

DENNYS EDUARDO LÓPEZ DAMIÁN

WILLIAM STEEVEN MEDINA MASA

GUAYAQUIL - ECUADOR

Año: 2021

## **DEDICATORIA**

En el presente proyecto lo dedico a mis padres que con su amor y sacrificio siempre me han apoyado en mis estudios.

A mis abuelos que siempre se han preocupado por mí, a mi abuela que, aunque ya no este en esta etapa de mi vida hubiese estado muy feliz.

A mi familia que siempre están en los momentos más difíciles y que son mi razón de seguir.

**William Steeven Medina Masa**

## **DEDICATORIA**

En el presente proyecto lo dedico a mi padre y madre que con su sacrificio y sabiduría han sabido guiarme por el camino del bien, apoyándome en cada paso de mi vida académica en los buenos y malos momentos.

Mis padres se preocupaban mucho de mi bienestar y siempre están cuando los necesito, sobre todo en los días difíciles

También no olvidaré a mis hermanos que siempre hemos pasado buenos momentos juntos, muchas gracias.

**Dennys López Damián**

## **AGRADECIMIENTOS**

Primeramente, agradezco a mis padres que me apoyan siempre y hermanos que han estado para mí, a mi familia en general que con su apoyo incondicional sigo yendo hacia adelante.

A todos los que fueron mis compañeros y en especialmente a los que llegaron a ser mis amigos, algunos fueron una guía y apoyo moral para poder superarme, ayudándome con mis problemas y enseñándome cosas nuevas.

Para finalizar un agradecimiento a todos los profesores con los cuales me he encontrado, ellos me han enseñado el valor que tiene esforzarse y la recompensa de ello, a mi Tutor de tesis el PhD. Miguel Andrés Realpe e Ing. Boris Vintimilla quienes han sido una guía en esta nueva etapa de vida.

**Dennys López Damián**

## **AGRADECIMIENTOS**

Agradezco a mis padres que siempre se preocupan por mí y me apoyan en todo lo posible, me han inculcado valores muy importantes y he aprendido de ambos.

A mis compañeros de la ESPOL desde el primer semestre hasta el último y a los que lograron ser mis amigos, dándome buenos consejos para ser una mejor persona.

Para finalizar a todos mis profesores que he tenido desde la escuela hasta la universidad, ellos me han enseñado como el conocimiento es tan valioso, a mi tutor de tesis el PhD. Miguel Andrés Realpe e Ing. Boris Veintimilla que nos han brindado su ayuda en esta última etapa universitaria.

**William Steeven Medina Masa**

## DECLARACIÓN EXPRESA

“Los derechos de titularidad y explotación, nos corresponde conforme al reglamento de propiedad intelectual de la institución; WILLIAM STEEVEN MEDINA MASA Y DENNYS EDUARDO LÓPEZ DAMIÁN damos nuestro consentimiento para que la ESPOC realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual”

A handwritten signature in black ink. The name 'William M.' is written in a cursive style, with a large, stylized initial 'M' that loops back to underline the name.

William Steeven Medina Masa

A handwritten signature in black ink. The name 'López Dennys' is written in a cursive style, with a large, stylized initial 'L' that loops back to underline the name.

Dennys Eduardo López Damián

# EVALUADORES

.....  
**Ing. Boris Xavier Vintimilla**

PROFESOR DE LA MATERIA

.....  
**Dr. Miguel Realpe Robalino**

PROFESOR TUTOR

## RESUMEN

En la actualidad el desconocimiento de la variación de la temperatura afecta la toma de decisiones de los agricultores, por lo que no conocen cuando es el momento ideal para sembrar, cosechar y que cantidad de agua regar en sus cultivos, debido a que la temperatura es un factor importante en la agricultura. Por eso, es necesario implementar un sistema de predicción de datos ambientales 100% funcional de una red de sensores inalámbrica en cultivos agrícolas para que facilite la toma de decisiones a los agricultores.

Se realizaron pruebas experimentales con un *dataset* de temperatura del INAMHI (Instituto Nacional de Meteorología e Hidrología) que sirvió como fuente de datos para realizar las predicciones. Se evaluaron 3 modelos de inteligencia artificial ARIMA, LSTM y Transformer que al variar sus parámetros y con diferentes métricas para analizar sus errores, se seleccionó el mejor modelo para ser implementado en el sistema de predicción. Adicionalmente, se implementaron interfaces gráficas para que el agricultor y administrador puedan interactuar con el sistema de predicción, para esto se usó el *framework* de Angular para el *frontend* y el *framework* Django de *python* para el *backend*.

El modelo LSTM fue el mejor modelo con un error MSE de 0.001596. En cuanto a la experiencia del usuario, esta fue satisfactoria, ya que la interfaz gráfica fue muy fácil de usar tanto como para el agricultor, como para el administrador.

Finalmente, se comprobó que el sistema de predicción logra cumplir con los objetivos planteados, permitiéndole al agricultor conocer una predicción a futuro de sus cultivos.

Como trabajo a futuro, se recomienda implementar un sistema de notificaciones, el cual alerte al agricultor cuando el cultivo tenga un factor ambiental no acorde a lo esperado.

**Palabras Clave:** Agricultura, Inteligencia artificial, *Machine Learning*, LSTM, ARIMA, Transformer.

## **ABSTRACT**

At this time, the lack of knowledge of the variation in temperature affects the decision-making of farmers, so they do not know when the ideal time is to sow, harvest and how much water to irrigate their crops because the temperature is an important factor in agriculture. Therefore, it is necessary to implement a 100% functional environmental data prediction system of a wireless sensor network in agricultural crops to facilitate decision-making for farmers.

Experimental tests were carried out with a temperature dataset from the INAMHI (National Institute of Meteorology and Hydrology) that served as a data source to make the predictions. Three artificial intelligence models ARIMA, LSTM and Transformer were evaluated. By varying their parameters and using different metrics to analyze their errors, the best model was selected to be implemented in the prediction system. Additionally, graphical interfaces were implemented so that the farmer and administrator can interact with the prediction system, for this the Angular framework was used for the frontend and the Django python framework for the backend.

The LSTM model was the best model with an MSE error of 0.001596. As for the user experience, it was satisfactory since the graphical interface was very easy to use for both the farmer and the administrator.

Finally, it was found that the prediction system achieves the stated objectives, allowing the farmer to know a future prediction of his crops.

As future work, it is recommended to implement a notification system, which alerts the farmer when the crop has an environmental factor that is not as expected.

**Keywords:** *Agriculture, Artificial Intelligence, Machine Learning, LSTM, ARIMA, Transformer.*

# ÍNDICE GENERAL

RESUMEN.....	I
<i>ABSTRACT</i> .....	II
ÍNDICE GENERAL.....	III
ABREVIATURAS.....	VI
SIMBOLOGÍA .....	VII
ÍNDICE DE FIGURAS .....	VIII
ÍNDICE DE TABLAS .....	XI
CAPÍTULO 1.....	12
1.    Introducción.....	12
1.1.    Descripción del problema.....	12
1.2.    Justificación del problema .....	14
1.3.    Objetivos .....	15
1.3.1.    Objetivo General .....	15
1.3.2.    Objetivos Específicos .....	15
1.4.    Descripción de los módulos del proyecto .....	15
1.4.1.    Módulo 1: Adquisición de datos .....	15
1.4.2.    Módulo 2: Evaluar modelos de predicción .....	16
1.4.3.    Módulo 3: Visualización de las predicciones .....	16
1.5.    Marco teórico .....	17
1.5.1.    Adquisición de datos.....	17
1.5.2.    Conjuntos de datos utilizados principalmente para el aprendizaje automático.....	17
1.5.3.    Predicción de datos en series temporales .....	18
1.5.4.    Visualizador Web .....	22
Grafana	23

InfluxDB 24

CAPÍTULO 2.....	26
2. Metodología.....	26
2.1. Análisis preliminar.....	26
2.1.1. Usuarios del sistema .....	27
2.2. Requerimientos.....	27
2.2.1. Funcionales.....	27
2.2.2. No funcionales .....	28
2.3. Adquisición de datos .....	28
2.4. Evaluación y selección del mejor modelo de predicción.....	30
2.4.1. Exploración de los modelos de predicción.....	30
2.4.2. Análisis exploratorio del dataset de temperatura.....	33
2.4.3. Definición de las librerías a usar para los modelos de predicción. ....	34
2.4.4. Selección del mejor modelo de predicción .....	34
2.5. Visualización de las predicciones .....	35
2.5.1. Modelo conceptual.....	36
2.5.2. Desarrollo de la pantalla para la visualización de las predicciones. ....	37
2.5.3. Desarrollo de la pantalla para la visualización de las predicciones vs. datos reales. ....	38
2.5.4. Desarrollo de la pantalla para la actualización del modelo de predicción. ....	40
CAPÍTULO 3.....	43
3. Resultados Y ANÁLISIS .....	43
3.1. Análisis exploratorio del dataset de temperatura .....	43
3.2. Evaluación de modelos .....	44
3.3. Evaluación del modelo final.....	53
3.4. Validación y resultados del sistema de predicción.....	53

3.4.1.	Pasos para la validación.....	54
3.4.2.	Resultados de la validación.....	55
3.5.	Costos.....	66
3.6.	Propuesta de implementación.....	67
CAPÍTULO 4.....		68
4.	Conclusiones Y Recomendaciones.....	68
4.1.	Conclusiones.....	68
4.2.	Recomendaciones.....	69
4.3.	Trabajos Futuros.....	69
BIBLIOGRAFÍA.....		70
ANEXOS.....		75
ANEXO A.....		75

## ABREVIATURAS

ESPOL	Escuela Superior Politécnica del Litoral
CIDIS	Centro de Investigación, Desarrollo e Innovación de Sistemas Computacionales
INAMHI	Instituto Nacional de Meteorología e Hidrología
FONAG	Fondo para la protección del Agua
ARIMA	Modelo autorregresivo integrado de media móvil
LSTM	Memoria a corto plazo
MSE	Error cuadrático medio
RMSE	Distancia media cuadrática mínima
MAE	Error absoluto medio
SQL	Lenguaje de consulta estructurado
RNN	Red neuronal recurrente
GRU	Unidad recurrente cerrada
IoT	Internet de las cosas
API	Interfaz de programación de aplicaciones
PIB	Producto interior bruto

## SIMBOLOGÍA

°C      Grados Centígrados

h      horas

t      tiempo

# ÍNDICE DE FIGURAS

Figura 1.1 Módulos generales del proyecto.....	13
Figura 1.2 Módulo de adquisición de datos .....	15
Figura 1.3 Módulo de evaluar modelos de predicción.....	16
Figura 1.4 Módulo de visualización de las predicciones.....	16
Figura 1.6 Arquitectura de una red LSTM [22].....	21
Figura 1.7 Arquitectura de una red GRU [24]. .....	22
Figura 2.1 Dataset de temperatura escogido [15].....	29
Figura 2.2 Temperatura vs. tiempo del <i>dataset</i> escogido [15]. .....	30
Figura 2.3 Comparación de LSTM con Transformer [39] .....	32
Figura 2.4 Diagrama de la base de datos de series de tiempo .....	33
Figura 2.5 Modelo conceptual del sistema.....	36
Figura 2.6 Visualización de pantalla de predicción .....	37
Figura 2.7 Pantalla de las predicciones vs. datos reales.....	39
Figura 2.8 Pantalla de reentrenamiento el modelo .....	40
Figura 2.9 Pantalla de reentrenamiento el modelo 2 .....	41
Figura 2.10 Pantalla de Pantalla para visualizar los modelos .....	41
Figura 3.1 Parámetros vs. métricas a evaluar .....	46
Figura 3.2 Predicción vs. temperatura real del modelo ARIMA.....	46
Figura 3.3 Hiperparámetros para el LSTM.....	49
Figura 3.4 Función de pérdida para el <i>dataset</i> de entrenamiento y prueba del LSTM .....	49
Figura 3.5 Predicción vs. Temperatura real del LSTM .....	50

Figura 3.6 Los mejores hiperparámetros para el modelo Transformer .....	52
Figura 3.7 Predicción vs. Temperatura real del modelo Transformer .....	53
Figura 3.8 Resultados de los mejores modelos .....	53
Figura 3.9 Resultados acerca de la tarea 1 del Agricultor .....	56
Figura 3.10 Resultados acerca de la tarea 2 del Agricultor .....	57
Figura 3.11 Resultados acerca de la tarea 3 del Agricultor .....	57
Figura 3.12 Resultados acerca de la facilidad de uso de la aplicación web Agricultor .....	58
Figura 3.13 Resultados acerca del diseño de la aplicación web Agricultor .....	59
Figura 3.14 Resultados acerca de entender las gráficas por el Agricultor .....	59
Figura 3.15 Resultados acerca de la tarea 1 del Administrador .....	60
Figura 3.16 Resultados acerca de la tarea 2 del Administrador .....	61
Figura 3.17 Resultados acerca de la tarea 3 del Administrador .....	61
Figura 3.18 Resultados acerca de la tarea 4 del Administrador .....	62
Figura 3.19 Resultados acerca de la tarea 5 de Administrador .....	63
Figura 3.20 Resultados acerca de la tarea 6 del Administrador .....	63
Figura 3.21 Resultados acerca de la tarea 7 del Administrador .....	64
Figura 3.22 Resultados acerca de la facilidad de uso de la aplicación web Administrador .....	65
Figura 3.23 Resultados acerca del diseño de la aplicación web Administrador .....	65
Figura 3.24 Resultados acerca de entender las gráficas por el Administrador .....	66
Figura 4.1 Formulario Administrador 1 .....	75
Figura 4.2 Formulario Administrador 2 .....	76
Figura 4.3 Formulario Administrador 3 .....	77

Figura 4.4 Formulario Administrador 4 .....	78
Figura 4.5 Formulario Administrador 5 .....	79
Figura 4.6 Formulario Agricultor 1 .....	80
Figura 4.7 Formulario Agricultor 2 .....	81
Figura 4.8 Formulario Agricultor 3 .....	82

## ÍNDICE DE TABLAS

Tabla 1 Estructura del dataset de temperatura.....	43
Tabla 3 Característica de los datos .....	43
Tabla 4 Histograma de frecuencias.....	44
Tabla 5 Test de estacionalidad.....	44
Tabla 6 Configuraciones del modelo ARIMA .....	45
Tabla 7 Resultados LSTM para el optimizador Adam .....	47
Tabla 8 Resultados LSTM para el optimizador RMSprop .....	47
Tabla 9 Resultados LSTM para el optimizador SGD .....	48
Tabla 10 Mejores configuraciones de hiperparámetros para el LSTM .....	48
Tabla 11 Resultado del Transformer para el optimizador Adam.....	50
Tabla 12 Resultado del Transformer para el optimizador RMSprop.....	51
Tabla 13 Resultado del Transformer para el optimizador SGD.....	51
Tabla 14 Mejores configuraciones de hiperparametros para el modelo Transformer	52
Tabla 15 Análisis de costos .....	66

# CAPÍTULO 1

## 1. INTRODUCCIÓN

En este capítulo se realizó la exploración acerca del problema que sufren los agricultores debido a las altas y bajas de temperatura, lo que afecta al rendimiento de los cultivos, por lo que abordando los objetivos del proyecto para que el agricultor cuente con las herramientas necesarias para prevenir la pérdida de los cultivos tales como la predicción de heladas y de temperaturas altas, se muestra la propuesta de un sistema de predicción que sea fácil de usar por los agricultores y les permita predecir la temperatura en su cultivo, lo cual ayudara al agricultor a tomar decisiones acertadas que puedan favorecer a una mejor producción, también se describen los módulos del proyecto que van a ser desarrollados en cada fase.

Para finalizar, en este capítulo se presenta la revisión de la literatura de cada componente importante del proyecto cómo es la adquisición de datos, la predicción de datos en series temporales y la visualización de las predicciones, en el cual se menciona las herramientas que se van a usar y los posibles modelos de inteligencia artificial.

### 1.1. Descripción del problema

En la actualidad la variación de la temperatura debido al cambio climático influye en la agricultura afectando el rendimiento de los cultivos y los ciclos del crecimiento de las especies agrícolas. Así mismo el cambio climático es un factor que favorece la aparición de plagas y enfermedades que pueden generar la muerte o un desarrollo menor al esperado de las plantas en los cultivos [1].

