



Disertación

Maestría en Ingeniería Informática - Computación Móvil

***Sistema de monitorización y predicción de datos
en un invernadero de rosas basado en
Wireless Sensor Networks***

SCHUBERT DAVID RODRIGUEZ QUEZADA

Leiria, *septiembre de 2017*



Disertación

Maestría en Ingeniería Informática - Computación Móvil

***Sistema de monitorización y predicción de datos
en un invernadero de rosas basado en
Wireless Sensor Networks***

SCHUBERT DAVID RODRIGUEZ QUEZADA

Disertación de Maestría realizada bajo la orientación del Doctor Carlos Fernando de Almeida Grilo, Profesor de la Escuela Superior de Tecnología y Gestión del Instituto Politécnico de Leiria y codirección de la Doctora Tatiana Marisol Gualotuña Álvarez, Profesora del Departamento de Ciencias de la Computación de la Universidad de las Fuerzas Armadas - ESPE.

Leiria, *septiembre de 2017*

Esta página fue intencionalmente dejada en blanco

Dedicatoria

Dedicado para la memoria de mi padre Schubert Fernando Rodríguez Montiel.

Esta página fue intencionalmente dejada en blanco

Agradecimientos

Primero agradecer a mi increíble madre Miryan, quién con su apoyo incondicional durante cada aspecto de mi vida me ha fortalecido e inspirado para ser cada vez una persona mejor. Para mi hermano quien durante muchos años se ha sentado junto a mí por el simple hecho de hacer compañía y alivianar el estrés; ahora conversamos de muchos temas. Muchas gracias mí querido Alexis.

Para mis queridos profesores quienes me mostraron una espectacular devoción para la enseñanza y genuino interés en que aprendiera cada lección dentro y fuera de las aulas de clase.

Mi querida profesora Tatiana Gualotuña me ayudado durante muchas etapas de mi vida académica, profesional y personal, muchas gracias por estar presente una vez más y seguir apoyando mi crecimiento profesional.

Un agradecimiento muy especial para usted profesor Carlos Grilo, quien más que un profesor se convirtió en una guía durante la consecución de este trabajo y más importante se convirtió en un incondicional amigo, de verdad muchas gracias por todo su apoyo desde el principio.

Esta página fue intencionalmente dejada en blanco

Resumen

Para muchos países el sector florícola constituye una buena parte del Producto Interno Bruto (PIB) y se convierte en una de las principales actividades económicas para su desarrollo. Con la finalidad de proveer las mejores condiciones para el tratamiento de un cultivo de rosas en invernaderos se diseñó una *Wireless Sensor Network* de bajo costo (*low cost*), que permita la recolección a través de sensores y visualizar el monitoreo de datos de condiciones ambientales que incurren en el crecimiento del cultivo, entre las de mayor relevancia: Temperatura, Luminosidad y Humedad de ambiente y suelo.

Se desarrolló una aplicación móvil y web que pueda ser accedida por agricultores con la finalidad de conocer el estado de su invernadero en todo momento y en todo lugar que cuente con acceso a Internet. Además, la aplicación provee la capacidad de efectuar predicciones de las condiciones ambientales en un intervalo de tiempo futuro, basado en los datos recolectados y técnicas de minería de datos (Data Mining), y permita que el agricultor pueda tomar decisiones sobre mitigación a riesgos que puedan comprometer la integridad de su cultivo.

Los resultados del modelo predictivo, aunque puedan ser objeto de mejora, fueron aceptables en valores absolutos con pequeñas diferencias para las cuatro variables ambientales de este estudio cuando son comparados con los valores reales. Estos resultados indican que con la efectiva mitigación de problemas en términos de pérdida de información pueden brindar valiosa información a los interesados.

La operacionalización de este trabajo toma como caso de estudio un invernadero de rosas disponible en el Instituto Agropecuario Superior Andino (IASA) perteneciente a la Universidad de las Fuerzas Armadas – ESPE en Ecuador.

Palabras-clave: Wireless Sensor Networks, Minería de Datos, Internet de las cosas, Agricultura de Precisión, Invernaderos Pasivos, Desarrollo de aplicaciones móviles

Esta página fue intencionalmente dejada en blanco

Abstract

For many countries, the flower sector constitutes a large part of the Gross Domestic Product (GDP) and becomes one of the main economic activities for its development. In order to provide the best conditions for the treatment of a rose crop in greenhouses, a low-cost Wireless Sensor Network was designed, which allows the collection through sensors and visualization of the data monitoring of environmental conditions that incur in crop growth, among the most relevant: Temperature, Luminosity and Humidity of environment and soil.

A mobile and web application was developed that might be accessed by farmers in order to know the state of their greenhouse at all time and in all places with Internet access. The application provides the ability to make predictions of weather conditions in a future time period, based on data collected and data mining techniques, and allows farmers to make decisions on risks mitigations that may compromise the integrity of their crop.

The results of the predictive model, although they can be subject of improvements, were acceptable in absolute values with small differences in error absolute values for the four environmental conditions for this study when they are compared with real values. These results indicate that effective mitigation of problems in terms of information lost could provide valuable information to stakeholders.

The operationalization of this work takes as a case study a rose greenhouse available at the Andean Higher Agricultural Institute (IASA) belonging to the University of the Armed Forces - ESPE in Ecuador.

Keywords: Wireless Sensor Networks, Data Mining, Internet of Things, Precision Agriculture, Greenhouses, Mobile Applications Development

Esta página fue intencionalmente dejada en blanco

Lista de figuras

Figura 1: Herramientas y Frameworks relacionados a iOS	13
Figura 2: Herramientas y Frameworks relacionados a Android	14
Figura 3: Herramientas y Frameworks relacionados a Crossover	15
Figura 4: Proceso de SEMMA de Data Mining.....	16
Figura 5: Ciclo del proceso de Data Mining.....	18
Figura 6: Representación del funcionamiento de un Red Neuronal Artificial.....	22
Figura 7: Tipos de funciones de activación de una neurona	23
Figura 8: Arquitectura General del Proyecto	26
Figura 9: Componente Xbee S2.....	34
Figura 10: Sensor de temperatura y humedad de ambiente DHT11	34
Figura 11: Sensor de humedad de suelo FC-28	35
Figura 12: Sensor de luminosidad BH1750.....	35
Figura 13: Implementación de WSN en Invernadero de IASA	37
Figura 14: Implementación de Nodo Recolector	38
Figura 15: Modelo de casos de Uso.....	45
Figura 16: Modelo conceptual de dominio	45
Figura 17: Modelo Entidad-Relación No-SQL.....	47
Figura 18: Modelo Navegacional.....	47
Figura 19: Pantalla de bienvenida e inicio de sesión	49
Figura 20: Menú principal	49
Figura 21: Pantalla de monitoreo de condiciones ambientales.....	49
Figura 22: Pantalla de mensaje sobre el monitoreo de condiciones ambientales	50
Figura 23: Predicción de condiciones ambientales.....	50
Figura 24: Arquitectura de aplicación móvil y web.....	51
Figura 25: Implementación de la pantalla de bienvenida	52

Figura 26: Implementación de la pantalla de inicio de sesión	53
Figura 27: Implementación del menú principal	53
Figura 28: Implementación de administración de nodos	54
Figura 29: Implementación de monitoreo de los nodos.....	54
Figura 30: Implementación de monitoreo gráfico	55
Figura 31: Implementación de pantalla de predicción.....	55
Figura 32: Implementación de pantalla de parametrización de umbrales.....	56
Figura 33: Análisis de código de aplicación y API.....	56
Figura 34: Muestra de datos crudos leída por los sensores.....	59
Figura 35: Diagrama de flujo del proceso de preparación de datos (Script Python)	60
Figura 36: Muestra de datos organizados por $m=30$	61
Figura 37: Modelo con algoritmo de regresión lineal para temperatura, $m=30$ y $h=3$	65
Figura 38: Modelo con algoritmo de regresión lineal para temperatura, $m = 30$, $h = 5$ y $C = 23$	66
Figura 39: Modelo con algoritmo de redes neuronales para humedad de ambiente, $m = 30$, $h = 6$ y $HL = 10$	66
Figura 40: Datos de las últimas 5h para el servicio web.....	71
Figura 41: Consumo de servicio web expuesto por el API.....	71

Esta página fue intencionalmente dejada en blanco

Lista de tablas

Tabla 1: Rango de valores de los sensores.....	36
Tabla 2: Tabla de correlación entre variables ambientales.....	61
Tabla 3: Factores de combinación de escenarios.....	62
Tabla 4: Preparación datos para la variable temperatura $m=30$, $h=3$, misma variable.....	63
Tabla 5: Preparación datos para la variable temperatura $m=30$, $h=3$, todas las variables ...	63
Tabla 6: Preparación datos para la variable luminosidad $m=15$, $h=2$, misma variable	64
Tabla 7: Preparación datos para la variable temperatura $m=30$, $h=2$, misma variable.....	64
Tabla 8: Preparación de los datos en el script.....	64
Tabla 9: Resultados de Regresión lineal para 2h a 6h con error de desviación estándar	67
Tabla 10: Resultados de Regresión lineal para 2h a 6h sin error de desviación estándar....	67
Tabla 11: Resultados de Redes Neuronales para 2h a 6h sin error de desviación estándar.	68
Tabla 12: Resultados de Support Vector Machines para 2h a 6h sin error de desviación estándar	69
Tabla 13: Resultados de Support Vector Machines para 5h luminosidad	70
Tabla 14: Resultados de Support Vector Machines para 5h humedad de ambiente	70
Tabla 15: Resultados de Support Vector Machines para 5h humedad de suelo	70

Esta página fue intencionalmente dejada en blanco

Lista de siglas

API	Application Programing Interface
BDD	Behavior Driven Development
CO ₂	Carbon Dioxide
CRISP	Cross-Industry Process for Data Mining
CSS	Cascading Style Sheets
CSV	Comma Separated Values File
DDP	Datagram Delivery Protocol
ESPE	Escuela Politécnica del Ejército
GNU	GNU's Not Unix
GPA	Greenhouse Prediction API
GUI	Graphical User Interface
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
IASA	Instituto Agropecuario Superior Andino
IDE	Integrated Development Environment
IP	Internet Protocol
IPL	Instituto Politécnico de Leiria
ISO	International Organization for Standardization
JS	JavaScript
KDD	Knowledge discovery in databases
KDLC	Knowledge Discovery Life Cycle
MVC	Model-View-Controller
NDK	Native Development Kit
PH	Potential of hydrogen
PIB	Producto Interno Bruto
PPM	Parts Per Million
REST	Representational State Transfer
RNA	Redes Neuronales Artificiales
SDK	Software Development Kit
SEMMA	Selecting-Exploring-Modifying-Modeling-Assessment
SQL	Structured Query Language
SVM	Support Vector Machines

TDD	Test Driven Development
TIC	Tecnología de la Información y Comunicación
UI	User Interface
UML	Unified Modeling Language
URL	Uniform Resource Locator
UWE	UML – Based Web Engineering
WSN	Wireless Sensor Network
XML	Extensible Markup Language
XLSX	Microsoft Excel Open

Esta página fue intencionalmente dejada en blanco

Índice

Dedicatoria (<i>facultativo</i>).....	iii
Agradecimientos (<i>facultativo</i>)	v
Resumen	vii
Abstract.....	ix
Lista de figuras	xi
Lista de tablas.....	xiv
Lista de siglas	xvi
Índice	xix
1. Introducción	2
1.1. Motivación	3
1.2. Objetivos y Contribuciones	4
1.3. Estructura del documento.....	4
2. Materia relacionada	6
2.1. Proyectos afines	6
2.2. Producción de rosas.....	8
2.2.1. Temperatura	9
2.2.2. Humedad ambiente	9
2.2.3. Humedad de suelo.....	10
2.2.4. Luminosidad	11
2.2.5. Enriquecimiento del CO2	11
2.3. Desarrollo de aplicaciones móviles	11
2.3.1. Desarrollo nativo iOS	13
2.3.2. Desarrollo nativo Android.....	14
2.3.3. Desarrollo Crossover Framework	15
2.4. Data Mining.....	16
2.4.1. Fases del Data Mining	18
2.5. Modelos de aprendizaje.....	20
2.5.1. Regresión Lineal.....	20
2.5.2. Redes Neuronales.....	21

2.5.3.	<i>Support Vector Machines</i>	23
2.6.	Resumen del capítulo	24
3.	Arquitectura y tecnologías utilizadas	26
3.1.	Arquitectura general	26
3.2.	Herramientas y frameworks	28
3.2.1.	IntelliJ IDEA 2016.2.3	28
3.2.2.	WebStorm 2016.2.3.....	28
3.2.3.	PyCharm 2016.2.3.....	29
3.2.4.	Framework Meteor 1.4.....	30
3.2.5.	Weka.....	31
3.2.6.	Pandas	32
3.3.	Resumen del capítulo	32
4.	Implementación de la Wireless Sensor Network	33
4.1.	Materiales	33
4.2.	Diseño de los Nodos	36
4.3.	Experiencias	38
4.3.1.	Conectividad	38
4.3.2.	Autosuficiencia	39
4.3.3.	Cobertura.....	39
4.3.4.	Componentes innecesarios	40
4.3.5.	Mal Funcionamiento.....	40
4.3.6.	Factor humano	40
4.3.7.	Resultados	41
4.4.	Resumen del capítulo	41
5.	Aplicación móvil	43
5.1.	Metodología de desarrollo	43
5.1.1.	Captura, análisis y especificación de requerimientos.....	43
5.1.2.	Diseño del sistema	44
5.2.	Modelos y diseño	44
5.2.1.	Modelo de casos de Uso	44
5.2.2.	Modelo conceptual.....	45
5.2.3.	Modelo Entidad - Relación.....	46
5.2.4.	Modelo Navegacional	47

5.2.5.	Modelo Interfaz Abstracta (UI)	48
5.3.	Desarrollo de la aplicación	50
5.3.1.	Arquitectura de la aplicación	51
5.3.2.	Funcionalidad de la aplicación	52
5.4.	Pruebas y Experiencias.....	56
5.5.	Resumen del capítulo	57
6.	Modelo predictivo	58
6.1.	Definición del problema.....	58
6.2.	Exploración y Preparación de los datos	58
6.2.1.	Datos crudos.....	59
6.2.2.	Limpieza de los datos.....	59
6.2.3.	Exploración de datos	61
6.2.4.	Preparación de datos.....	62
6.3.	Modelamiento	65
6.4.	Evaluación	66
6.4.1.	Regresión Lineal.....	67
6.4.2.	Redes Neuronales.....	68
6.4.3.	Support Vector Machines	69
6.4.4.	Otras variables ambientales	69
6.5.	Despliegue	71
6.6.	Resumen del capítulo	72
7.	Conclusiones y trabajo futuro	73
8.	Bibliografía	75

Esta página fue intencionalmente dejada en blanco

CAPÍTULO I

1. Introducción

Para el Ecuador, uno de los principales productos de exportación son las rosas. El Ecuador es un país que principalmente radica sus ingresos anuales en actividades primarias. Gran parte de su evolución es buscar el crecimiento económico en áreas que no impliquen la explotación de recursos naturales como el petróleo. En esta última década el país experimentó notorios cambios en cuestiones económicas. En el año 2007 a 2009 se registró un crecimiento promedio entre 1.9% llegando a ser 4.0% de crecimiento promedio. No obstante, en el segundo semestre tuvo su primera caída de consideración al caer el precio del petróleo un 48% de su cotización en el año anterior dejando en evidencia la dependencia de la explotación del crudo como principal fuente de ingreso, aunque este impacto duró apenas unos cuantos meses (El Telégrafo, 2016).