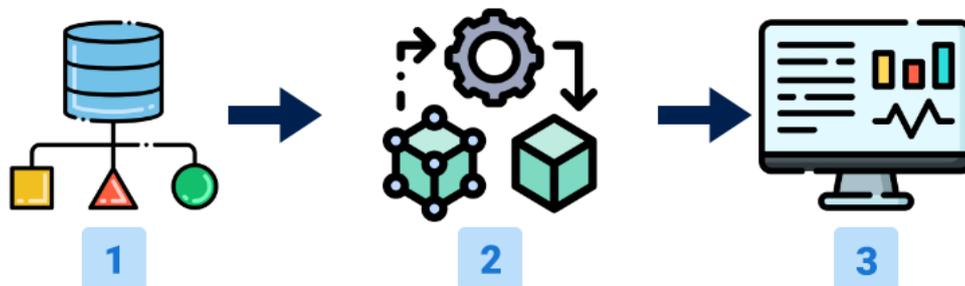
Se conoce que los pequeños y medianos agricultores no utilizan la inteligencia artificial para poder predecir las variaciones climáticas en sus cultivos, debido a que el sistema que usan es tradicional, en el cual los niveles de ingresos siguen siendo bajos y depende de la agricultura como principal fuente de sustento [2], esto se puede evidenciar en el estudio realizado en la ciudad de Milagro a 208 fincas el año 2020 el cual se propuso investigar las causas del muy bajo índice de uso de las tecnologías informáticas, demostrando que el 97,6 % no poseen ningún tipo de sistema automatizado para llevar el control agrícola. Entre las razones por que no disponen de un sistema automatizado

se revela que el 20.68 % desconoce de su existencia, el 33.65 % referencia que el mismo no se ajusta a sus necesidades, mientras que el 29.8 % por la gran dificultad o complejidad de su manejo [3].

Es por eso que, en el presente proyecto, se plantea utilizar un modelo de predicción que permita conocer previamente las variaciones de temperaturas en el sector agrícola mejorando la toma de decisiones y favoreciendo a una mejor producción.

Para desarrollar este proyecto se lo dividió en tres módulos principales, los cuales serán desarrollados en el siguiente orden:

1. Adquisición de datos
2. Evaluar los modelos de predicción
3. Visualización de las predicciones



**Figura 1.1 Módulos generales del proyecto.**

## 1.2. Justificación del problema

Actualmente, la agricultura se constituye como el cuarto sector más importante en la generación de ingresos para el Ecuador, este es uno de los ejes principales de la economía del país, ya que representa hasta un 8 % del PIB y es responsable de más 2,2 millones de puestos de trabajo [4], pero anualmente se pierden 150.542,20 hectáreas de cultivos y el 80 % de las pérdidas se concentran en el cacao, maíz duro seco, arroz, palma africana, café, maíz suave seco y plátano. Las causas de pérdida de cultivos son sequías, heladas, plagas, enfermedades, inundaciones entre otros [5].

Con lo anteriormente expuesto, es muy importante que el agricultor cuente con las herramientas necesarias para prevenir las pérdidas en los cultivos, tales como la predicción de heladas que es la disminución de la temperatura del aire a un valor igual o inferior al punto de congelación del agua 0 °C, cuando esto sucede se produce la sublimación del vapor de agua sobre hojas o tallos de plantas en forma de cubierta de hielo dañando a la planta interna o externamente. Sin embargo, algunos cultivos de origen tropical experimentan daños cuando están sometidos a temperaturas por debajo de 12,5 °C [6]. De la misma manera, las temperaturas altas crean condiciones para que las plantas experimenten cambios morfológicos, fisiológicos y bioquímicos, debido a que las temperaturas mayores a 40 °C pueden causar quemaduras en hojas y brotes jóvenes, senescencia foliar y abscisión, inhibición de brotes y decoloración de frutos, estos daños afectan gravemente al rendimiento de los cultivos [7]. Por lo que al no tener las herramientas necesarias el agricultor no sabe que plantas sembrar, debido a que existen variedades de plantas que son más resistentes al cambio de temperatura, si adelantar o retrasar la siembra, agregar más insumos (nutrientes, agua) para poder mantener el nivel de metabolismo en las plantas y evitar el daño.

Por esta razón, este proyecto tiene como propuesta la implementación de un sistema de predicción que sea fácil de usar por los agricultores y pueda predecir la temperatura en sus cultivos, mediante el uso de un modelo de predicción y una interfaz gráfica que le permita visualizar las predicciones, de tal forma que el agricultor pueda tomar decisiones acertadas las cuales favorecerán a una mejor producción.

Para finalizar, con este proyecto se aporta una base de nuevos sistemas de predicción para los cultivos en busca de mejorar la situación del agricultor en el Ecuador e incorporar la tecnología como fuente de innovación en la agricultura.

### 1.3. Objetivos

#### 1.3.1. Objetivo General

Implementar un sistema de predicción de datos ambientales de una red de sensores inalámbrica en cultivos agrícolas para facilitar la toma de decisiones a los agricultores.

#### 1.3.2. Objetivos Específicos

1. Adquirir datos de temperatura para el entrenamiento de los modelos de predicción.
2. Evaluar modelos que predigan la temperatura en un intervalo de tiempo a futuro.
3. Visualizar las predicciones de datos de temperatura en una interfaz gráfica.

### 1.4. Descripción de los módulos del proyecto

#### 1.4.1. Módulo 1: Adquisición de datos

- a. Explorar bancos de datos climáticos.
- b. Obtener el *dataset* apropiado de temperatura para entrenar el modelo.



Figura 1.2 Módulo de adquisición de datos

### 1.4.2. Módulo 2: Evaluar modelos de predicción

- a. Explorar modelos de regresión e inteligencia artificial.
- b. Realizar un análisis exploratorio del *dataset* de temperatura
- c. Evaluar los modelos de predicción de temperatura.



Figura 1.3 Módulo de evaluar modelos de predicción.

### 1.4.3. Módulo 3: Visualización de las predicciones

- a. Desarrollar la interfaz gráfica para la visualización de las predicciones.
- b. Visualizar los datos en series de tiempo mediante un gráfico lineal.
- c. Implementar funcionalidades para filtrar el rango de tiempo de la predicción.



Figura 1.4 Módulo de visualización de las predicciones

## **1.5. Marco teórico**

### **1.5.1. Adquisición de datos**

#### **Banco de datos**

Un banco de datos es un repositorio donde se almacenan conjuntos de datos o *datasets* pertenecientes al mismo contexto para que las personas los puedan usar, teniendo en cuenta que existen bancos de datos abiertos y cerrados, son ampliamente utilizados en entornos científicos y de investigación con el objetivo de almacenar la información experimental [8].

#### **Dataset**

Un *dataset* es un conjunto de datos tabulados de cualquier tipo, cada columna del *dataset* representa una variable y cada fila corresponde a cualquier dato que se está tratando, actualmente se han vuelto populares por la facilidad de acceso a un gran volumen información pública y privada para la construcción de modelos de Big Data [9].

### **1.5.2. Conjuntos de datos utilizados principalmente para el aprendizaje automático**

#### **Temperaturas diarias de Mauna Loa**

El Mauna Loa, uno de los dos volcanes que conforman el Parque Nacional de los Volcanes de Hawái, se eleva más de 3962 m sobre el nivel del mar y se extiende otros 12 802 m por debajo del mar y el lecho marino, con un clima principalmente cálido [10]. [11] Este *dataset* registra las temperaturas desde el 2014 hasta el 2018, el cual contiene la temperatura mínima, máxima y promedio durante los días de medición [11].

#### **Temperaturas diarias del Monte Washington**

El monte Washington ubicado en el Noroeste de Estados Unidos es el pico más alto con 1917 m. Es famoso por su tiempo peligrosamente errático. Tuvo un récord durante 76 años por tener la mayor ráfaga de viento en la superficie de la tierra de 372 km/h, su

clima es principalmente frío [12]. Este *dataset* registra las temperaturas desde el 2014 hasta el 2018, el cual contiene la temperatura mínima, máxima y promedio durante los días de medición [13].

### **Temperaturas por hora de Guayaquil**

Santiago de Guayaquil es la ciudad más poblada de la República del Ecuador, es un importante centro de comercio con influencia a nivel regional en el ámbito comercial, de finanzas, cultural, y de entretenimiento, es la capital de la provincia del Guayas, se localiza en la costa del Pacífico, en la región litoral de Ecuador, con un clima principalmente cálido [14].

Este *dataset* registra las temperaturas por cada hora de la estación de monitoreo del INAMHI (Instituto Nacional de Meteorología e Hidrología) en la ciudad de Guayaquil ubicado en la Universidad de Guayaquil en la estación M1271 Guayaquil (Facultad CCNN) [15].

### **1.5.3. Predicción de datos en series temporales**

#### **Modelo autorregresivo**

Es un modelo que usa combinación lineal de variables aleatorias del pasado para predecir datos de un tiempo dado del futuro. El nombre de autorregresión es porque se aplica la regresión de la variable contra sí mismo. Es utilizado mayormente en series de tiempo. Para utilizar el modelo la serie de tiempo tiene como requisito que sea estacionaria [16].

Un ejemplo de la ecuación del modelo es:

$$Y_t = c + \theta Y_{t-1} + \varepsilon_t$$

Donde  $Y_t$  la predicción del valor en el tiempo  $t$ ,  $c$  es una constante,  $Y_{t-1}$  es el valor de la predicción en el tiempo  $t-1$  multiplicado por el valor de  $\theta$  y  $\varepsilon_t$  es el error en el que varía en el tiempo  $t$ .

## **ARIMA**

El modelo autorregresivo integrado de promedio móvil (ARIMA por sus siglas en inglés), es un modelo estadístico utilizado para la predicción de datos en series temporales. Tiene tres componentes el autorregresivo, el integrado y medias móviles. El autorregresivo es el que utiliza los valores del pasado para realizar el modelo. El integrado observa las diferencias entre los valores estadísticos y los valores del pasado con la finalidad de que estos datos sean estacionarios, manteniendo, así la media, la varianza y la autocorrelación constante a través del tiempo. Y medias móviles es la dependencia entre el error y el valor observado con las observaciones del pasado [17].

## **Ventana deslizante**

Es un algoritmo para la predicción de datos utilizando una regresión. Su funcionamiento es que, dado un flujo de datos como una secuencia, se agrupa esa secuencia en un tamaño fijo de ventana. Este tamaño de la ventana se mantiene constante y cuando lleguen nuevos flujos de datos se eliminarán los antiguos. El desafío de este modelo es encontrar el perfecto tamaño de la ventana la cual la define el usuario. Tiene tres pasos principales. El primero es escoger el algoritmo que se quiere para la regresión. El segundo paso es escoger el tamaño óptimo de la ventana para usarla en el entrenamiento. Y el tercero es buscar el tamaño de ventana correcto para la predicción [18].

## **Red neuronal recurrente**

Una red neuronal recurrente (RNN por sus siglas en inglés), es capaz de procesar datos de tipo secuencias como: videos, conversaciones y texto, no se emplean para clasificar un dato, sino para generar una nueva secuencia. Su origen fue en la década de 1980, pero debido a los altos requerimientos en computación han sido complicadas de entrenar, hasta estos últimos años en donde hay mayor progreso computacionalmente.

Esta red usa la recurrencia para producir la salida y no solo usa la entrada actual sino también la que se generó previamente. Se puede decir que estas utilizan un cierto tipo de memoria para lograr el objetivo deseado [19]. Estas redes tienen un gran problema cuando tienen una larga secuencia de un tiempo "t" en el pasado, ya que a largo plazo se les dificulta aprender, esto se conoce como el "Desvanecimiento del gradiente".

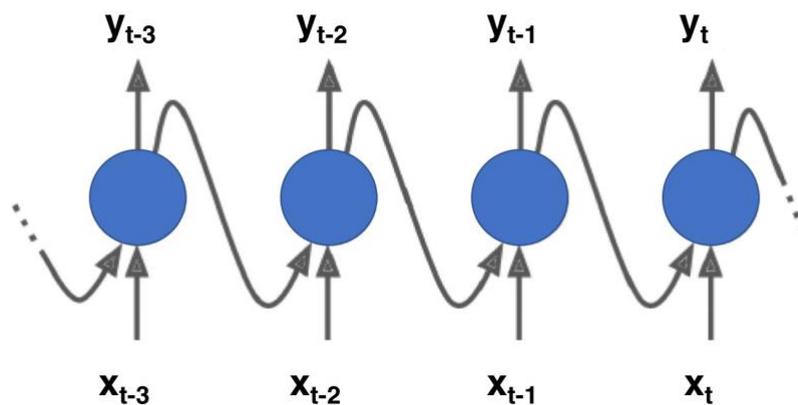


Figura 1.5 Arquitectura de una red recurrente [20]

En la Figura 1.5 se observa que la neurona necesita el estado de memoria del instante previo en el tiempo como parte de su entrada para el siguiente instante en el tiempo.

### Red neuronal con memoria a corto plazo extendida

La red neuronal con memoria a corto plazo extendida (LSTM por sus siglas en inglés) es una variación de una red recurrente solo que esta puede llevar información a lo largo del tiempo o espacios temporales quedándose con la información más importante o descartándola. Esta red resuelve el problema del desvanecimiento del gradiente que presenta la RNN añadiendo una célula de memoria y compuertas que permite el flujo de la información a lo largo de periodos extendidos de tiempo [21].

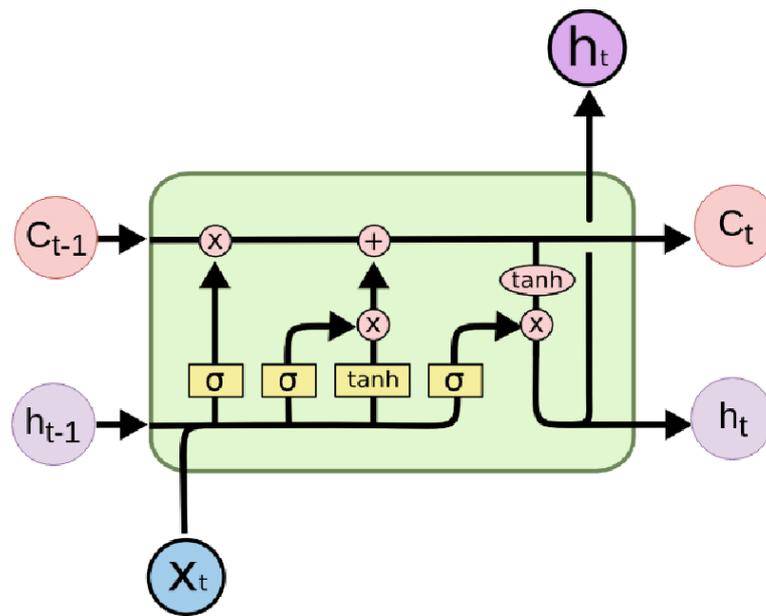


Figura 1.5 Arquitectura de una red LSTM [22]

En la Figura 1.3 se observa las diferentes puertas de una LSTM. Estas puertas se encargan de traspasar lo más importante a las iteraciones siguientes.

## GRU

Es una variación de una red recurrente que propone evitar el problema del desvanecimiento del gradiente. Esta red usa dos puertas una de actualización y otra de reinicio. La puerta de reinicio se encarga de la memoria a corto plazo y de determinar lo que no es importante y borrarlo. La puerta de actualización se encarga de la memoria a largo plazo y de decidir qué información se debe pasar a la siguiente iteración, se observa las diferentes puertas en la Figura 1.7 [23].

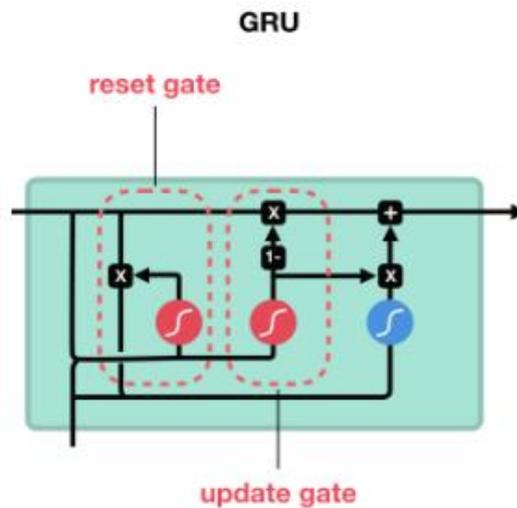


Figura 1.6 Arquitectura de una red GRU [24].

## Transformer

Es un tipo de red neuronal usado mayormente para el procesamiento de lenguaje natural. Este modelo transforma una secuencia de entrada a otra secuencia de salida como es el reconocimiento de voz o una traducción de un lenguaje a otro. Esta red complementa a los modelos de la RNN y las convolucionales logrando mejoras en precisión y eficiencia. Este modelo utiliza un concepto llamado “atención”, que es cuando se recibe una secuencia se decide en cada paso que otra parte de la secuencia es importante [25].