El Ecuador tiene buscado diferentes formas de crecimiento económico y, después de la construcción, con un PIB de 0.87%, uno de los pilares es la agricultura, con un PIB de 0.45%, que más contribuye al desarrollo del país. La producción de rosas en el Ecuador es un pilar económico, situándolos entre los principales productores a nivel mundial debido a su geografía que hace un lugar idóneo durante todo el año. Esto es debido a que el día y la noche duran exactamente lo mismo y sus estaciones no son mayormente diferenciadas en función de variables ambientales y pueden ser controladas.

Según datos de (Expoflores, 2016), los meses de febrero y mayo son los de mayor exportación, correspondiendo, respectivamente, a 114,2 millones y 71.2 millones de usd en el año 2016, por las festividades de aquellos meses dando un total de 364.3 millones en el periodo comprendido entre enero a mayo, manteniendo una constante relación entre los años 2014 y 2015.

El cultivo apropiado de este producto, radica en ciertas variables ambientales tales como: temperatura ambiente, humedad relativa, humedad del suelo e intensidad luminosa. El sector agrícola en el país se encuentra principalmente en puntos geográficos remotos y con dificultades en acceso a tecnología. Sin embargo, los agricultores que se dedican al área florícola requieren de construir un invernadero para proteger el cultivo de condiciones climáticas desfavorables. Los

invernaderos suelen ser pasivos y son estructuras rudimentarias y no están provistas de ventiladores, sistemas de calefacción ni sistemas de riego.

Existen soluciones tecnológicas que permiten el monitoreo de áreas de cultivo, sin embargo, su implementación representa un alto costo; herramientas como un fitómetro que permite monitorear el ambiente a tiempo real puede superar los 300 mil dólares. Por esta razón, el sector florícola, especialmente el de rosas, puede ser beneficiado con el uso de tecnología, especialmente con soluciones de bajo costo que puedan ser adquiridas e implementadas.

1.1. Motivación

El sector agrícola a nivel mundial se ha visto beneficiado con el avance vertiginoso de las tecnologías de la información y la comunicación, a través de la aplicación de soluciones tecnológicas orientadas especialmente a mejorar la producción y productividad de los cultivos (REDCEDIA, 2016).

En países como Ecuador, el acceso a recursos tecnológicos aún es una limitante, debido principalmente a los altos costos y al desconocimiento de las soluciones existentes en el mercado (REDCEDIA, 2016). Por lo tanto, teniendo en cuenta las limitaciones financieras, tiene sentido que el sector florícola apueste por implementar tecnologías de bajo costo y aprovechar las TIC para mejorar el proceso de cultivo.

Existen algunos escenarios en los que resulta beneficioso contar con implementaciones tecnológicas. Por ejemplo, durante el proceso de desarrollo de la flor, las condiciones climáticas pueden ser monitorizadas, permitiendo a los agricultores tomar evidencias del estado actual de su cultivo y alterar las condiciones de su invernadero de modo que no sobrepasen umbrales que puedan afectar la productividad. De la misma forma, los datos que fueron recolectados pueden ser usados para construir un algoritmo, con la finalidad de prever anticipadamente la ocurrencia de situaciones que puedan afectar al cultivo y tomar acciones correctivas al respecto.

Debido a la problemática de que estos cultivos son realizados manualmente, surge la necesidad de realizar estudios para optimizar el adecuado cultivo de este producto de exportación para el Ecuador.

1.2. Objetivos y Contribuciones

El objetivo general de este trabajo consiste en desarrollar un sistema de monitoreo y predicción en el cultivo de rosas en ambientes controlados. Se pretende implementar un algoritmo predictivo que permita prever los valores de diferentes variables ambientales con el menor porcentaje de error posible. Además, se pretende desarrollar una aplicación móvil que refleje los resultados obtenidos y permita el monitoreo en plataformas Web, Android y iOS, utilizando tecnologías y *frameworks* de desarrollo ágil.

El proyecto de investigación se define como prototipal, por lo que un primer objetivo es la de la implementación de una Red de Sensores Inalámbricos (WSN, del inglés Wireless Sensor Networks), en la que se ensamblan dos nodos de datos y un nodo coordinador para obtener las lecturas de las condiciones climáticas con la capacidad de comunicarse por red (hito para el cual se trabaja conjuntamente con el equipo del laboratorio de aplicaciones móviles de la Universidad de las Fuerzas Armadas, Ecuador), así como también, se utiliza una sección del espacio físico de un invernadero de rosas en el Instituto Agropecuario Superior Andino (IASA). Como segundo objetivo se definió el desarrollo de la aplicación web y móvil, al que se denomina GPA (*Greenhouse Prediction Application*), misma que se encarga de mostrar información sobre los datos obtenidos de los sensores. El último objetivo es la fase final de este proyecto, basado en los datos obtenidos en fases previas, y consiste en el desarrollo de un algoritmo predictivo de las variables ambientales que permita al agricultor tomar medidas preventivas sobre la predicción capaz de prever la tendencia de las variables climáticas para que esos valores se mantengan dentro del intervalo de valores admisibles. Para la consecución del modelo predictivo se usa algoritmos de *Machine Learning*.

1.3. Estructura del documento

Se termina esta introducción con una breve descripción de los restantes capítulos de la disertación. El segundo capítulo presenta la materia relacionada, trabajos afines que se han realizado y el marco referencial sobre el que el proyecto es sustentado. El tercer capítulo muestra la arquitectura general considerada para este proyecto y las herramientas consideradas para el desarrollo del sistema. El Capítulo 4 muestra las experiencias y el proceso de implementación de

la WSN (herramientas, arquitectura y el diseño de los nodos). El Capítulo 5, define la implementación de la aplicación móvil, la metodología, los modelos necesarios para diseñar la aplicación y las experiencias durante el desarrollo. El Capítulo 6, muestra el proceso de preparación de los datos y define la fase en la que se desarrollan pruebas del modelo de predicción, su implementación y posterior evaluación. Finalmente, el último capítulo está reservado para las conclusiones y para la descripción de algunas direcciones posibles para trabajo futuro.

CAPÍTULO II

2. Materia relacionada

El presente capítulo dispone la presentación de los trabajos que presentan relación con el enfoque de esta investigación; así como también dispone de los conceptos teóricos del estado del arte en proyectos para monitorización en invernaderos. Algunos de los parámetros de mayor incidencia para el cultivo y crecimiento de rosas también son expuestos, además de conceptos técnicos relacionados al desarrollo de aplicaciones móviles y otros que aportan el conocimiento acerca de diferentes técnicas que podrían ser implementadas para el tratamiento de los datos.

2.1. Proyectos afines

Actualmente existen algunos estudios que podrán dar contexto a la propuesta que se desea implementar respecto al diseño del algoritmo predictivo. En relación al uso de técnicas de predicción en el área agrícola se pueden tomar ciertos estudios con diferentes enfoques. Los estudios en esta rama son realizados para incrementar la producción con un óptimo uso de recursos sustentables y mantenerlos.

Según (Bhagawati, Sen, Shukla, & Bhagawati, 2016), en la década de los noventa no se dio mucha importancia a repositorios con datos de mediciones. Debido a la falta de técnicas y métodos eficientes para ser tratadas, estos datos eran archivados más no procesados.