### 1.5.4. Visualizador Web

#### Interfaz de usuario

Una interfaz de usuario es el medio por el cual una persona controla una aplicación de software o dispositivo de hardware, esta interfaz puede incluir pantallas, textos, botones, gráficos, etc. Las interfaces de usuario más comunes son los sitios web, ya que están disponibles para los usuarios con solo acceder a la URL del sitio. Los elementos que conforman la interfaz de usuario van desde los controles de entrada que permiten introducir información en el sistema por parte de los usuarios, componentes de

navegación que ayudan a los usuarios a moverse por la interfaz [26], existen diferentes tipos las que pueden ser aplicaciones web, híbridas y nativas.

## **Aplicación web**

Las aplicaciones web se ejecutan por medio de un navegador, y no necesitan ser instaladas en el ordenador, debido a que los datos o archivos utilizados están almacenados en un servidor, de manera que el servidor envía la información a los dispositivos cuando es requerida, una de las principales ventajas es que no se necesita instalar la aplicación, ya que se accede a través de un navegador funcionando en diferentes plataformas y dispositivos [27], también se puede construir la aplicación web para que se adapte a diferentes pantallas los que se conoce como diseño adaptable.

Para construir aplicaciones web existen diferentes tecnologías, *frameworks* y librerías, como ReactJS, Angular y VueJS los cuales son los más usados, gracias a ellos se puede optimizar el tiempo de desarrollo, ya que evitan la repetición de código y se puede usar patrones que faciliten la legibilidad y mantenimiento [28].

## **Capa de acceso a datos *backend***

El *backend* es un componente de acceso a los datos de un software, donde el usuario normal no puede acceder, ya que contiene la lógica de la aplicación que maneja dichos datos [29]. Este se encarga de todos los procesos necesarios para que una aplicación de usuario funcione, ya que realiza cálculos y devuelve la información que se solicita.

## **Grafana**

Grafana es un software de visualización y análisis de código abierto, con el que se pueden consultar, visualizar y explorar los datos sin importar donde se encuentren, ya que proporciona herramientas para convertir los datos de series de tiempo en gráficos y visualizaciones. Las empresas lo usan para monitorear su infraestructura y análisis de registros, principalmente para mejorar su eficiencia operativa, porque se puede hacer un

seguimiento de los usuarios y los eventos, ya que cuenta con herramientas de automatización, recopilación, gestión y visualización de datos [30].

## **Prometheus**

Prometheus es un sistema de alerta y monitoreo de código abierto que recopila métricas y proporciona un lenguaje de consulta para explorar los datos, se utiliza con frecuencia para recopilar datos de clústeres de orquestación de contenedores como Kubernetes y OpenShift, ya que es el núcleo de un sólido sistema de monitoreo y alerta para clústeres, operadores y aplicaciones. Se usa a gran escala por las grandes organizaciones empresariales, pero es igualmente útil en proyectos pequeños debido a que recopila datos para aficionados [31].

Las principales características distintivas de Prometheus en comparación con otros sistemas de monitorización son: permite un modelo de datos multidimensional (series de tiempo definidas por el nombre de la métrica y el conjunto de dimensiones clave / valor) y posee un lenguaje de consulta flexible para aprovechar esta dimensionalidad [32].

## **InfluxDB**

InfluxDB es una base de datos de series de tiempo de código abierto, es decir cada dato está asociado con una marca de tiempo particular que puede ser fecha y hora, esta base de datos consta de mediciones que actúan como contenedor de etiquetas, campos y columnas de tiempo. Al comparar InfluxDB con SQL, InfluxDB está diseñado para almacenar un gran volumen de datos de series de tiempo, sin embargo, SQL también puede manejar datos de series de tiempo, pero no está estrictamente definido para este propósito [33].

## **QuestDB**

QuestDB es una base de datos SQL de código abierto para aplicaciones de servicios financieros, IoT, *machine learning*, DevOps y observabilidad, está diseñada para

procesar datos en series de tiempo y expone una API REST de alto rendimiento con compatibilidad con PostgreSQL [34].

Implementa ANSI SQL con extensiones nativas para funciones de lenguaje orientadas al tiempo, estas extensiones facilitan la correlación de datos de varias fuentes mediante uniones de series de tiempo y relacionales, por lo que puede lograr un alto rendimiento a partir de un modelo de almacenamiento orientado a columnas, una de las principales características es que permite hacer una prueba en vivo desde el navegador sin tener que instalar nada, en el cual se puede realizar consultas en un conjunto de datos de 1.600 millones de filas de 10 años de viajes en taxi en Nueva York [34].

En el siguiente capítulo se hablará de la metodología en donde se profundizará sobre el procedimiento que se utilizó para llevar a cabo el proyecto.

# CAPÍTULO 2

## 2. METODOLOGÍA

En este capítulo se muestra la propuesta de solución y se explora las diferentes herramientas y tecnologías que mejor se adaptan al proyecto, así mismo como el mejor *dataset* y la evaluación de los modelos de predicción para seleccionar el mejor, mediante la aplicación de métricas de error, el modelo seleccionado va a ser usado en el sistema de predicción.

Para finalizar en la interfaz gráfica se va a definir la integración de las nuevas pantallas de visualización con el proyecto integrador ya existente “Sistema integral de monitoreo de cultivos a partir de datos obtenidos mediante red de sensores inalámbricos georreferenciados”.

Además, este capítulo contiene los pasos que fueron seguidos para el diseño del sistema, los cuales son:

### 2.1. Análisis preliminar

Este proyecto es la continuación del proyecto integrador “Sistema integral de monitoreo de cultivos a partir de datos obtenidos mediante red de sensores inalámbricos georreferenciados” que fue realizado en el primero término del 2021 de la ESPOL, el cual cuenta con una red de sensores que guardan los registros en una base de datos InfluxDB, estos datos se pueden visualizar mediante una interfaz gráfica realizada en Angular que consume los servicios de un *backend* en Django, los datos para el entrenamiento del modelo se obtienen de los sensores de temperatura de los cultivos, sin embargo, como el sistema no se encuentra implementado en el campo se insertaron datos simulando un sensor de temperatura, luego se va a agregar al sistema existente el modelo de predicción y las pantallas de visualización.

### 2.1.1. Usuarios del sistema

- **Agricultor.** – Ingresa a la pantalla de visualización de las predicciones y selecciona el sensor a visualizar o el promedio de todos los sensores, también filtra las predicciones en diferentes períodos de tiempo y descarga los reportes en PDF.
- **Administrador.** – Ejecuta el entrenamiento del modelo con los nuevos datos y visualiza los datos predichos vs los datos reales.

## 2.2. Requerimientos

### 2.2.1. Funcionales

- **Agricultor: Observar las gráficas de las predicciones de temperatura**  
El agricultor ingresa a la pantalla de predicciones, la cual por defecto muestra las predicciones de temperatura de una semana del cultivo y finca seleccionados.
- **Agricultor: Cambiar el tiempo que se presentan las predicciones**  
El agricultor puede cambiar el tiempo de las predicciones, las cuales por defecto va a ser de una semana, por lo que podrá seleccionar un rango de fecha de las predicciones que desea observar, y podrá efectuar cambios en la escala de la visualización pudiendo seleccionar y visualizar las predicciones por hora o por día.
- **Agricultor: Cambiar de cultivo a visualizar las predicciones**  
El agricultor puede seleccionar en la pantalla de predicciones el cultivo y ver las predicciones correspondientes.
- **Agricultor: genera reporte de las predicciones observadas**

El agricultor puede generar reportes de las gráficas de las predicciones que incluyen los datos del cultivo, finca y el rango de tiempo, este reporte se podrá guardar en el dispositivo como un archivo PDF.

- **Administrador: Iniciar actualización del modelo**

El administrador realiza un reentrenamiento del modelo usando los datos almacenados en la base de datos, para esto selecciona el rango de fechas y reentrena el modelo, también pudiendo actualizar el modelo con un archivo y podrá descargar el modelo del cultivo como un archivo.

- **Administrador: Visualizar reportes de datos predichos**

El administrador visualiza reportes de datos predichos por modelo en un rango de tiempo pasado vs. los datos reales de los sensores, el rango de fecha que puede seleccionar es desde cuando se entrenó el modelo hasta el día actual.

### 2.2.2. No funcionales

- **Mantenibilidad:**

El sistema de predicción debe permitir ser modificado de manera efectiva y que los nuevos componentes agregados funcionen correctamente.

- **Accesibilidad:**

La pantalla de visualización debe permitir al agricultor navegar por sus componentes de una manera fácil y amigable.

### 2.3. Adquisición de datos

En esta sección se exploró diferentes bancos de datos o repositorios, como GitHub y el FONAG (Fondo para la protección del Agua), los cuales contenían *datasets* de variables climáticas y específicamente se buscó datos de temperatura, se encontró 3 *datasets* relevantes, 2 de GitHub y 1 del FONAG.

Los datases del GitHub contienen los registros de las temperaturas desde el 2014 hasta el 2018, los cuales constan de la temperatura mínima, máxima y promedio diario, los datos de temperatura son del Monte Whasintong y Mauna Loa, los cuales tienen un clima frío y cálido respectivamente.

Con respecto al *dataset* del FONAG, este contiene los registros de temperatura por hora, al extraer el *dataset* se seleccionaron las temperaturas que van desde el 1 de marzo del 2018 hasta el 1 de marzo del 2021, siendo en el transcurso de 3 años, con un total de 24753 registros de temperatura, la descarga del *dataset* se hizo mediante la página web del FONAG [15], por lo que también se pudo seleccionar la estación de observación de la ciudad de Guayaquil ubicada en la Universidad de Guayaquil con código M1271 Guayaquil (Facultad CCNN), estas mediciones fueron realizadas por el Instituto Nacional de Meteorología e Hidrología (INAMHI).

### Información Estaciones INAMHI

**Información:** Esta consulta muestra la información de las estaciones del INAMHI. Todos los campos son obligatorios para realizar la consulta

**Aviso:** El campo parametro depende de los campos estación y frecuencia

Estacion  
M1271 GUAYAQUIL (FACULTAD CCNN) ▾

Frecuencia  
horario ▾

Parametro  
TEMPERATURA AIRE PROMEDIO ▾

Fecha de Inicio  
01/03/2018

Fecha de Fin  
01/03/2021

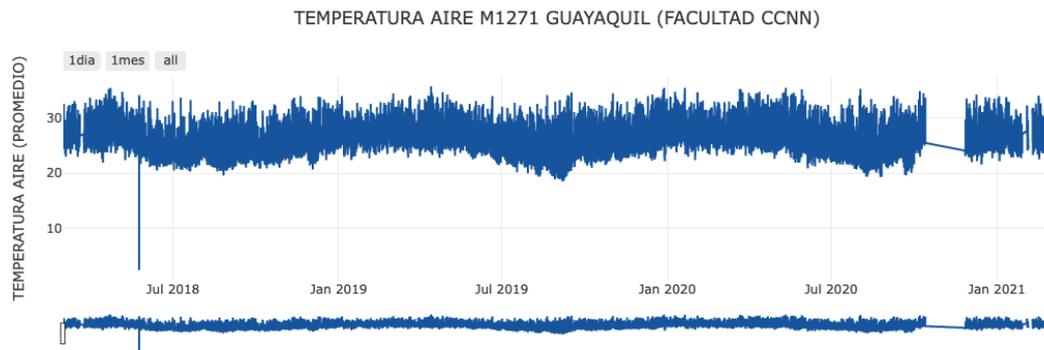
Buscar Exportar Limpiar

Código: M1271  
Nombre: GUAYAQUIL (FACULTAD CCNN)  
Tipo: METEOROLOGICA  
Latitud: -2.14650000  
Longitud: -79.91730000

Meteorológica  
Hidrológica

Leaflet | Map data © OpenStreetMap contributors, CC-BY-SA, Imagery © Mapbox

**Figura 2.1 Dataset de temperatura escogido [15].**



Nota: Los tiempos de respuesta y la información mostrada dependen del INAMHI

**Figura 2.2** Temperatura vs. tiempo del *dataset* escogido [15].

Para finalizar, el *dataset* seleccionado que va a ser usado en el proyecto, es el del FONAG de la ciudad de Guayaquil, debido a que este es más relevante porque está acorde al contexto del proyecto, el mismo que será implementado en un clima cálido y las mediciones se registran cada hora proporcionando más información, lo que permitirá realizar predicciones más precisas, porque la temperatura en el transcurso del día puede variar considerablemente.

## 2.4. Evaluación y selección del mejor modelo de predicción

Esta sección tiene como objetivo explorar los diferentes modelos de predicción, realizar un análisis exploratorio del *dataset*, para conocer sus características y estadísticos, finalmente se evalúa los modelos seleccionados mediante de cambio de sus hiperparámetros con las métricas de error MSE, RMSE y MAE para seleccionar el más apropiado a utilizar en la solución del proyecto.

### 2.4.1. Exploración de los modelos de predicción.

En esta sección se realizó la descripción de los modelos seleccionados que son el modelo ARIMA, LSTM Y Transformer.

#### ARIMA

Se eligió ARIMA por ser un modelo utilizado mucho en la estadística y servirá como *baseline* para los otros modelos. Además, es un modelo fácil de desarrollar comparado con LSTM y Transformer.

El modelo ARIMA se describe como la siguiente ecuación:

$$Y_t = \varphi_1 Y_{t-1} + \dots + \varphi_p Y_{t-p} + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \dots - \theta_q \varepsilon_{t-q}$$

Donde  $\varphi$  es el coeficiente autorregresivo que se determinó,  $\theta$  es el coeficiente de media móvil que se encontró,  $\varepsilon$  término de error y  $Y_{t-p}$  es la serie rezagada p periodos en el tiempo [35].

## LSTM

Se eligió LSTM porque es ampliamente usado en series de tiempo y en procesamiento natural del lenguaje [36]. Además, que este modelo soluciona el problema de desvanecimiento de gradiente que presentan las redes neuronales recurrentes. En este modelo la entrada de los datos debe estar en forma secuencial y debe tener un tamaño de *timestep*.

*Timestep* es un importante hiperparámetro para el LSTM, el cual es el tamaño de la secuencia que utiliza el modelo para predecir el valor siguiente. Un ejemplo de esto es una secuencia de números múltiplos de 10 con un *timestep* de 3 [37].

[10, 20, 30] = 40

[20, 30, 40] = 50

[30, 40, 50] = 60

Como se aprecia en el primer ejemplo dado, una secuencia de múltiplos de 10 con un *timestep* de 3 el modelo LSTM predecirá el valor siguiente que este caso es el 40. En el modelo propuesto tendrá un *timestep* de 168 que representa una semana, dado que

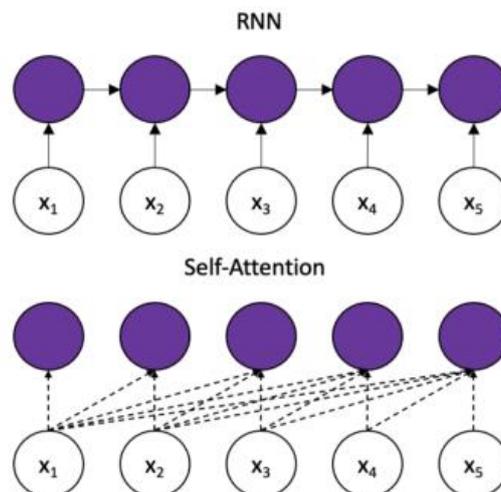
cada registro del *dataset* se almacena cada hora, por lo tanto, tenemos  $24 * 7 = 168$  [38]. Es decir que se necesitan 168 registros para predecir el registro siguiente.

Se usará 1 capa de LSTM como input y 2 capas ocultas de LSTM de 50 neuronas y una capa de salida con una sola neurona con la función de activación Lineal.

Como función de pérdida se utilizará el error cuadrático medio, que es el promedio de las distancias de los valores reales con los valores predichos y como optimizador se utilizará Adam.

## Transformer

Una de las desventajas de utilizar LSTM es que sufre de memoria de corto plazo en secuencias muy largas, por eso se escogió el modelo Transformer que a comparación con la LSTM no ocupa la ocurrencia y usa un concepto que se llama la atención.



**Figura 2.3 Comparación de LSTM con Transformer [39]**

Como se observa en la Figura 2.3, Transformer utiliza conexiones directas en todos los tiempos ya sean anteriores o a futuro, permitiendo que las secuencias sean más largas que utilizando una LSTM [39].

La arquitectura del modelo que se usó es de un transformador que consta de 3 bloques: Auto-atención enmascarado, codificador-decodificador y *feed forward*. Como optimizador se usó Adam.