En la industria de la agricultura las predicciones ambientales cumplen un rol importante, dado que una gran parte de las actividades que se realizan son fuertemente afectadas por las condiciones climáticas. Sin embargo, cada tipo de cultivo tiene un conjunto de condiciones mínimas, óptimas y máximas para desarrollar su crecimiento basado en parámetros como temperatura y precipitación.

Hace algunos años atrás, el desarrollo de modelos matemáticos capaces de correlacionar datos con un porcentaje de error aceptable era una idea improbable por causa de la poca o nula recolección de datos. Actualmente la evolución de diferentes algoritmos de predicción con

mayor aproximación a un resultado fiable permite procesar grandes volúmenes de datos y, a su vez, dar un mejor criterio a agricultores para mejorar la productividad de sus cultivos en función del crecimiento y costo manteniendo un ambiente supervisado y controlado.

Una investigación desarrollada por (Gao, Malik, Santoso, & Zhu, 2016), utiliza datos que han sido recogidos desde 1840 en Canadá con datos recolectados en seis estaciones, disponible en una base de datos en línea, con mediciones de la mínima, máxima y temperatura promedio; cantidad total de lluvia, nieve y precipitación, humedad relativa, dirección y velocidad del viento, punto de temperatura de rocío y presión atmosférica. Además, cuenta con la posición en función de latitud, longitud y elevación de la estación. Cuando existen fallas del equipamiento se completa los datos faltantes utilizando la interpolación polinomial de LaGrange. El *dataset* es basado en un modelo de serie temporal (*Time Series Modeling*). La Universidad de Victoria remarca la primera la dependencia del tiempo y las tendencias estacionarias para no definir la predicción como un proceso de regresión lineal. La solución que su estudio revela es el tratamiento de los datos como un método auto-regresivo.

El estudio realizado por (L. Zhao, 2013) propone un sistema inteligente de predicción para el área agrícola basado en sensores inalámbricos. El sistema debe proveer de información necesario para la toma oportuna de decisiones. El sistema alimenta una base de conocimiento y los relaciona con reglas de inferencia difusas. Sin embargo, el estudio no menciona ningún caso de éxito para esta propuesta.

Otro trabajo relacionado al estudio de ciertos comportamientos, aunque enfocado en la predicción de plagas y pestes, es el de (A. K. Tripathy, 2013). Éste estudio propone monitorear la relación entre plagas, ambiente y calidad del cultivo. Los datos obtenidos fueron procesados con técnicas de minería y de regresión multivariada para obtener parámetros útiles que den relevancia a la predicción generada.

El estudio de (J. Su, 2012) propone dos tipos de algoritmo, similar al estudio realizado por la Universidad de Victoria enfocado únicamente en parámetros de temperatura, el primero con lógica difusa adaptativa y el otro denominado de promedio (medida de media estadística); según

sus autores la ventaja que aportan este tipo de algoritmos es su simplicidad y precisión frente a otros algoritmos de lógica difusa.

Un enfoque dado por (Bhagawati, Sen, Shukla, & Bhagawati, 2016), utiliza el modelo *Knowledge Discovery Life Cycle* (KDLC), para enfrentar cultivos utilizando datos en sistemas de información geográfica. El estudio de KDCL (S. W. Lee, 1998) empíricamente demuestra la validez de las clasificaciones y reglas de aprendizaje, siendo un enfoque basado en descubrir conocimiento en multi-estrategia. El estudio de (R. Jain, 2009) muestra el potencial de tres máquinas de aprendizaje, encontrando que un árbol de decisión basado en C4.5 infiere mejores resultados que una tradicional regresión lineal, enfocado a la difusión de enfermedades.

Por otra parte, un estudio más general con diferentes puntos de predicción en agricultura fue dado por (G. E. Meyer, 2004), mostrando que, comparado a métodos tradicionales, las máquinas de aprendizaje basados en lógica difusa usando el algoritmo ZGK, brinda mejor precisión y pueden ser potencialmente útiles para sensores remotos, administración de cultivos y control de plagas en agricultura de precisión.

2.2. Producción de rosas

La taxonomía de la rosa, como flor ornamental de cultivo, perteneciente a la familia Rosaceae cuyo nombre científico es *Rosa spp*, es ampliamente valorada con una creciente demanda en el mercado desde la década de los noventa. Existen diferentes variedades de flores y que presentan diferentes características en su morfología. Las especies que son comúnmente comercializadas son resultados de híbridos de especies de flores ya desaparecidas. Así para la flor cortada se utilizan tipos de té híbrida y en menor medida floribunda, ésta especie permite que las flores en racimos se abran simultáneamente” (Info Agro, 2016) .

Económicamente sus principales mercados de consumo están en Europa con Alemania, Estados Unidos y Japón. Es un cultivo bastante especializado que demanda de un gran número de cuidados. Existen invernaderos en Italia, Holanda, Francia, España y Alemania, con presencia en otros países en menor medida. En América latina se ha acrecentado su cultivo principalmente en países como México, Colombia y Ecuador, siendo este último el objeto principal de estudio por su ubicación geográfica.

La flor como tal debe poseer cualidades deseadas para corte como un tallo largo y rígido que oscile entre los 50 a 70 cm, follaje brillante, aptitud para el cultivo sin suelo y ser de apertura lenta y una buena resistencia a enfermedades (Info Agro, 2016).

Los cultivares de las rosas se clasifican en rosas grandes y multiflores, las rosas rojas siendo las de mayor demanda en el mercado global, seguido de las de coloración rosa, las amarillas y naranjas estando en aumento, mientras que finalmente las blancas y bicolors (Aurelio Barrera, 2007).

Los parámetros climáticos para un correcto cultivo de rosas, varían acorde a su variedad más las cuantificaciones estándar de estas pueden ser dispuestas en ciertos rangos. Según (Aurelio Barrera, 2007), los requerimientos climáticos que son necesarios para la mayoría de los cultivares de rosas, son temperatura, iluminación y ventilación y bióxido de carbono en invernadero.

2.2.1. Temperatura

Para la mayoría de los cultivares, las temperaturas óptimas para el crecimiento se requieren valores que estén cercanos a los 17 °C durante la noche y a 28 °C durante el día. Cualquier medida de temperatura ligeramente superior o inferior, pero fuera de ese rango podría ralentizar el desarrollo de la flor produciendo efectos como flores con exagerado número de pétalos, usualmente más pequeñas de lo normal o colores pálidos. Otros autores como (Universidad Nacional Abierta y a Distancia, 2008), establecen que las medidas ideales para el crecimiento están entre 17 °C a 25 °C, con límites de 14 °C y 28 °C. La rosa al ser una planta termo-periódica requiere de variaciones de temperatura durante el día.

Para cultivos forzados en invernaderos se debería garantizar una temperatura bastante alta durante el periodo de formación y un poco elevada durante la época de poda (30 °C) y bajarla durante los días posteriores a 25 °C en el día y 12 °C en la noche.

2.2.2. Humedad ambiente

La humedad ambiente es un parámetro estrechamente ligado a la salud en el crecimiento de la planta. Los rosales necesitan una humedad ambiental relativamente alta, regulada mediante

procesos de ventilación y de nebulización o humedeciendo los pasillos en las horas más calurosas del día (Aurelio Barrera, 2007), en algunas ocasiones con el uso de ventiladores de flujo horizontal o extractores de presión.

Se debe mantener una relación constante entre la temperatura del aire y la higrometría (rama de la Física que estudia la producción de la humedad atmosférica y la medida de sus variaciones), especialmente en temporadas calurosas. Durante el tiempo de poda, la humedad debe estar entre 85 y 90% mismas que favorece el crecimiento de las yemas y el crecimiento de la flor.

Durante el tiempo de recolección (30 días antes del corte) o punto corte se debe mantener una humedad oscilante entre 70% a 75%. Finalmente, se reduce la humedad al 60% hasta que finalice el ciclo.

Si durante cualquier momento del ciclo la humedad descendiera a cualquier valor inferior al 60% se producirían daños por deshidratación, presencia de plagas como ácaros o de enfermedades como el mildew polvoso (Universidad Nacional Abierta y a Distancia, 2008).

2.2.3. Humedad de suelo

La infiltración de agua, permite tener una adecuada presencia de ésta en el suelo, debido a que la mayor parte de incidentes relacionados a la baja producción es por insuficiente humedad en el suelo, sea por resequedad o por evaporación. Muchas de las malas producciones están relacionadas específicamente a este fenómeno y no con la escasez de lluvias.

La rosa es una planta, con parámetros exigentes con respecto al suelo ya que éste debe ser profundo y ligeramente ácido con un PH entre 6 y 6,5 (Universidad Nacional Abierta y a Distancia, 2008); mismo que será despreciado en el alcance de este proyecto por no ser un parámetro que presente cambios radicales; mientras que las condiciones de humedad de suelo o del sustrato deben estar apropiadamente aireadas y drenadas para evitar encharcamientos. Una medida de humedad del suelo apropiado para el cultivo de rosas debe ser un porcentaje entre 55% a 65% de presencia de agua. El cálculo se determina obteniendo una relación entre el peso del agua evaporada durante el secado y de su volumen, una técnica de cálculo volumétrico. Otro método es el gravimétrico calculado por el peso del agua dentro de la matriz del suelo. (Universidad Nacional Abierta y a Distancia, 2008).

2.2.4. Luminosidad

La iluminación está estrechamente ligada con el crecimiento de la planta, siendo la temporada de verano mucho más óptima para la producción de las flores. En países europeos que geográficamente se encuentran alejados de la línea ecuatorial, se usan prácticas de iluminación con lámparas de vapor de sodio a un nivel de 3000 lux, mejorando la producción invernal en cantidad y en calidad, aunque el aspecto económico de esta práctica implica valores elevados. En contraste, en las temporadas de verano se emplean técnicas de oscurecimiento progresivamente para evitar deformidades en las flores (Info Agro, 2016). Zonas con baja iluminación producen colores menos brillantes en el tallo y en los pétalos e incrementa la proliferación de hongos.

La ventaja en las zonas ecuatoriales es que la intensidad lumínica no sufre de cambios abruptos ya que los días y las noches duran aproximadamente lo mismo durante todo el año, al ser constante la luminosidad el cultivo de las rosas no se ve afectado (Universidad Nacional Abierta y a Distancia, 2008).

2.2.5. Enriquecimiento del CO₂

El nivel de CO₂ es un parámetro que condiciona el crecimiento de la planta. En muchas zonas las primeras horas de la mañana son demasiado frías para ventilar y escasea la presencia del dióxido de carbono; es necesario que sea aportado artificialmente, elevándolo a 1000 ppm (Aurelio Barrera, 2007); mientras que en el atardecer si se realiza el cierre de la ventilación este parámetro tiende a descender por la actividad fotosintética (Info Agro, 2016).

2.3. Desarrollo de aplicaciones móviles

El desarrollo de aplicaciones móviles ha ido alcanzando su auge gracias a conceptos como el Internet de las cosas, mismo que provee de un enfoque de que cualquier sistema de información debe ser accesible a través de interfaces tangibles desde cualquier lugar, en cualquier momento; mientras tenga conexión a Internet, destacando el principio de ubicuidad.

Las interfaces con mayor auge y debido a su alta comercialización son los dispositivos móviles, siendo aquellos que basan su sistema operativo en iOS y Android los que gozan con mayor acogida en el mercado.

Según (Abrahamsson, 2010) el desarrollo de una aplicación nativa es una tarea que implica un gran desafío, debido a las exigentes demandas específicas y restricciones técnicas del ambiente móvil, tales como: capacidades limitadas y rápida evolución de los terminales de los dispositivos, varios estándares, protocolos y tecnologías de red, necesidad de operar en una variedad de diferentes plataformas, necesidades específicas de usuarios de terminales móviles y estrictos tiempos de requerimientos de mercado.

Las principales diferencias en los dispositivos móviles son concernientes a características físicas, como el tamaño de la pantalla, el peso, las dimensiones del dispositivo, sensores adicionales, mecanismos de entrada de datos y expansión. Otras características determinantes incluyen espacio de memoria, capacidad de la batería y el propio sistema operativo (Abrahamsson, 2010). El desarrollo sobre plataformas nativas permite explotar ciertas características propias de cada sistema operativo más reciente y en ciertos casos explotar características de hardware como en teléfonos con sensores de huellas digitales o con segundas pantallas, además de ciertas librerías propias que son publicadas a partir de ciertas versiones del sistema operativo impidiendo que la aplicación se ejecute en versiones previas a la especificación de las librerías recientemente añadidas.

Como solución a estos problemas que pueden ocasionarse, se han propuesto soluciones informáticas que permitan desarrollar exitosamente para las dos plataformas sin necesidad de cambio, por lo que se han desarrollado *crossover frameworks*, cuya finalidad es agilizar el desarrollo cuando el requerimiento es para ambas plataformas. Aunque este tipo de herramientas muestran ciertas desventajas a la hora de explotar todas las capacidades, son soluciones ampliamente acogidas por los desarrolladores actualmente.

Según (Andre Charland, 2011), el plan original para las aplicaciones de terceros (*third party applications*), era usar tecnologías orientadas a la web. Aunque agiliza el desarrollo en términos de performance, las aplicaciones basadas en web han sido desmerecidas en las comparaciones.

El costo de desarrollar una aplicación nativa para cada una de las plataformas es bastante elevado, según (Andre Charland, 2011). En un *startup* de un juego indie, podría únicamente ser costado el desarrollo de una plataforma presentando un impedimento para el departamento de

soporte de tecnologías de la información (TIC), siendo excluyente a usuarios que no cuenten con el dispositivo que cumplan con las características deseadas o el sistema operativo.

Otra consideración es en razón de rendimiento o performance, en la que el procesamiento de imágenes o de 3D puede ser más rápido en aplicaciones nativas, sin embargo, en aplicaciones construidas para fines empresariales el rendimiento es apenas considerable para ser una limitante.

2.3.1. Desarrollo nativo iOS

Según (Traeg, 2013), la mayoría de aplicaciones que se ejecutan en iOS, están escritas en el lenguaje Objective-C. El IDE oficial de Apple es XCode que actualmente se encuentra en la versión 8.0. El único sistema operativo soportado para esta herramienta son los Mac OS X, misma que esta provista del SDK de iOS y un simulador las cuales son de libre uso. Sin embargo, para publicar aplicaciones es necesario una licencia de desarrollador cuyo costo es de \$9900 (dólares americanos) por año. El otro lenguaje que ha ganado popularidad en estos últimos años es el lenguaje Swift, mismo que permite la incorporación de otro paradigma de programación funcional.

En la Figura 1 se muestran los logotipos del dispositivo iOS, el IDE y el lenguaje, que soportan al desarrollo nativo de iOS; XCode enfatiza en el modelo de diseño MVC, el mismo que se ha popularizando por ser el más simple de los patrones que es capaz de separar la lógica de la aplicación de la GUI. Principalmente, utiliza una herramienta embebida en el IDE para el diseño de interfaz para cada dispositivo en el que se ubican los diferentes elementos según restricciones en su posición y tamaño; para administrar estas GUI se ocupan clases Controladoras y de Navegación.