### 2.4.2. Análisis exploratorio del dataset de temperatura

En esta sección se realizó un análisis exploratorio con el objetivo de estudiar el *dataset*, mediante estadísticas que permiten identificar las características de los datos tales como: mediana, media, máximo, mínimo, desviación estándar y la frecuencia mediante un histograma, se usó RStudio para el análisis, ya que con su lenguaje R permite una mayor velocidad en el tratamiento y la manipulación de los datos al trabajar con grandes volúmenes, además esta herramienta es de software libre [40].

Los datos están ordenados desde la fecha más reciente hasta la fecha más antigua, también la frecuencia de los datos es por día y por hora es decir van a haber 24 observaciones por cada día, por lo que este *dataset* contiene un total de 24753 registros de temperatura de los 3 años seleccionados, el *dataset* va a ser separado en un 70 % para el entrenamiento y un 30 % para la validación de los modelos, cabe mencionar que para ingresar los datos a los tres modelos se realizó un procesamiento previo que es el remplazar los valores aberrantes y nulos con el promedio de las temperaturas de sus vecinos, que son valores de temperatura una hora antes y después.

También, para poder integrar el sistema de predicciones con el existente, se insertó el *dataset* de temperatura en InfluxDB en el formato de la tabla del sensor de temperatura como se muestra en la Figura 2.4.

Temperatura	
planta	string
id_sensor	string
finca	string
usuario	string
valor	float
mínimo	float
máximo	float
latitud	float
longitud	float
timestamp	time

**Figura 2.4 Diagrama de la base de datos de series de tiempo**

### **2.4.3. Definición de las librerías a usar para los modelos de predicción.**

En esta sección se utilizó los tres modelos ARIMA, LSTM y Transformer, para poder generar estos modelos se usó el lenguaje de programación *python*.

Para generar el modelo ARIMA se usó la librería *statsmodels* debido a que proporciona las herramientas para poder realizar análisis de series de tiempo. Además, esta librería posee una clase llamada ARIMA que con el método `fit()` permite estimar un modelo de acuerdo con los datos disponibles, y, en cambio, `predict()` permite predecir los valores futuros.

Para generar el modelo LSTM se utilizó la librería Keras, debido a que permite crear diferentes arquitecturas de redes y así poder configurar dicha arquitectura para pronosticar de manera más precisa, para esto se usó la clase LSTM de la librería.

Finalmente, para generar el último modelo el modelo Transformer se usó la librería *pytorch*, debido a su adaptabilidad para el tema de Transformers con la clase *nn.Transformer*.

El entorno de desarrollo para estos modelos es Google Colab (*Colaboratory*), esta herramienta es un producto de *Google Research* que permite escribir y ejecutar código de *python* en el navegador, lo que facilita el desarrollo, ya que no se debe configurar el entorno de trabajo previamente y permite el trabajo colaborativo remotamente.

### **2.4.4. Selección del mejor modelo de predicción**

Para seleccionar el mejor modelo se usaron las métricas, error cuadrático medio MSE, distancia media cuadrática mínima RMSE y el error absoluto medio MAE, las cuales mide el error entre los datos predichos y los valores conocidos, para esto se usó el 30 % de *dataset* de validación, por lo que también se configuraron los modelos con diferentes hiperparámetro y el modelo que tuvo el menor error se seleccionó para ser usado en el sistema de predicción.

Finalmente, al integrar el proyecto con el sistema existente se agregó el modelo seleccionado por cada finca y cultivo registrados, por lo que se crean funciones que permiten hacer consultas de predicciones en un rango de fecha y modificar el

modelo. Para hacer disponible la API hacia la interfaz gráfica, se agregaron servicios que permiten hacer consultas de las predicciones, comparaciones de los datos predichos con los reales y actualizar el modelo.

## **2.5. Visualización de las predicciones**

Esta sección tiene como objetivo mostrar el modelo conceptual a usar en la implementación de la solución, por lo que se define las herramientas y librerías a usar en el *backend* y *frontend*. También se muestra los bosquejos de las pantallas a realizar en la interfaz gráfica explicando cuáles van a ser sus componentes y funcionalidad.

2.5.1. Modelo conceptual

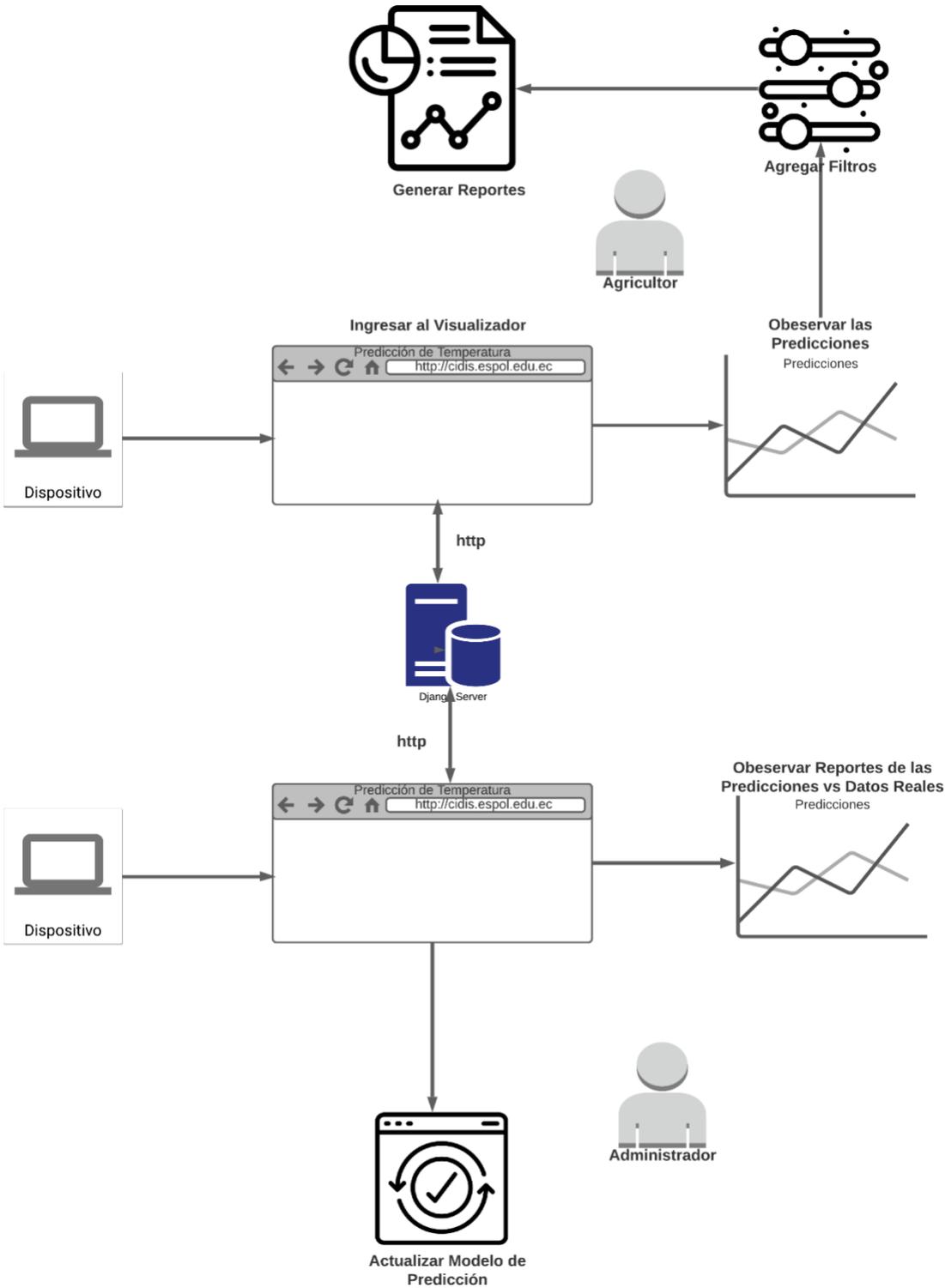
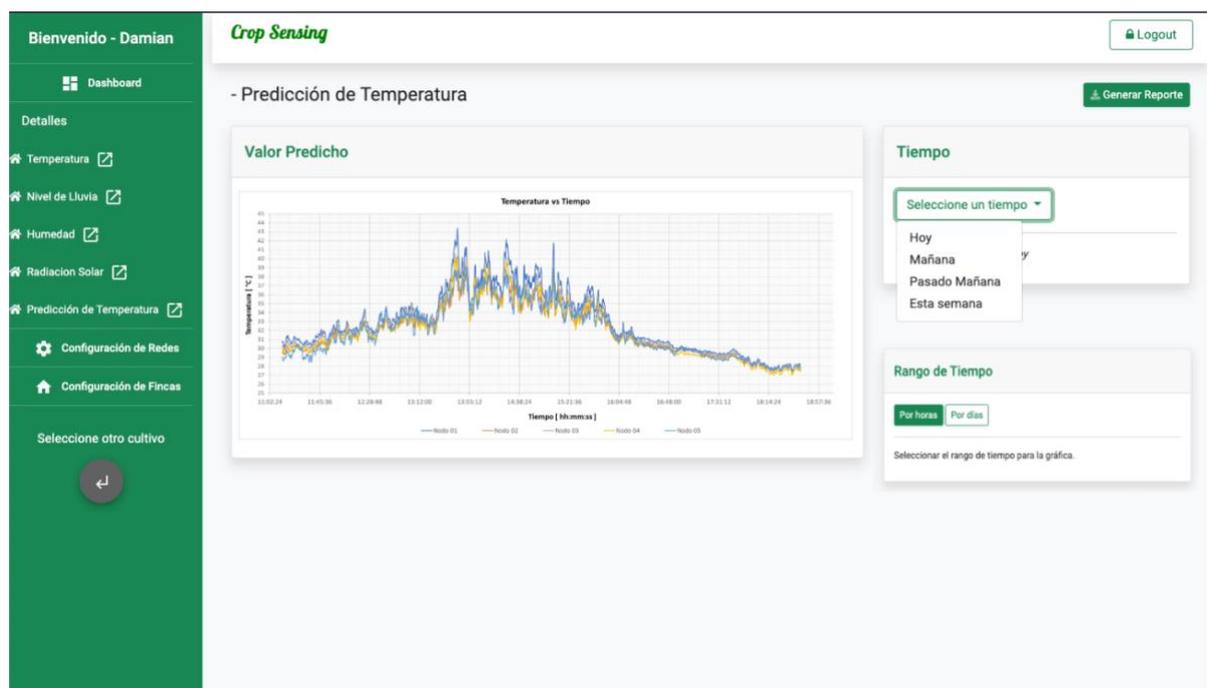


Figura 2.5 Modelo conceptual del sistema

Con base en los usuarios del sistema se obtuvo el siguiente modelo conceptual, que muestran los pasos que el agricultor y administrador realizan en la interfaz para observar las predicciones con sus diferentes filtros y generar los reportes, en el caso del administrador el reentrenamiento del modelo y visualizar los reportes de datos predicho con los reales, la transferencia de los datos con la interfaz gráfica y el servidor se realiza mediante peticiones HTTP, por lo que el navegador envía una solicitud para poder mostrar los datos de temperatura reales y predichos, esta solicitud será procesada por el *backend* tomando los datos de la base de datos y retornando al navegador esta información en un formato JSON.

## 2.5.2. Desarrollo de la pantalla para la visualización de las predicciones.



**Figura 2.6 Visualización de pantalla de predicción**

En esta pantalla el agricultor observa las predicciones de temperatura mediante un gráfico lineal de series de tiempo tiempos vs. temperatura, en el eje (y) se muestra el día u hora de las predicciones, mientras que en el eje (x) el valor de temperatura predicho.

El Agricultor puede interactuar en la pantalla con tres elementos, el botón (Generara Reporte) que le permite descargar las predicciones con la información del cultivo en un archivo PDF, este archivo se guardará automáticamente en el dispositivo con el nombre de la fecha de visualización y el ID del cultivo, sin embargo, si se encuentra en una computadora esto será manejado por gestor de descargas del sistema por lo cual al momento de guardar el agricultor podrá cambiar el nombre del archivo como considere.

El seleccionador del tiempo (Seleccione un tiempo) permite elegir el rango de tiempo a futuro de las predicciones de temperatura al seleccionar un rango y confirmar la predicción, la gráfica de predicción de temperatura va a cambiar con los valores de predicción de la selección.

Por último, tiene disponible los botones de escala de tiempo Por hora y Por días, que le permite cambiar el rango o escala de la gráfica, en el caso de seleccionar por horas las predicciones se muestran por hora y en el caso de seleccionar por días se muestra promedio diario de las predicciones de temperatura en el rango de tiempo seleccionado.

### **2.5.3. Desarrollo de la pantalla para la visualización de las predicciones vs. datos reales.**

En esta pantalla el administrador observa las predicciones de temperatura mediante un gráfico lineal de series de tiempo tiempos vs. temperatura, en el eje (y) se muestra el día u hora de las predicciones, mientras que en el eje (x) el valor de temperatura predicho, también se incluirá una gráfica superpuesta de los datos reales de temperatura en el mismo periodo de tiempo.



**Figura 2.7 Pantalla de las predicciones vs. datos reales**

El Administrador puede interactuar en la pantalla con tres elementos, el seleccionador de tiempo (Seleccione un tiempo) que permite cambiar el rango tiempo de las predicciones de temperatura pasadas y los datos reales de temperatura en el mismo periodo, al seleccionar un rango y confirmar la gráfica de predicción de temperatura va a cambiar con los valores predichos y reales seleccionados.

Las gráficas de las temperaturas predichas y reales se encuentran superpuestas, lo que permite observar las diferencias de las temperaturas reales que fueron registradas por los sensores con las que proporciona el modelo de predicción.

Al igual que el agricultor el administrador tiene disponible los botones escala de tiempo Por hora y Por días, que le permite cambiar la escala de la gráfica, en el caso de seleccionar por horas las predicciones se muestran por hora y en el caso de seleccionar por días se muestra el promedio diario de las predicciones de temperatura en el tiempo seleccionado.

Por último, el administrador también puede descargar los reportes en PDF, solamente que en estos reportes se mostraran los datos reales con los predichos e información adicional del modelo como el cultivo y finca al que pertenece.

#### 2.5.4. Desarrollo de la pantalla para la actualización del modelo de predicción.

En esta pantalla el administrador realiza un reentrenamiento del modelo usando los datos almacenados en la base de datos, por lo que selecciona el rango de fechas y luego reentrena el modelo.