Figura 1: Herramientas y Frameworks relacionados a iOS

2.3.2. Desarrollo nativo Android

Según (Google, 2016), el conjunto de dispositivos que soportan Android presenta un amplio rango de dispositivos con diferentes características tanto en software como en hardware. El IDE oficial soportado por Google es Android Studio que actualmente se encuentra en la versión 3.0, basado en Java. Sin embargo, esta versión en particular permite un desarrollo en C/C++ utilizando NDK, como desarrollo nativo. El manejo de los recursos como GUI y dependencias para acceder a las diferentes características de los dispositivos está basado en XML con controladores en Java, soportando de múltiples lenguajes para su implementación. Google provee de una licencia de desarrollador para que las aplicaciones puedan ser publicadas en Google Play, tienda oficial para Android.

Los dispositivos con Android al no ser específicos de una compañía se consideran como un sistema operativo más abierto al desarrollo, por la que la mayoría de APIs son desarrollados por terceros lo que da un mayor soporte el momento de desarrollar una aplicación. El tipo de desarrollo que implica Android Studio también es enfocado implícitamente con MVC.

La Figura 2 presenta los logotipos del dispositivo *target*, la empresa y el lenguaje de desarrollo nativo para Android. En la última conferencia de Google del año 2017 (Google Inc., 2017), se presentó un nuevo lenguaje de desarrollo como oficial para el desarrollo nativo llamado Kotlin (Jetbrains, 2017). Este lenguaje promete ser el sucesor de Java como lenguaje oficial de Android y simplificar la formalidad del lenguaje que sugería la inherente necesidad de formalidad y por ende complejidad en secuencias sencillas en dispositivos móviles.



Figura 2: Herramientas y Frameworks relacionados a Android

2.3.3. Desarrollo Crossover Framework

Las aplicaciones de terceros que permiten ocupar las capacidades de desarrollo orientado a la Web, permiten aplicaciones que se ejecuten nativamente.

Herramientas con PhoneGap, permiten manipular las librerías Webkit de los dispositivos móviles y desplegar aplicaciones en los navegadores. Sin embargo, otro tipo de *frameworks* han llevado más allá este concepto permitiendo que las aplicaciones se ejecuten como si fueran nativas, percepción del usuario final, aprovechando las librerías de los navegadores para generar aplicaciones propias para cada sistema operativo.

Según (Abrahamsson, 2010), el problema es la deficiencia el momento de explotar la capacidad de hardware de un dispositivo o sensores propios de ese modelo podrían no funcionar o ser utilizados en este tipo de aplicaciones. El enfoque dependerá completamente del alcance de la aplicación y de ser necesario optar por un desarrollo multiplataforma.

El desarrollo de aplicaciones multiplataforma para dispositivos móviles se enfoca en el uso de lenguajes propiamente definidos para web como HTML, CSS y JavaScript, éste último siendo el principio de este tipo de aplicaciones, ya que se accede a la manipulación de este lenguaje sobre los otros que necesitan de compilación (Andre Charland, 2011).

En la Figura 3, se muestran las principales tecnologías como: Meteor, CordovaJS y JavaScript, que hacen posible este tipo de desarrollo de *crossover*. Debido a que, al ser implementados en lenguajes principalmente orientados para web, se necesitan de *plugins* o paquetes previamente instalados que permitan la trans-compilación del código a un lenguaje nativo referenciados en 2.3.1 y 2.3.2; más específicamente en el caso de iOS. Uno de los *plugins* más utilizados y con mayor soporte de la comunidad es el de CordovaJS (Apache, 2017).



Figura 3: Herramientas y Frameworks relacionados a Crossover

2.4. Data Mining

La minería de datos (del inglés, Data Mining) puede definirse como el proceso de descubrimiento de nuevas y significativas relaciones, patrones, perfiles, anomalías y tendencias el momento de examinar grandes volúmenes de datos. Esta área ha crecido mucho debido al bajo costo de almacenamiento, el crecimiento en la capacidad de recolectar datos y de generar información (Asencios, 2004).

Según (SAS Institute, 2017), la minería de datos se puede puntualizar con las siglas SEMMA, Seleccionar (*Selecting*), Explorar (*Exploring*), Modificar (*Modifying*), Modelar (*Modeling*) y Valorar (*Assessment*) grandes volúmenes de datos con la finalidad de buscar para descubrir patrones desconocidos para ser usados como ventaja competitiva en la industria. La Figura 4, extraída de (Pérez López, 2017), ilustra el proceso de SEMMA.

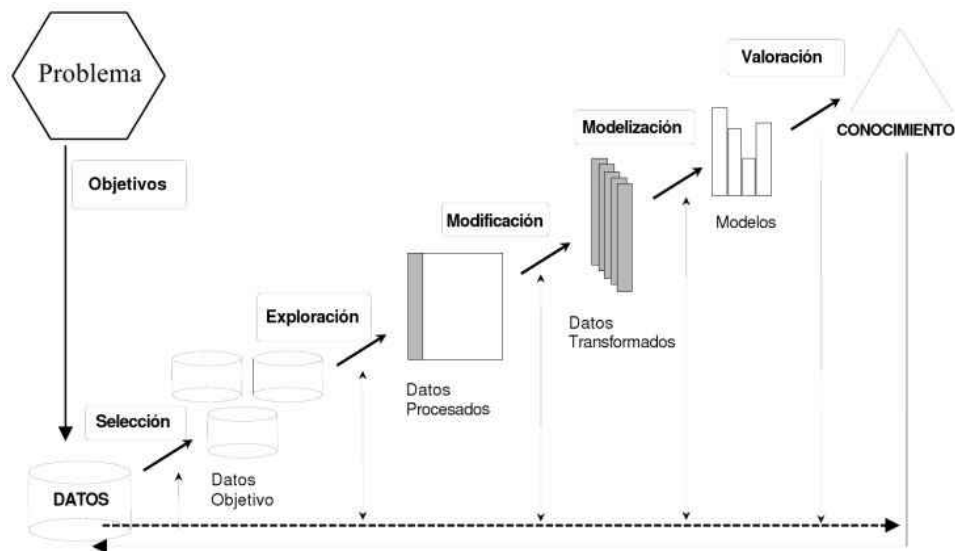


Figura 4: Proceso de SEMMA de Data Mining

El proceso de minería de datos busca como finalidad automatizar el descubrimiento del conocimiento que se encuentra contenido en los datos almacenados en grandes repositorios de datos. En base a técnicas que escarban información con la finalidad de encontrar tendencias, entre los principales tipos de técnicas se encuentran: redes neuronales, reconocimiento de patrones, regresiones lineales y no lineales, lógica difusa, algoritmos genéticos, etc.

El enfoque para este trabajo es el análisis predictivo, que se define como una rama del Data Mining, enfocada netamente al análisis de probabilidades y tendencias futuras. Basado en datos existentes de hechos pasados y actuales, se pretende encontrar patrones de comportamiento que permitan realizar predicciones sobre eventos a futuro. Los modelos que se desarrollen permiten encontrar potenciales riesgos u oportunidades. (Fundación Big Data, s.d.).

El elemento principal de los modelos de análisis de predicción es el *predictor*. Múltiples de ellos pueden combinarse para formar modelos predictivos más complejos. Primero, se prioriza la recolección de los datos y la formulación de un modelo de predicción (Evaluando Software, 2013). Los modelos pueden aplicarse a varias áreas de investigación, meteorología, seguridad, genética, economía, marketing, agricultura, tecnologías de la información, entre otras.

Los modelos predictivos son resultado de la aplicación de algoritmos sobre los datos. Según la distribución y complejidad de los mismos, da los criterios del tipo de algoritmo a ser seleccionado y las pruebas que han de ser realizadas sobre los datos del modelo.

Los datos de entrada son relevantes y valorados según la cantidad de información que se pueda obtener de ellos, para optimizar la toma de decisiones o mejorar la comprensión sobre fenómenos. (Asencios, 2004). Estas técnicas se enfocan en el descubrimiento del conocimiento (KDD, del inglés, *Knowledge Discovery in Databases*).

Según (Molina, 2000), KDD se puede definir como “La extracción no trivial de información potencialmente útil a partir de un gran volumen de datos, en el cual la información está implícita, donde se trata de interpretar grandes cantidades de datos y encontrar relaciones o patrones, para conseguirlo harán falta técnicas de aprendizaje, estadística y bases de datos”.

Las tareas principales de KDD serían la inducción de reglas, los problemas de clasificación y *clustering*, la detección de dependencias, el reconociendo de patrones, el modelado predictivo (Asencios, 2004), siendo estas las principales técnicas para solucionar el problema, que se desea afrontar en este estudio.

2.4.1. Fases del Data Mining

Las fases de un proceso de minería de datos son referenciadas en ciertas metodologías, una de las principales es CRISP (SV Europe, 2015), que es una metodología de minería de datos que fue habitual en organismos privados y públicos. Éste estándar se constituye como modelo y una guía en sus primeros años, se constituye en seis fases. Algunas de estas fases son bidireccionales lo que implica la revisión parcial o total de las fases anteriores. A continuación, se presenta cada una de las fases enfatizando sus características principales. Aunque actualmente la metodología no es la principal tendencia, es muy versátil para extraer ideas en cuanto a problemas similares.

La Figura 5, muestra el ciclo normal de las fases de minería de datos, correspondientes al diseño iterativo que el problema exige, ubicando como centro los datos, la definición del problema, exploración de los datos, preparación, modelamiento, evaluación y despliegue.

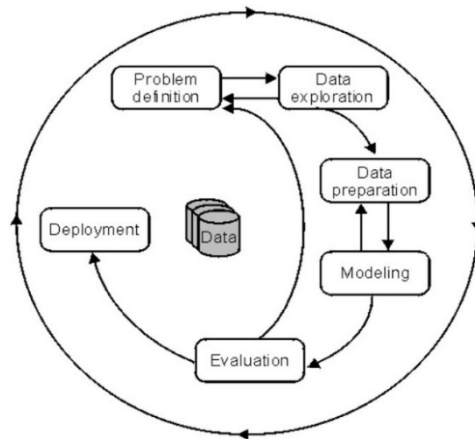


Figura 5: Ciclo del proceso de Data Mining

- **Comprensión del negocio**

Consiste en aprehender los objetivos y requerimientos del negocio, estableciendo cuáles de estos serán su contexto inicial y criterios de éxito; luego, se evalúa la situación inventariando los recursos, los supuestos y las terminologías propias del negocio. Posteriormente, se evalúan cuáles son los objetivos de la minería de datos, el objeto como tal del estudio, sus objetivos y los criterios de éxito. Finalmente, se especifica cuáles serán las herramientas, técnicas y la planificación a seguir.

- **Comprensión de los datos**

Esta fase consiste principalmente en familiarizarse con los datos en función de los objetivos del negocio y consiste en la recopilación inicial de datos, la descripción de las tipificaciones de los datos, la exploración de los datos y finalmente validación de la calidad de los mismos.

- **Preparación de los datos**

Consiste en obtener visiblemente el conjunto de datos a ser minado, también llamado *dataset*, la selección de las técnicas, la selección de los datos, la limpieza de los mismos, la construcción de los datos a manera de *dataset*, su integración y formateo.

- **Modelado**

Esta fase consiste en aplicar las técnicas de minería seleccionadas a los conjuntos de datos. Un breve resumen de la fase sería seleccionar la técnica de modelado, posteriormente diseñar la evaluación de verificación, construir el modelo y finalmente validarlo según sus principales indicadores.

- **Evaluación**

A partir de los modelos resultantes en la fase de modelado se determina y posteriormente verifica la significancia de los datos en el contexto del negocio. En esta fase la principal actividad es la evaluación del modelo, además de la revisión del proceso anterior con la finalidad de constatar la validez del modelo. Finalmente, se establece el conjunto de acciones a seguir, depurar el error del modelo puede ser una de ellas, procesar otro conjunto de datos u otras configuraciones en los algoritmos.

- **Despliegue**

La fase final de la metodología consiste en integrar el modelo con las tareas de toma de decisiones de la organización. Como pasos a considerar se tiene que realizar una planificación para el despliegue y planificar el monitoreo y mantenimiento del resultado del modelo.

2.5. Modelos de aprendizaje

El modelo de aprendizaje destaca a varios tipos de análisis predictivo, los cuales son el objeto de este estudio. Las siguientes secciones describen brevemente los modelos de predicción seleccionados para la implementación de este trabajo.

Entre los principales está el modelo predictivo, el cual se basa en analizar datos anteriores, y su finalidad, en nuestro caso es la de prever el valor de una variable momento futuro. Además, algunos modelos buscan patrones discriminatorios cuando la data es igual o similar; éste análisis usualmente es aplicado en la recolección de datos a tiempo real. (Fundación Big Data, s.d.).

2.5.1. Regresión Lineal

La regresión lineal es un modelo matemático usado en estadística que permite hallar el valor aproximado de una variable aleatoria o independiente y cuando una variable dependiente x toma un valor conocido. Este algoritmo implica un supuesto de que los valores presentan una tendencia continua y lineal, sea esta creciente o decreciente. Esto obliga que para que este método sea efectivo y como objetivo busca determinar el coeficiente de correlación entre las variables del modelo.

Usualmente los valores observados de la variable independiente muestran una diferencia de los que la función predice. Ese valor de descrito se le conoce como error de la función (ϵ), que es la diferencia entre los valores observados a los predichos. Esta diferencia se le conoce como variación residual. La estimación de los parámetros de la función se utiliza ajuste por mínimos cuadrados. Sin embargo, esta técnica supone que las variaciones se encuentren dentro de una distribución normal, manteniendo concordancia en el valor que difiere. Para verificar las suposiciones es necesaria la observación la distribución de los residuos y su relación con la variable dependiente (González, 2010). Si las regresiones son de una sola característica, unidimensional, y su relación es una recta, la función sería del tipo: $\hat{y} = w_0 + w_1x_1$, en el que w_0 se define como el valor del punto de corte de la línea de regresión con el eje de la variable dependiente y w_1 la pendiente o coeficiente de regresión. Si la variable presentara n dimensiones

y su regresión sea múltiple la función de predicción sería del tipo $\hat{y}_i = x_i w = w_0 + w_1 x_{i,1} + w_2 x_{i,2} + w_3 x_{i,3} + \dots + w_n x_{i,n}$.

Encontrar la función en la que la diferencia de los cuadrados de los resultados observados y predichos sea el menor, es correspondiente a encontrar los coeficientes de regresión w , por lo que se necesita diferenciar la ecuación $E(w) = \sum_i (y_i - x_i w)^2$.

Después de resolverlo, en una notación plenamente matricial $w = (X^T X)^{-1} X^T y$, en la que la matriz w , se expresa como función lineal de la matriz de datos observados y , asumiendo que $(X^T X)$ tiene inversa para todo (González, 2010).

2.5.2. Redes Neuronales

Las Redes Neuronales Artificiales (RNA), se inspiran en el funcionamiento del sistema nervioso biológico, pretendiendo replicar la sinapsis de las neuronas en sistema nervioso. Intenta emular el comportamiento del cerebro y los conceptos acerca de él.