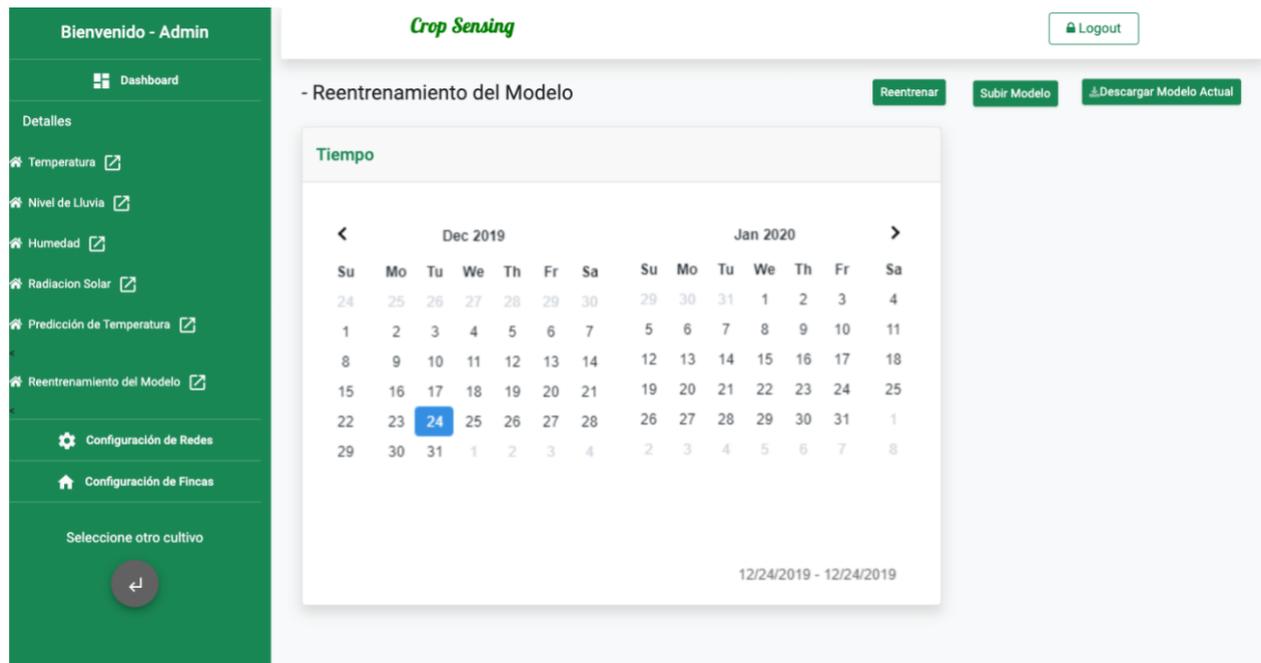
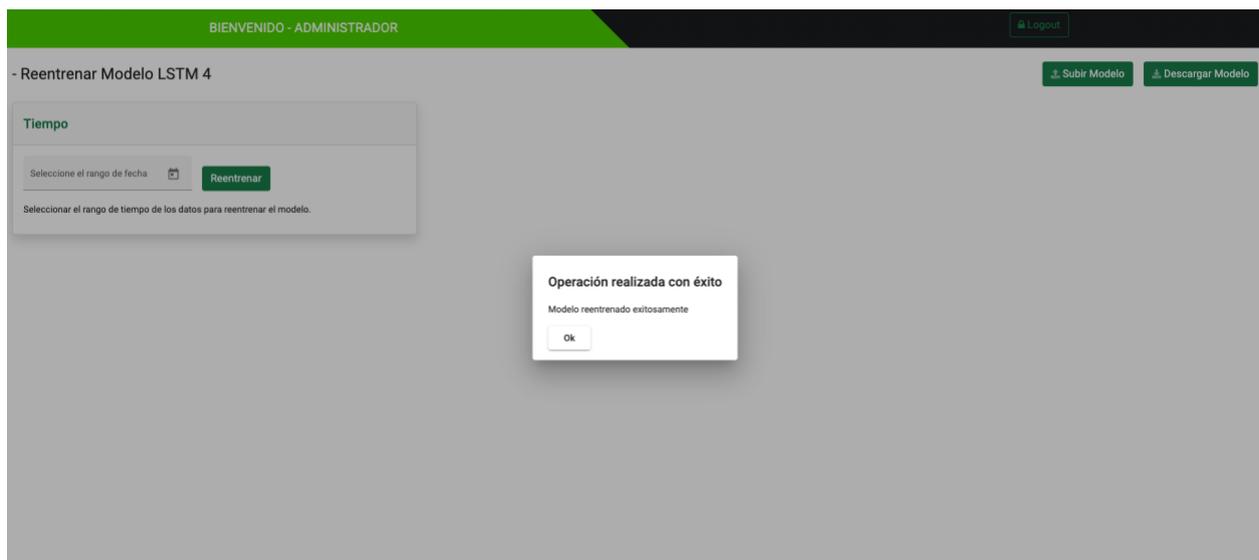


Figura 2.8 Pantalla de reentrenamiento el modelo

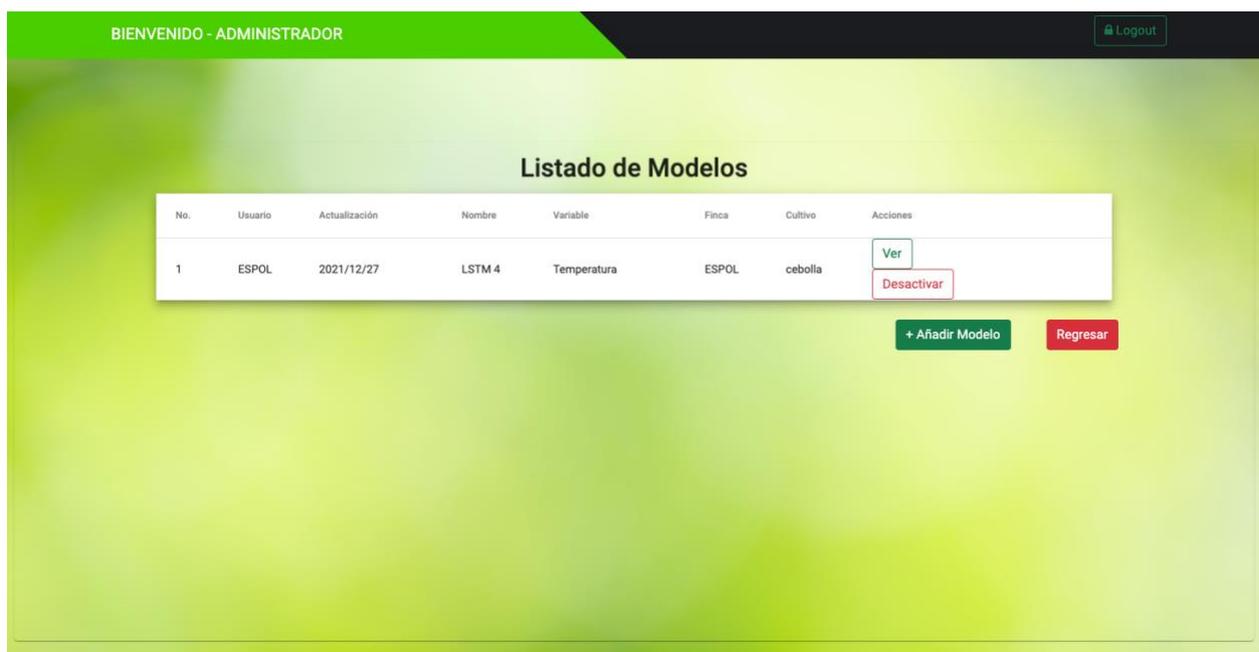
El botón (Reentrenar) permite comenzar el reentrenamiento, al finalizar el entrenamiento del nuevo modelo se muestra un diálogo con el mensaje de sí se realizó con éxito u ocurrió un error como se ve en la Figura 2.9.

El botón (Subir Modelo) permitirá reemplazar el modelo actual con un modelo desde un archivo.

También, tiene un botón (Descargar Modelo) que le permite guardar el modelo como un archivo el dispositivo.



**Figura 2.9 Pantalla de reentrenamiento el modelo 2**



**Figura 2.10 Pantalla de Pantalla para visualizar los modelos**

Finalmente, como se observa en la Figura 2.10 se implementó una pantalla para que el administrador pueda agregar, activar, desactivar y visualizar los modelos de predicción que están listados por el usuario, cultivo y finca al que pertenecen.

En el siguiente capítulo se muestran los resultados obtenidos de la propuesta a la solución, junto con un análisis de los mismos, se hace un énfasis en la selección del

modelo que obtuvo mejores resultados y en el análisis de usabilidad de la interfaz gráfica para el agricultor y administrador.

# CAPÍTULO 3

## 3. RESULTADOS Y ANÁLISIS

En este capítulo se muestran los resultados y análisis de los tres módulos que se llevaron a cabo para implementar nuestra solución. Primero se menciona el análisis del *dataset* obtenido del INAMHI, la evaluación de los 3 modelos de predicción ARIMA, LSTM y Transformer para su consiguiente elección, finalmente se muestra el análisis del uso de la interfaz gráfica con los usuarios administrador y agricultor.

### 3.1. Análisis exploratorio del dataset de temperatura

En esta sección se muestra el análisis del *dataset* obtenido del INAMHI, así como sus valores estadísticos.

**Tabla 1 Estructura del dataset de temperatura**

fecha	valor
2021-03-01 23:00:00	25.5
2021-03-01 22:00:00	25.6
2021-03-01 21:00:00	25.5
2021-03-01 20:00:00	25.6
2021-03-01 19:00:00	26.1

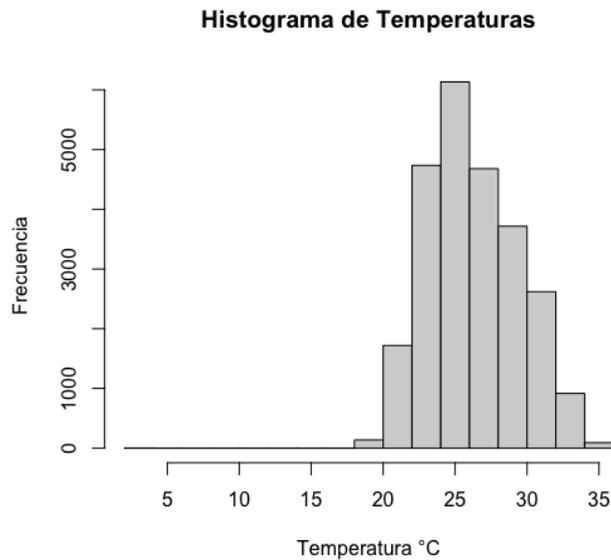
En la Tabla 2 se observa la estructura del *dataset* que está compuesta de dos columnas, la fecha y el valor de la temperatura en ese instante de tiempo.

**Tabla 2 Característica de los datos**

Característica	Resultado
Mínimo	2.40
Mediana	25.90
Promedio	26.31
Máximo	35.80
Nulos	2
Desviación estándar	3.14

En la Tabla 3 se observa los principales estadísticos del *dataset* como son el mínimo, mediana, promedio, máximo, desviación estándar y valores nulos.

**Tabla 3 Histograma de frecuencias**



En la Tabla 4 se aprecia la frecuencia de los datos de temperatura notamos que el promedio es 26.31 y posibles valores aberrantes.

### 3.2. Evaluación de modelos

En esta sección se presentan los resultados de la evaluación de los modelos escogidos para así seleccionar el mejor modelo de inteligencia artificial para integrarlo al sistema de predicción.

### ARIMA

**Tabla 4 Test de estacionalidad**

Test	p-value
adfuller	0
KPSS	0.01

Para aplicar el modelo ARIMA primero se necesita que la serie de tiempo a usar sea estacionaria.

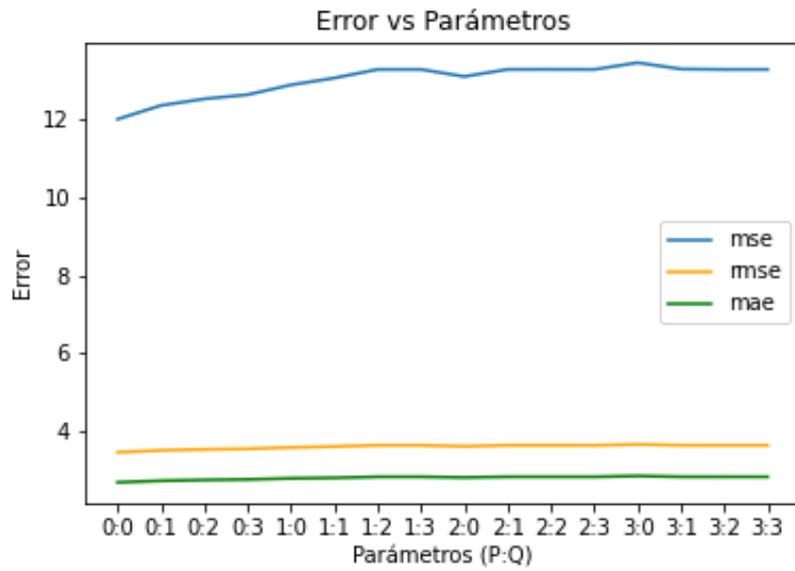
En la Tabla 5 se observa los resultados de la prueba de estacionalidad dando como p-value menor a 0.05 se puede rechazar la hipótesis nula y de esta manera concluyendo que la serie es estacionaria.

Ya conociendo que este es una serie estacional se utilizó el modelo estacional de ARIMA SARIMAX (P, D, Q, s), por lo que con los hiperparámetros que se generaron los diferentes modelos de ARIMA fueron P y Q, por lo que se mantuvo el orden de integración del proceso  $D=0$ , ya que la serie es estacionaria y no necesita ser diferenciada, por otro lado, la periodicidad  $s=24$ , puesto que los datos son por hora y queremos las predicciones diarias.

**Tabla 5 Configuraciones del modelo ARIMA**

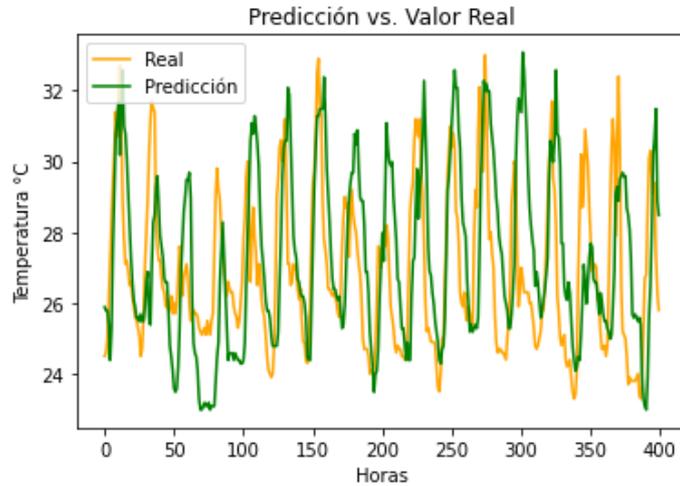
P	Q	MSE	RMSE	MAE
0	0	11.999158	3.463980	2.695714
0	1	12.350638	3.514348	2.735605
0	2	12.518904	3.538206	2.755240
0	3	12.626074	3.553319	2.768725
1	0	12.877207	3.588483	2.796899
1	1	13.053109	3.612909	2.810071
1	2	13.273673	3.643305	2.836778
1	3	13.273362	3.643263	2.836750
2	0	13.092617	3.618372	2.820742
2	1	13.273950	3.643343	2.836804
2	2	13.275655	3.643577	2.836846
2	3	13.273179	3.643237	2.836675
3	0	13.445948	3.666872	2.857909
3	1	13.284244	3.644756	2.837763
3	2	13.273360	3.643262	2.836696
3	3	13.273049	3.643220	2.836622

En la Tabla 6 tenemos las diferentes configuraciones del modelo ARIMA con las respectivas métricas de los errores que se evalúa, observando que la mejor configuración es la primera fila siendo  $P=0$  y  $Q=0$ .



**Figura 3.1 Parámetros vs. métricas a evaluar**

En la Figura 3.1 se observa que P y Q = 0 es la mejor configuración.



**Figura 3.2 Predicción vs. temperatura real del modelo ARIMA**

En la Figura 3.2 se observa la predicción de la temperatura con el valor real en un rango específico de horas.

## LSTM

En este modelo los hiperparámetros que se variaron fueron el optimizador, el tamaño del *batch* y la cantidad de épocas.

**Tabla 6 Resultados LSTM para el optimizador Adam**

batch_size	epochs	learning	Optimizer	MSE	RMSE	MAE
		rate				
60	15	0.0001000	Adam	0.001983	0.04453	0.03366
40	15	0.0001000	Adam	0.0017	0.04172	0.03089
50	15	0.0001000	Adam	0.001596	0.0399	0.02877
50	20	0.0001000	Adam	0.001636	0.04044	0.0295
80	12	0.0001000	Adam	0.00256	0.0506	0.03819

Para el optimizador Adam sus resultados se reflejan en la Tabla 7, quedando como mejor resultado la configuración de hiperparámetros mostrada en la tercera fila.

**Tabla 7 Resultados LSTM para el optimizador RMSprop**

batch_size	epochs	learning rate	Optimizer	MSE	RMSE	MAE
60	15	0.0001000	RMSprop	0.001663	0.04078	0.030435
40	15	0.0001000	RMSprop	0.00205	0.0452	0.03411
50	15	0.0001000	RMSprop	0.001600	0.0400	0.0294
50	20	0.0001000	RMSprop	0.001927	0.0439	0.03314
80	12	0.0001000	RMSprop	0.001701	0.0412	0.02939

Así mismo, para el optimizador RMSprop los resultados se reflejan en la Tabla 8 quedando como mejor resultado la configuración de hiperparámetros mostrada en la tercera fila.

**Tabla 8 Resultados LSTM para el optimizador SGD**

batch_size	epochs	learning		Optimizer	MSE	RMSE	MAE
		rate					
60	15	0.0001000		SGD	0.00355	0.18874	0.16391
40	15	0.0001000		SGD	0.026033	0.161349	0.14005
50	15	0.0001000		SGD	0.0313	0.1770	0.15187
50	20	0.0001000		SGD	0.02355	0.15347	0.1336
80	12	0.0001000		SGD	0.03036	0.1742	0.1492

Para el optimizador SGD los resultados se reflejan en la Tabla 9, quedando como mejor resultado la configuración de hiperparámetros mostrada en la cuarta fila.

**Tabla 9 Mejores configuraciones de hiperparámetros para el LSTM**

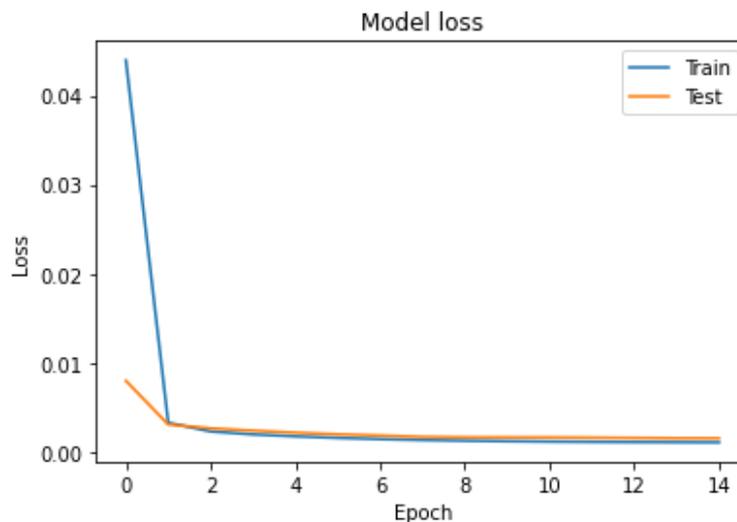
batch_size	epochs	learning		Optimizer	MSE	RMSE	MAE
		rate					
50	15	0.0001000		Adam	0.001596	0.0399	0.02877
50	15	0.0001000		RMSprop	0.001600	0.0400	0.0294
50	20	0.0001000		SGD	0.02355	0.15347	0.1336

Finalmente, de las mejores configuraciones de hiperparámetros por optimizador como se ve en la Tabla 10, se elige la primera fila que es la mejor configuración para el modelo LSTM.

```
params = {
    "batch_size": 50,
    "epochs": 15,
    "lr": 0.00010000,
    "time_steps": 168,
    "optimizer": Adam
}
```

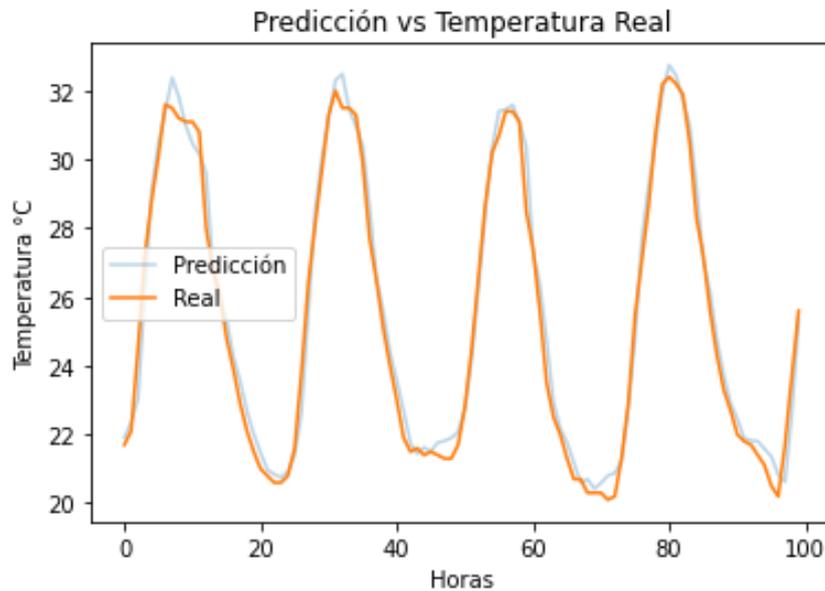
**Figura 3.3** Hiperparámetros para el LSTM

En la Figura 3.3 se muestra la mejor configuración con un tamaño del *batch* de 50, con una cantidad de épocas de 15, una tasa de aprendizaje de 0.0001, con un *timestep* de 168 y con el optimizador Adam.



**Figura 3.4** Función de pérdida para el *dataset* de entrenamiento y prueba del LSTM

En la Figura 3.4 se observa que a medida que avanza el número de épocas tanto como para el *dataset* de entrenamiento y como el de test la función de pérdida disminuye.



**Figura 3.5 Predicción vs. Temperatura real del LSTM**

En la Figura 3.5 se observa la predicción del *dataset* de prueba para 4 días, en donde se puede notar una buena predicción.

### Transformer

En este modelo los hiperparámetros que se variaron fueron el optimizador y la ventana de predicción.

**Tabla 10 Resultado del Transformer para el optimizador Adam**

batch_size	epochs	learning		Optimizer	MSE	RMSE	MAE
		rate	forecast_window				
1	1000	0.001	24	Adam	2.411	1.552	1,0036
1	1000	0.001	48	Adam	5,255	2,29	1,303
1	1000	0.001	72	Adam	2.533	1,5916	1,2001
1	1000	0.001	96	Adam	3.504	1.87	1,0955
1	1000	0.001	120	Adam	1.076	1,0376	0,8901

En la Tabla 11 se observa la variación de los hiperparámetros dando las respectivas métricas, la última es la mejor configuración para el optimizador Adam.

**Tabla 11 Resultado del Transformer para el optimizador RMSprop**

batch_size	epochs	learning		Optimizer	MSE	RMSE	MAE
		rate	forecast_window				
1	1000	0.001	24	<u>RMSprop</u>	0.4836	0.6954	0.6831
1	1000	0.001	48	<u>RMSprop</u>	1,633	1,2779	0,287
1	1000	0.001	72	<u>RMSprop</u>	1,5577	1,248	0.9360
1	1000	0.001	96	<u>RMSprop</u>	1,2143	1,101	0.7216
1	1000	0.001	120	<u>RMSprop</u>	1,7277	1,314	1,0323

En la Tabla 12 se observa que la cuarta fila es la mejor configuración para el optimizador RMSprop.

**Tabla 12 Resultado del Transformer para el optimizador SGD**

batch_size	epochs	learning		Optimizer	MSE	RMSE	MAE
		rate	forecast_window				
1	1000	0.001	24	<u>SGD</u>	1	1	1
1	1000	0.001	48	<u>SGD</u>	1	1	1
1	1000	0.001	72	<u>SGD</u>	2	2	0.9057
1	1000	0.001	96	<u>SGD</u>	1	1,0761	1
1	1000	0.001	120	<u>SGD</u>	2	2	1

En la Tabla 13 se observa que la cuarta fila es la mejor configuración para el optimizador SGD.

**Tabla 13 Mejores configuraciones de hiperparametros para el modelo Transformer**

batch_size	epochs	learning			MSE	RMSE	MAE
		rate	forecast_window	Optimizer			
1	1000	0.001	120	Adam	1.076	1	0,8901
1	1000	0.001	96	<u>RMSprop</u>	1	1,101	0.7216
1	1000	0.001	96	<u>SGD</u>	1	1,0761	1

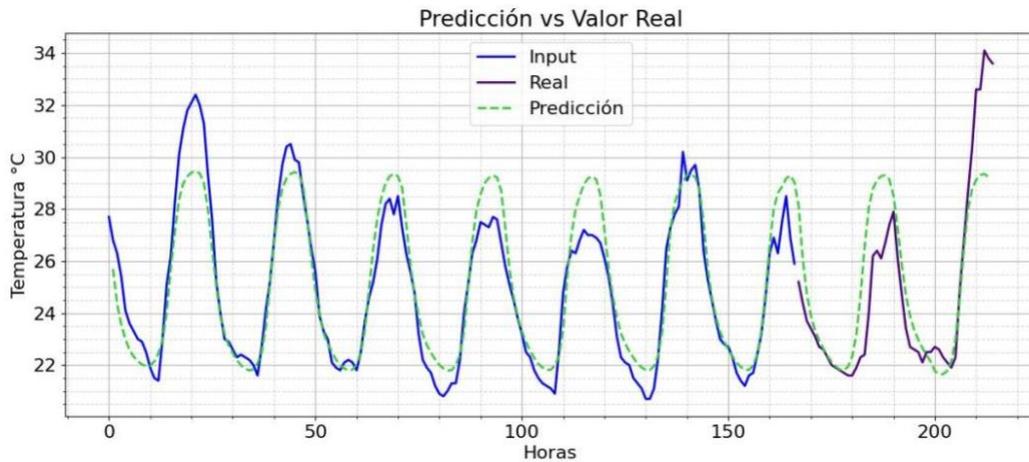
Finalmente, de las mejores configuraciones de hiperparámetros por optimizador como se ve en la Tabla 14, se elige la última configuración que es la mejor para el modelo Transformer.



```
params = {  
  "batch_size": 1,  
  "epochs": 1000,  
  "lr": 0.001,  
  "forecast_window": 48,  
  "optimizer": SGD  
}
```

**Figura 3.6 Los mejores hiperparámetros para el modelo Transformer**

Como se observa en la Figura 3.6, esos son los mejores hiperparámetros para el modelo Transformer con un tamaño de batch de 1, la cantidad de épocas 1000, tasa de aprendizaje de 0.001, la ventana de predicción de 48 y con el optimizador SGD.



**Figura 3.7 Predicción vs. Temperatura real del modelo Transformer**

En la Figura 3.7 se observa la predicción del *dataset* de prueba en un tiempo de 200 horas.

### 3.3. Evaluación del modelo final

Dada las evaluaciones que se realizaron a los modelos, se escogieron las mejores configuraciones que menor error dieron.

Modelo	MSE	RMSE	MAE
LSTM	0.001596	0.0399	0.02877
Transformer	0.777	0.8815	0.8571
ARIMA	11.999158	3.463980	2.695714

**Figura 3.8 Resultados de los mejores modelos**

Como se observa en la Figura 3.8 el mejor modelo de predicción es el LSTM, dado que tiene el menor error comparado con los otros modelos.

### 3.4. Validación y resultados del sistema de predicción

En esta sección se describe las experiencias de los usuarios acerca del producto realizado (aplicación web) esto con el fin de verificar que el software cumple con los

requerimientos estipulados en la sección 2.2 por lo que se realizó pruebas con usuarios reales para que nos ofrezcan una retroalimentación sobre su experiencia.

Las pruebas fueron realizadas de la siguiente manera:

Se hicieron entrevistas a 6 usuarios de los cuales 3 son para perfiles de agricultor y 3 para administrador en un rango de edad de 20 a 35 años, se dio acceso a la aplicación a cada usuario para que pueda probar su funcionamiento y se observó si los usuarios realizaron las tareas asignadas con éxito.

Los usuarios con el perfil de agricultor deben tener un conocimiento bajo acerca del funcionamiento de la web y uso del computador, además también deben conocer como ingresar a un sitio web con clave y contraseña.

Los usuarios con el perfil de administrador deben tener un conocimiento medio acerca del funcionamiento de la web y del computador, además también deben conocer como subir y descargar archivos a la web.

Se desarrolló un formulario que permite analizar la experiencia de los usuarios en la aplicación, posteriormente se hizo un análisis del cuestionario para conocer aspectos positivos y negativos de la solución.

#### **3.4.1. Pasos para la validación**

Para realizar las entrevistas se siguieron los siguientes pasos:

1. Se mostró la aplicación web a los usuarios y se dio una explicación acerca de su funcionamiento donde se detallaron las secciones que posee, se proporcionó un usuario y contraseña que está configurado dependiendo el perfil del usuario.
2. En el caso del administrador se le otorgó un enlace para que pueda acceder a la aplicación web, ya que la entrevista se hizo en línea, en el caso del agricultor se le proporcionó un dispositivo físico para que pueda probar.
3. Cuando los usuarios ya ingresaron al sitio, se les solicitó que efectúen 3 tareas al agricultor y 7 tareas al administrador.

Agricultor:

1. Ingresar al cultivo cebolla y seleccionar el parámetro de predicción de temperatura para la finca ESPOL.
2. Seleccionar un rango de tiempo para visualizar las predicciones del parámetro seleccionado anteriormente.
3. Descargar el reporte de las predicciones visualizadas.

Administrador:

1. Crear un nuevo modelo de predicción para el cultivo cebolla y finca Teresita.
  2. Seleccionar un rango de tiempo para visualizar la comparación de predicciones con los datos reales.
  3. Editar nombre del modelo de predicción.
  4. Descargar el reporte de la comparación de predicciones con los datos reales visualizados.
  5. Reentrenar el modelo mediante un rango de fecha seleccionado.
  6. Descargar el modelo de predicción.
  7. Reentrenar el modelo con la subida de un archivo de extensión h5.
4. En el transcurso de la entrevista se respondieron las dudas de los usuarios acerca de la aplicación.
  5. Al final de la entrevista, se le proporcionó un enlace al formulario de Google Forms para obtener la retroalimentación de cada usuario, las entrevistas fueron realizadas anónimamente para respetar la privacidad de las personas, además de realizar preguntas sobre la dificultad de las tareas se preguntó sobre aspectos generales de la aplicación, como su facilidad de uso, el diseño y acerca de las gráficas presentadas, para realizar la encuesta se utilizó la escala de likert que permite medir el grado de conformidad de los usuarios con las diferentes tareas realizadas.

El contenido de las entrevistas se encuentra en la sección de ANEXO A.

### **3.4.2. Resultados de la validación**

En esta sección se detalla los resultados de las tareas realizadas por los usuarios.

## Agricultor:

### Tarea 1: Ingresar al cultivo cebolla y seleccionar el parámetro de predicción de temperatura para la finca ESPOL.

En la Figura 3.9 se muestra que al 66.7 % de los usuarios les resultó muy fácil ejecutar la tarea, mientras que al 33.3 % les fue fácil, esto se debe a que a los usuarios le tomó un tiempo en encontrar la sección “Predicción de Temperatura”.

¿Qué tan difícil fue realizar la tarea 1?

3 respuestas

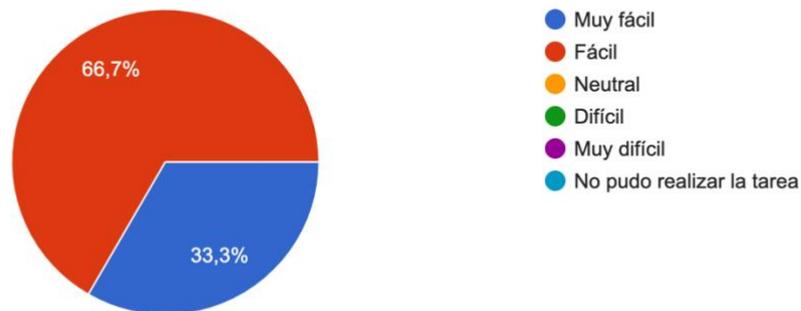


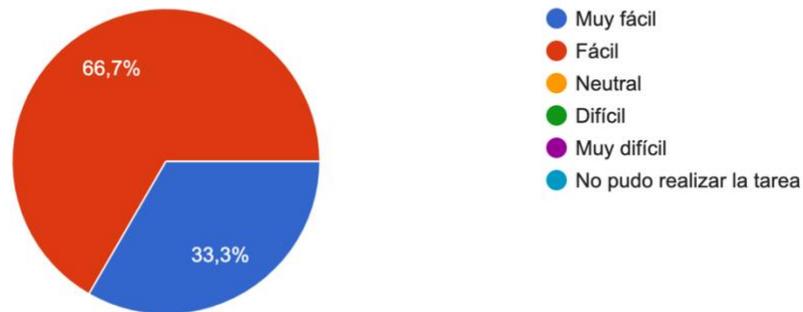
Figura 3.9 Resultados acerca de la tarea 1 del Agricultor

### Tarea 2: Seleccionar un rango de tiempo para visualizar las predicciones del parámetro seleccionado anteriormente.

En la tarea 2, podemos observar en la Figura 3.10, se muestra que al 66.7 % de los usuarios les resultó muy fácil realizar la tarea, mientras que al 33.3 % les resultó fácil.

¿Qué tan difícil fue realizar la tarea 2?

3 respuestas



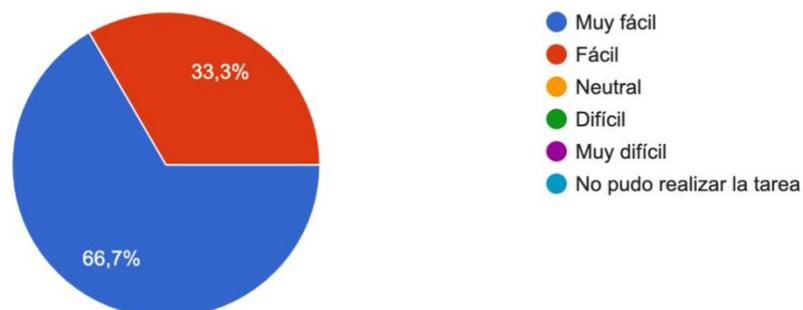
**Figura 3.10 Resultados acerca de la tarea 2 del Agricultor**

### Tarea 3: Descargar el reporte de las predicciones visualizadas.

En la tarea 3, se aprecia en la Figura 3.11, se muestra que al 66.7 % de los usuarios les resultó muy fácil realizar la tarea, mientras que al 33.3 % les resultó fácil.

¿Qué tan difícil fue realizar la tarea 3?

3 respuestas



**Figura 3.11 Resultados acerca de la tarea 3 del Agricultor**

### ¿Qué tan fácil ha sido usar la aplicación web?

En esta pregunta se observa que tan fácil ha sido utilizar la aplicación, de manera general a los agricultores les parece fácil de usar, pero ocurrieron algunos problemas con la experiencia de usuario como que les tomó un poco más de tiempo encontrar las secciones.

¿Qué tan fácil ha sido usar la aplicación web?

3 respuestas

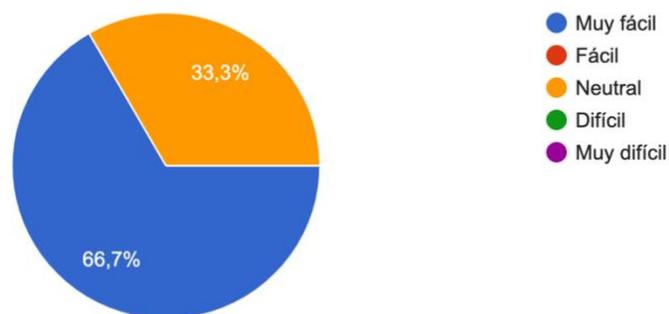


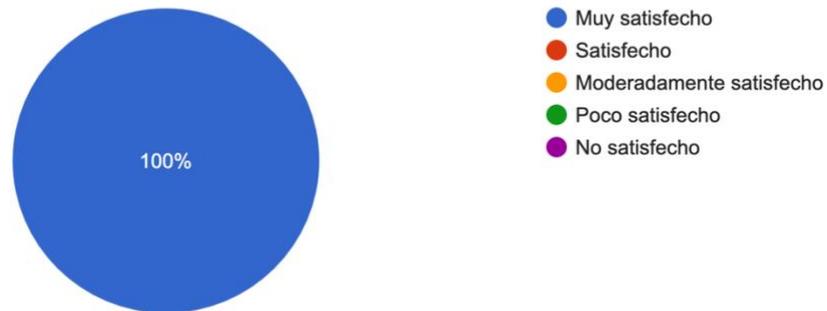
Figura 3.12 Resultados acerca de la facilidad de uso de la aplicación web Agricultor

### ¿Qué tan satisfecho se siente con el diseño de la aplicación web?

También se preguntó acerca del diseño de la aplicación y todos los agricultores coincidieron que el diseño es adecuado, debido a que el sistema de predicción es algo novedoso para ellos.

¿Qué tan satisfecho se siente con el diseño de la aplicación web?

3 respuestas



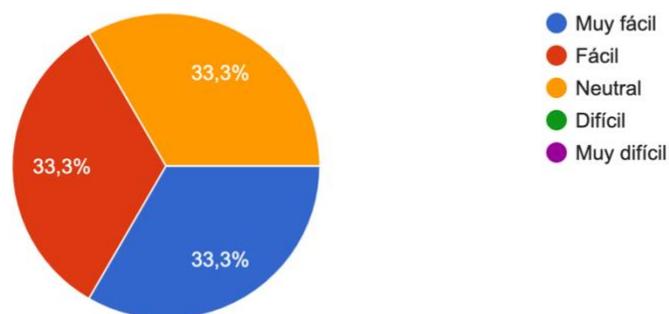
**Figura 3.13 Resultados acerca del diseño de la aplicación web Agricultor**

¿Qué tan fácil fue entender las gráficas visualizadas?

Finalmente, se preguntó sobre qué tan fácil es interpretar las gráficas visualizadas en la aplicación web, se observó que para el 66.7 % le ha sido fácil interpretar, mientras que para el 33.3 % les pareció neutral.

¿Qué tan fácil fue entender las gráficas visualizadas?

3 respuestas



**Figura 3.14 Resultados acerca de entender las gráficas por el Agricultor**

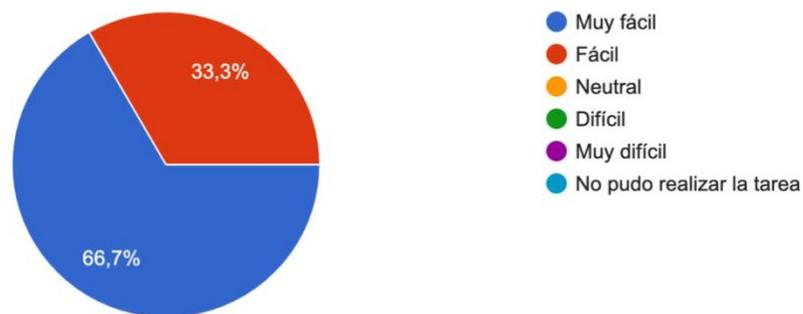
**Administrador:**

**Tarea 1: Crear un nuevo modelo de predicción para el cultivo cebolla y finca Teresita.**

En la Figura 3.15 se muestra que al 66.7 % de los usuarios les resultó muy fácil realizar la tarea, mientras que al 33.3 % les fue fácil crear un nuevo modelo, esta diferencia fue, ya que a un usuario le tomó un poco más de tiempo encontrar el botón “Administrar modelos” en la pantalla principal.

¿Qué tan difícil fue realizar la tarea 1?

3 respuestas



**Figura 3.15 Resultados acerca de la tarea 1 del Administrador**

**Tarea 2: Seleccionar un rango de tiempo para visualizar la comparación de predicciones con los datos reales.**

En la tarea 2, se aprecia en la Figura 3.16 que para el 66.7 % les resultó fácil mientras que al 33.3 % les resultó difícil esta tarea, esto se debe a que a estos usuarios le tomó un tiempo ingresar al modelo.

¿Qué tan difícil fue realizar la tarea 2?

3 respuestas

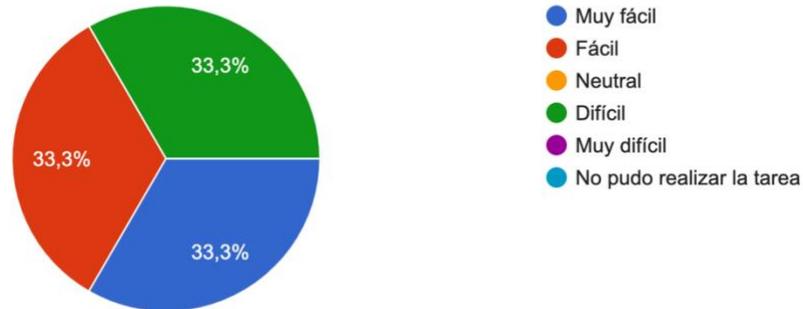


Figura 3.16 Resultados acerca de la tarea 2 del Administrador

### Tarea 3: Editar nombre del modelo de predicción.

En la tarea 3, permite observar en la Figura 3.17 que para el 66.7 % les resultó muy fácil mientras que al 33.3 % les resultó fácil esta tarea.

¿Qué tan difícil fue realizar la tarea 3?

3 respuestas

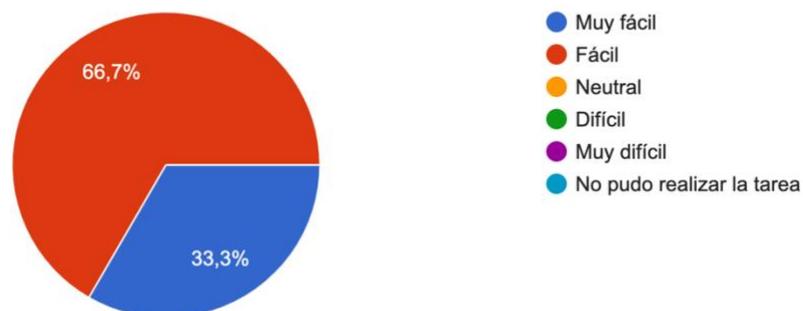


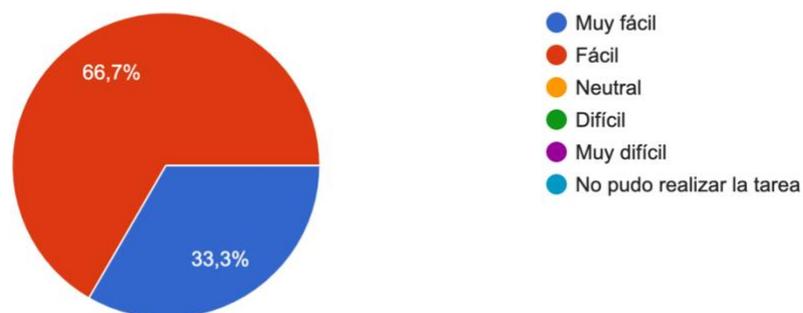
Figura 3.17 Resultados acerca de la tarea 3 del Administrador

#### **Tarea 4: Descargar el reporte de la comparación de predicciones con los datos reales visualizados.**

En la tarea 4, se observa en la Figura 3.18 que para el 66.7 % les resultó muy fácil mientras que al 33.3 % les resultó fácil esta tarea, esto se debe porque a un usuario se le recargó la página y ya no existían un rango de tiempo para descargar el reporte, por lo que ahora el botón de descargar solo está disponible cuando exista un rango de fecha seleccionado.

¿Qué tan difícil fue realizar la tarea 4?

3 respuestas



**Figura 3.18 Resultados acerca de la tarea 4 del Administrador**

#### **Tarea 5: Reentrenar el modelo mediante un rango de fecha seleccionadas.**

En la tarea 5, se puede observar en la Figura 3.19 que para el 66.7 % les resultó fácil mientras que al 33.3 % les resultó neutral esta tarea, esto se debe a que a un usuario se le permitió elegir el día actual en el rango por lo que dio un error, ya que aún no se obtienen los datos del día completo, por lo que se cambió la lógica de la selección y solo se permitió elegir desde el día anterior.

¿Qué tan difícil fue realizar la tarea 5?

3 respuestas



Figura 3.19 Resultados acerca de la tarea 5 de Administrador

### Tarea 6: Descargar el modelo de predicción.

En la tarea 6, podemos observar en la Figura 3.20 que para el 66.7 % les resultó muy fácil mientras que al 33.3 % les resultó fácil esta tarea.

¿Qué tan difícil fue realizar la tarea 6?

3 respuestas

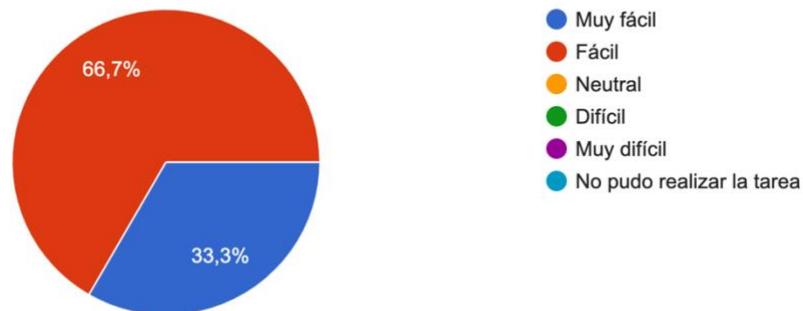
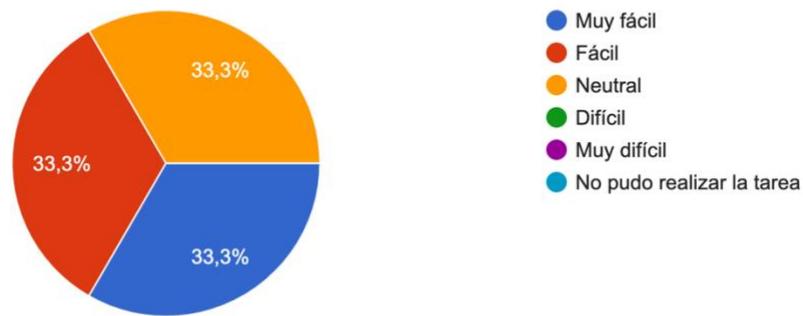


Figura 3.20 Resultados acerca de la tarea 6 del Administrador

### Tarea 7: Reentrenar el modelo con la subida de un archivo de extensión h5.

En la tarea 7, podemos observar en la Figura 3.21 que para el 66.7 % les resultó fácil mientras que al 33.3 % les resultó neutral esta tarea.

¿Qué tan difícil fue realizar la tarea 7?  
3 respuestas



**Figura 3.21 Resultados acerca de la tarea 7 del Administrador**

**¿Qué tan fácil ha sido usar la aplicación web?**

En esta pregunta se observa que tan fácil ha sido utilizar la aplicación, de manera general la aplicación es fácil de usar, pero ocurrieron algunos problemas con la experiencia de usuario que fueron solucionados.

¿Qué tan fácil ha sido usar la aplicación web?

3 respuestas

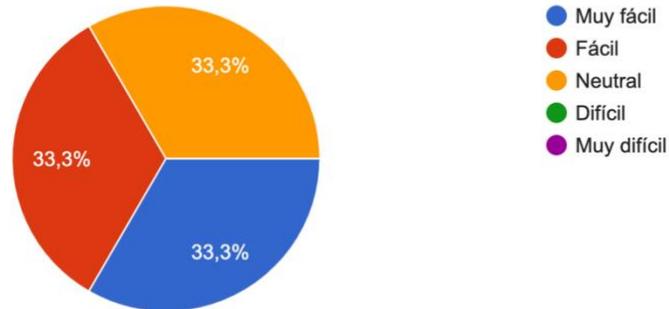


Figura 3.22 Resultados acerca de la facilidad de uso de la aplicación web Administrador

¿Qué tan satisfecho se siente con el diseño de la aplicación web?

También se preguntó acerca del diseño de la aplicación y a la mayoría de los usuarios coincidieron que el diseño es adecuado, pero se necesitaron ajuste con la posición y estado de los botones.

¿Qué tan satisfecho se siente con el diseño de la aplicación web?

3 respuestas

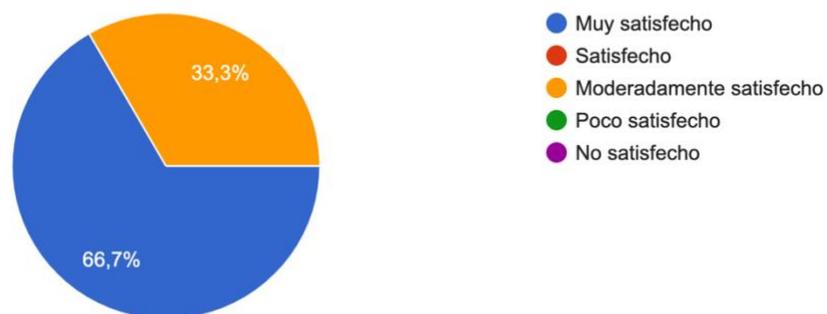
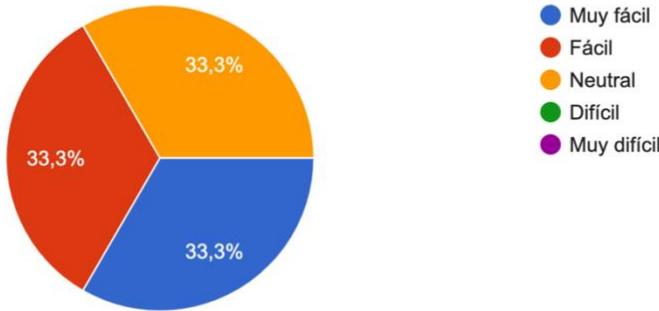


Figura 3.23 Resultados acerca del diseño de la aplicación web Administrador

¿Qué tan fácil fue entender las gráficas visualizadas?

Finalmente, se preguntó sobre qué tan fácil es interpretar las gráficas visualizadas en la aplicación web, se observó que para el 66.7 % le ha sido fácil interpretar, mientras que para el 33.3 % mencionó que se puede mejorar las gráficas para que sean más fácil de interpretar y entender.

¿Qué tan fácil fue entender las gráficas visualizadas?  
3 respuestas



**Figura 3.24 Resultados acerca de entender las gráficas por el Administrador**

**3.5. Costos**

A continuación, se detalla el costo del desarrollo del proyecto, para lo cual solamente se toma en cuenta el costo humano de desarrollo, para calcular el precio por hora se utilizó como referencia el salario promedio mensual de un desarrollador de software en la ciudad de Quito que es de \$1.500 dólares americanos mensuales [41].

**Tabla 14 Análisis de costos**

Desarrollo del sistema de predicciones			
Ítem	Descripción	Precio Hora	Precio Total
1	Desarrollo Frontend (Angular)	\$10	\$300
2	Desarrollo Backend (Django)	\$10	\$400
Total			\$700

Con respecto al software usado este fue totalmente open-source por lo que no tendrá un costo adicional al desarrollo del proyecto, también para el costo de despliegue en los servidores, esto no se toma en cuenta, debido a que los componentes del proyecto

*backend* y *frontend* van a ser implementados en los servidores del CIDIS (Centro de Investigación, Desarrollo e Innovación de Sistemas Computacionales) dentro de la ESPOL.

### **3.6. Propuesta de implementación**

Para la implementación del sistema de predicción se recomienda mantener una arquitectura monolítica, el cual con los dos componentes *backend* y *frontend* deben ser alojados en un servidor local, al contar con acceso a los servidores del CIDIS ubicados en la ESPOL, antes de empezar el despliegue del sistema se recomienda instalar todas las dependencias necesarias y levantar los servicios que se van a utilizar, como el contenedor de Docker con InfluxDB, nodejs y *python*, para que el sistema esté siempre disponible se necesita un servidor que esté conectado a la red y operativo todo el tiempo.

Para obtener más información acerca de la estructura de los servicios y levantar los servicios necesarios se debe consultar la guía de despliegue del sistema de predicción *Crop Sensing* en el CIDIS.

En el último capítulo se muestran las conclusiones y recomendaciones de los resultados obtenidos en este y finalmente se mencionan los trabajos futuros que se podrían agregar en la solución realizada.

# CAPÍTULO 4

## 4. CONCLUSIONES Y RECOMENDACIONES

En esta sección se presentan las respectivas conclusiones de los resultados obtenidos del proyecto. Además, se dan recomendaciones de cómo se debería mejorar en el sistema de predicción y mejoras a futuro.

### 4.1. Conclusiones

- De acuerdo con el objetivo general del proyecto se ha logrado desarrollar un sistema de predicción que ayuda a los agricultores a conocer la temperatura a futuro de sus cultivos para que puedan tomar decisiones apropiadas en los mismos. Además, mediante el análisis de la usabilidad de la interfaz gráfica podemos decir que es fácil de usar para el agricultor y administrador.
- Con base en la evaluación de los datos obtenidos de las distintas fuentes, se concluye que INAMHI proporcione datos más acordes con las necesidades del proyecto.
- Entre los modelos de predicción de temperatura evaluados, se encontró que el mejor fue la red LSTM obteniendo un error MSE de 0.001596.
- Algunos agricultores presentaron cierta dificultad al ingresar a la pantalla de predicción, ya que les tomo un poco más del tiempo identificar el botón apropiado, pero el inconveniente desaprecio después del segundo intento. Por lo tanto, se concluye que algunos agricultores, les va a tomar un poco más de tiempo entender el funcionamiento del sistema de predicción.
- Se realizaron las respectivas pantallas para el agricultor permitiéndole consultar las predicciones a un tiempo futuro de los cultivos y fincas registrados.
- Se realizó las respectivas pantallas para el administrador permitiéndole consultar, reentrenar y comparar los datos de predicción de los modelos existentes con los datos reales dado un periodo de tiempo.

## **4.2. Recomendaciones**

- Se recomienda tener una gran cantidad de datos para entrenar el modelo, porque entre más datos alimenten al modelo este proporcionara una mejor predicción.
- Para que exista compatibilidad en el sistema, se recomienda que todas las funcionalidades implementadas estén disponibles y todos los componentes estén en línea, con esto se evitara que se produzcan errores.
- Se recomienda tener conocimientos básicos sobre contenedores y el uso de Docker para que al momento de despliegue de InfluxDB no se presenten dificultades.

## **4.3. Trabajos Futuros**

- Dado el trabajo realizado, se recomienda en versiones posteriores implementar un sistema de notificaciones, el cual alerte al agricultor cuando el cultivo tendrá un factor ambiental no acorde a lo esperado.
- Para una mejora a futuro, se puede hacer que el sistema recomiende una acción a proceder al agricultor cuando un factor ambiental del cultivo no cumple con el rango normal esperado.
- Para versiones futuras, analizar las acciones que toma el agricultor con respecto a las recomendaciones que se ofrecen con el objetivo de mejorarlas, ya que el agricultor registrara si estas fueron beneficiosas o no para el cultivo.
- Automatizar los procesos en los cultivos, en el caso de temperatura si el cultivo está en un invernadero debe permitirse abrir o cerrar de forma automática las ventanas cuando se detecte un cambio de temperatura.

# BIBLIOGRAFÍA

- [1] L. Castro, J. Yépez y C. Wittmer, «Impacto del cambio climático en la agricultura de subsistencia en el Ecuador,» *Fundación Carolina* , p. 95, 2012.
- [2] A. R. V. E. Michael Morris, «Panoramas alimentarios futuros Reimaginando la agricultura en América Latina y el Caribe,» *Grupo Banca Mundial*, p. 246, 2020.
- [3] W. R. G. A. J. D. Roger Freire, «Estudio del uso de apps en las actividades agrícolas de las pequeñas unidades productivas en el sector de Milagro - Ecuador,» *risti*, p. 16, 2020.
- [4] M. d. A. y. Ganadería, «Agricultura, la base de la economía y la alimentación,» 2019. [En línea]. Available: <https://www.agricultura.gob.ec/agricultura-la-base-de-la-economia-y-la-alimentacion/>.
- [5] E. M. T. V. F. M. I. C. F. Econ. MBA. Ana María Sánchez, «Diagnóstico sector agrícola Ecuador,» *Universidad Técnica de Ambato*, 2020.
- [6] J. P. d. M.-A. Richard L Snyder, «El daño producido por las heladas,» *Protección contra las heladas*, 2010.
- [7] Intagri, «El Estrés Vegetal Parte I: Estrés por Altas Temperaturas,» 2015. [En línea]. Available: <https://www.intagri.com/articulos/nutricion-vegetal/estres-vegetal-parte-1-estres-por-altas-temperaturas>.
- [8] Wikipedia, «Base de datos,» 2016. [En línea]. Available: [https://es.wikipedia.org/wiki/Base\\_de\\_datos](https://es.wikipedia.org/wiki/Base_de_datos).
- [9] Keepcoding, «¿Qué son los Datasets y cómo encontrarlos?,» 2020. [En línea]. Available: <https://keepcoding.io/blog/que-son-datasets/>.

- [10] Gohawaii, «Maunaloa,» 2021. [En línea]. Available: <https://www.gohawaii.com/es/node/35851>.
- [11] Github, «MaunaLoaDailyTemps,» 2019. [En línea]. Available: <https://github.com/gupta24789/Machine-Learning-Datasets/blob/master/MaunaLoaDailyTemps.csv>.
- [12] Dartmouth-Hitchcock, «Mt. Washington,» 2021. [En línea]. Available: <https://dhproviders.org/portfolio/nh-outdoors/>.
- [13] Github, «MtWashingtonDailyTemps,» 2019. [En línea]. Available: <https://github.com/gupta24789/Machine-Learning-Datasets/blob/master/MtWashingtonDailyTemps.csv>.
- [14] Tagsa, «GUAYAQUIL, PERLA DEL PACÍFICO,» 2017. [En línea]. Available: <http://www.tagsa.aero/guayaquil.html>.
- [15] INAMHI, «Temperatura Aire M1271 Guayaquil (Facultad CCNN),» 2021. [En línea]. Available: <http://sedc.fonag.org.ec/reportes/inamhi>.
- [16] P. Rodo, «Economipedia,» 8 Junio 2019. [En línea]. Available: <https://economipedia.com/definiciones/modelo-ar1.html>.
- [17] A. Hayes, «Inestopedia,» 12 Octubre 2021. [En línea]. Available: <https://www.inestopedia.com/terms/a/autoregressive-integrated-moving-average-arima.asp>.
- [18] R. H. A. S. H.S. Hota, «Time Series Data Prediction Using Sliding Window Based RBF Neural Network,» *International Journal of Computational Intelligence Research*, p. 12, 2017.
- [19] M. Sotaquirá, «codificandobits,» 8 06 2019. [En línea]. Available: <https://www.codificandobits.com/blog/introduccion-redes-neuronales-recurrentes/>.
- [20] J. Torres, «torres,» 22 Septiembre 2019. [En línea]. Available: <https://torres.ai/redes-neuronales-recurrentes/>.

- [21] M. Sotaquirá, «codificandobits,» 20 Junio 2019. [En línea]. Available: <https://www.codificandobits.com/blog/redes-lstm/>.
- [22] A. Casallas, «researchgate,» Agosto 2020. [En línea]. Available: [https://www.researchgate.net/figure/Celda-de-una-red-neuronal-recurrente-LSTM-Donde-los-valores-de-C-son-los\\_fig3\\_343450066](https://www.researchgate.net/figure/Celda-de-una-red-neuronal-recurrente-LSTM-Donde-los-valores-de-C-son-los_fig3_343450066).
- [23] S. Saxena, «Analytics Vidhya,» 17 Marzo 2021. [En línea]. Available: <https://www.analyticsvidhya.com/blog/2021/03/introduction-to-gated-recurrent-unit-gru/>.
- [24] M. Phi, «towardsdatascience,» 24 Septiembre 2018. [En línea]. Available: <https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21>.
- [25] G. Lawton, «TechTarget,» 5 Abril 2021. [En línea]. Available: <https://searchenterpriseai.techtarget.com/feature/Transformer-neural-networks-are-shaking-up-AI>.
- [26] Hubspot, «Qué es la interfaz de usuario y qué tipos existen,» 2021. [En línea]. Available: <https://blog.hubspot.es/marketing/interfaz-usuario>.
- [27] S. Inc., «¿Qué es una Aplicación Web?,» 2019. [En línea]. Available: <https://www.strappinc.com/blog/strapp-datos/que-es-una-aplicacion-web>.
- [28] Glajumedia, «TOP FRAMEWORKS DE DESARROLLO WEB EN EL 2020,» 2020. [En línea]. Available: <https://www.glajumedia.com/top-frameworks-de-desarrollo-web-en-el-2020/>.
- [29] Platzi, «Qué es Frontend y Backend: diferencias y características,» 2017. [En línea]. Available: <https://platzi.com/blog/que-es-frontend-y-backend/>.
- [30] Skedler, «Everything You Need to Know about Grafana,» 2021. [En línea]. Available: <https://www.skedler.com/blog/everything-you-need-to-know-about-grafana/>.

- [31] C. Collins, «Monitor your home's temperature and humidity with Raspberry Pis and Prometheus,» 2021. [En línea]. Available: <https://opensource.com/article/21/7/home-temperature-raspberry-pi-prometheus>.
- [32] Prometheus, «Github,» 2021. [En línea]. Available: <https://github.com/prometheus/prometheus>.
- [33] Medium, «InfluxDB,» 2019. [En línea]. Available: <https://medium.com/@malkisandamini95/influxdb-e283d95bf3b8>.
- [34] QuestDB, «Github,» 2021. [En línea]. Available: <https://github.com/questdb/questdb>.
- [35] H. Á. T. G. Gloria Amaris, «Aplicación de modelo ARIMA para el análisis de series de volúmenes anuales en el río Magdalena,» *Scielo*, p. 5, 2017.
- [36] M.-A. R.-R. J. P.-C. J.-L. B.-M. Luis-Roberto Jácome-Galarza, «Time Series in Sensor Data Using State-of-the-Art Deep Learning Approaches: A Systematic Literature Review,» *Smart Innovation, Systems and Technologies*, vol. 252, 2021.
- [37] Caner, «Medium,» 25 Marzo 2020. [En línea]. Available: <https://medium.com/@canerkilinc/selecting-lstm-hyperparameter-timesteps-edf27a243a9>. [Último acceso: 23 Noviembre 2021].
- [38] Caner, «Medium,» 25 Marzo 2020. [En línea]. Available: <https://medium.com/@canerkilinc/selecting-lstm-hyperparameter-timesteps-edf27a243a9>. [Último acceso: 23 Noviembre 2021].
- [39] Natasha, «Medium Klingenbrunn,» 19 Febrero 2021. [En línea]. Available: <https://medium.com/mlearning-ai/transformer-implementation-for-time-series-forecasting-a9db2db5c820>. [Último acceso: 23 Noviembre 2021].
- [40] «Socialnautas,» 3 Julio 2018. [En línea]. Available: <https://www.socialnautas.es/frente-a-excel/>. [Último acceso: 23 Noviembre 2021].

[41] Glassdoor, «<https://www.glassdoor.com.mx>,» 27 01 2022. [En línea]. Available: [https://www.glassdoor.com.mx/Sueldos/quito-software-developer-sueldo-SRCH\\_IL.0,5\\_IM1362\\_KO6,24.htm](https://www.glassdoor.com.mx/Sueldos/quito-software-developer-sueldo-SRCH_IL.0,5_IM1362_KO6,24.htm).

[42] P. P. Torralba, «iebs,» 21 Mayo 2021. [En línea]. Available: <https://www.iebschool.com/blog/redes-neuronales-convolucionales/>.

# ANEXOS

## ANEXO A

### Formulario Administrador

**\*Obligatorio**

¿Qué tan difícil fue realizar la tarea 1? \*

Muy fácil

Fácil

Neutral

Difícil

Muy difícil

No pudo realizar la tarea

¿Qué tan difícil fue realizar la tarea 2? \*

Muy fácil

Fácil

Neutral

Difícil

Muy difícil

No pudo realizar la tarea

Figura 4.1 Formulario Administrador 1

¿Qué tan difícil fue realizar la tarea 3? \*

- Muy fácil
- Fácil
- Neutral
- Difícil
- Muy difícil
- No pudo realizar la tarea

¿Qué tan difícil fue realizar la tarea 4? \*

- Muy fácil
- Fácil
- Neutral
- Difícil
- Muy difícil
- No pudo realizar la tarea

**Figura 4.2 Formulario Administrador 2**

¿Qué tan difícil fue realizar la tarea 5? \*

- Muy fácil
- Fácil
- Neutral
- Difícil
- Muy difícil
- No pudo realizar la tarea

¿Qué tan difícil fue realizar la tarea 6? \*

- Muy fácil
- Fácil
- Neutral
- Difícil
- Muy difícil
- No pudo realizar la tarea

**Figura 4.3 Formulario Administrador 3**

¿Qué tan difícil fue realizar la tarea 7? \*

- Muy fácil
- Fácil
- Neutral
- Difícil
- Muy difícil
- No pudo realizar la tarea

¿Qué tan fácil ha sido usar la aplicación web? \*

- Muy fácil
- Fácil
- Neutral
- Difícil
- Muy difícil

Figura 4.4 Formulario Administrador 4

¿Qué tan satisfecho se siente con el diseño de la aplicación web? \*

- Muy satisfecho
- Satisfecho
- Moderadamente satisfecho
- Poco satisfecho
- No satisfecho

¿Qué tan fácil fue entender las gráficas visualizadas? \*

- Muy fácil
- Fácil
- Neutral
- Difícil
- Muy difícil

**Figura 4.5 Formulario Administrador 5**

# Formulario Agricultor

\*Obligatorio

¿Qué tan difícil fue realizar la tarea 1? \*

- Muy fácil
- Fácil
- Neutral
- Difícil
- Muy difícil
- No pudo realizar la tarea

¿Qué tan difícil fue realizar la tarea 2? \*

- Muy fácil
- Fácil
- Neutral
- Difícil
- Muy difícil
- No pudo realizar la tarea

Figura 4.6 Formulario Agricultor 1

¿Qué tan difícil fue realizar la tarea 3? \*

- Muy fácil
- Fácil
- Neutral
- Difícil
- Muy difícil
- No pudo realizar la tarea

¿Qué tan fácil ha sido usar la aplicación web? \*

- Muy fácil
- Fácil
- Neutral
- Difícil
- Muy difícil

**Figura 4.7 Formulario Agricultor 2**

¿Qué tan satisfecho se siente con el diseño de la aplicación web? \*

- Muy satisfecho
- Satisfecho
- Moderadamente satisfecho
- Poco satisfecho
- No satisfecho

¿Qué tan fácil fue entender las gráficas visualizadas? \*

- Muy fácil
- Fácil
- Neutral
- Difícil
- Muy difícil

**Figura 4.8 Formulario Agricultor 3**