La simulación funciona como un sistema en paralelo, siendo eficiente en la resolución de muchos problemas de aprendizaje (Juan Jesús Romero, 2007). El interés que este paradigma resulta de la habilidad de aprender automáticamente y su capacidad de funcionar de forma aceptable en presencia de información inexacta o deteriorada.

En el sistema nervioso biológico existen neuronas y en cada una de éstas, se presentan tres tipos de componentes con particular importancia en la comprensión del funcionamiento de una neurona artificial: Dendritas, Soma y Axón.

La neurona puede estar conectada a centenas o miles de otras neuronas, la neurona es constituida de un cuerpo celular del cual salen varias ramas denominadas dendritas, y una rama aislada más larga llamada axón. Las dendritas se interconectan a los axones de otras neuronas en uniones conocidas como sinapsis y se propaga entre las neuronas a través de las dendritas y de una compleja reacción química que libera sustancias químicas que hacen incrementar o descender el potencial eléctrico de un cuerpo celular.

Entonces, una red neuronal es un proceso de información con determinadas características de desempeño en común al de las redes neuronales biológicas. Las RNA, consisten en un conjunto

interconectado de unidades de procesamiento llamado neuronas, cuya funcionalidad apenas se asemeja al de una neurona biológicas. En la Figura 6, se observa que cada neurona artificial tiene un conjunto de conexiones al exterior con funcionalidad para múltiples entradas y salidas. Cada conexión tiene un peso asociado que se convierte en el principal medio de almacenamiento de información en una red.

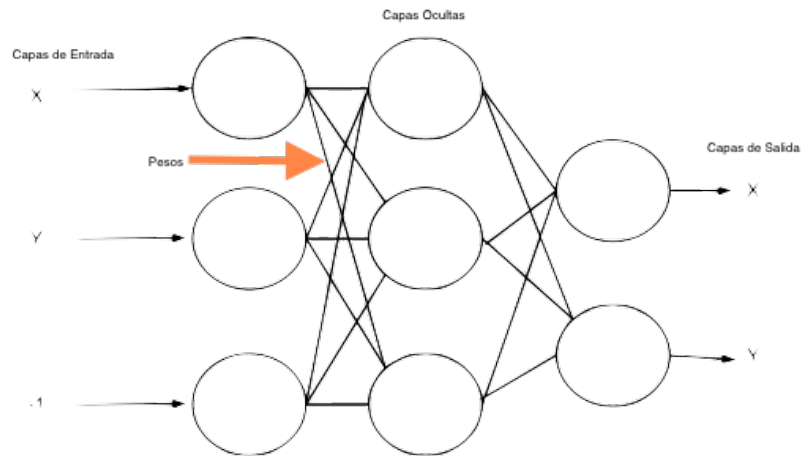


Figura 6: Representación del funcionamiento de un Red Neuronal Artificial

Cuando se pretende construir una red neuronal es necesario definir cuantas neuronas se van a utilizar, el tipo de estas neuronas y como es que las neuronas van a estar conectadas. Posteriormente se inician los pesos de entrada en la fase de entrenamiento. Durante esta fase los pesos van a ser modificadas hasta que las salidas de la red funcionen como se pretende obteniendo el mejor resultado posible.

En cada neurona el funcionamiento es simple: recibiendo valores de la sinapsis de las conexiones de entrada, calcula un valor de activación y es enviada a las conexiones de salida. Usualmente, el valor de activación se calcula en función de una sumatoria pesada de los pesos (cada entrada se multiplica por el peso de la ligación donde pasa). Este proceso se repite de neurona en neurona a través de toda la red.

La función que actúa sobre el cálculo de los pesos es denominada como función de activación. En la Figura 7, se muestran las funciones de activación más comúnmente usadas en las neuronas, (a) corresponde a la función escalón toma valores en un conjunto finito de valores, (b)

corresponde a una función señal que es una función par que extrae el signo de un número real y (c) corresponde a la función sigmoidea que es una función que usualmente presenta valores estables en los extremos y un clímax de crecimiento en una parte intermedia.

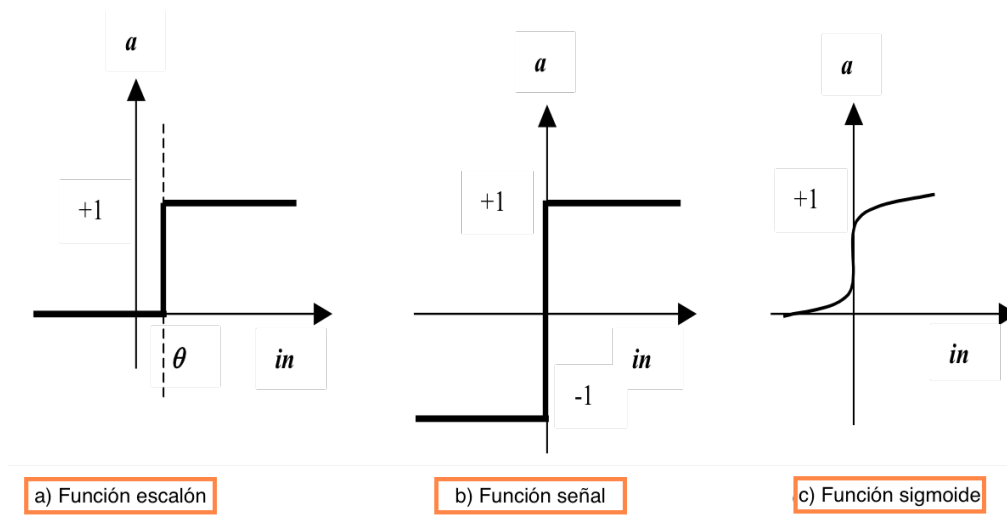


Figura 7: Tipos de funciones de activación de una neurona

Tal como el cerebro, las RNA tienen la capacidad de aprendizaje. Existen varias formas de entrenar una RNA. En una de las más utilizadas y la que es utilizada en este trabajo se procede, de forma sucinta, así: Se presentan sucesivamente ejemplos de entrenamiento como entrada de la red. Para cada ejemplo, se calcula la salida de la red e a continuación, se modifican los pesos de la red con base en la diferencia entre la salida verificada e la salida deseada para la instancia. Este proceso se repite hasta que el error de la red sea suficientemente pequeño para todas las instancias de entrenamiento. Después de este proceso se testa la red con un conjunto de ejemplos de teste para verificar se la red tiene la capacidad de generalización.

2.5.3. Support Vector Machines

Las máquinas de soporte vectorial del inglés (*Support Vector Machines*, SVM) son un conjunto de algoritmos de aprendizaje supervisado, estrechamente relacionados a problemas de clasificación y regresión. Una de las ideas simples detrás de este algoritmo es que, dado un conjunto de puntos conocidos, de dos posibles categorías, buscar predecir si un nuevo punto, del cual, se desconoce la categoría, pertenece a una categoría o a otra. Para la implementación de

este trabajo, debido a la distinta naturaleza su acercamiento es intentar la resolución adaptando SVM a funciones de regresión lineal, su funcionamiento se asemeja mucho al de una red neuronal ya que utilizan perceptrones multicapa y funciones del Kernel.

SVM produce un modelo que puede ser expresado en un número pequeño y limitado en términos de vectores y ser aplicado a funciones no lineales usando Kernel. Tal como en regresiones lineales, la idea básica consiste en encontrar una función que se aproxime bien a los puntos de entrenamiento intentando reducir el error. Su mayor diferencia frente a la técnica de regresión lineal es que todas las desviaciones especificadas por el usuario son descartadas. También, al reducir el error, paralelamente se pretende maximizar la simplicidad de la función. Otra diferencia es que principalmente se pretende minimizar el error absoluto antes que el error cuadrado.

Usualmente la función de SVM puede ser expresada como:

$$x = b + \sum_{i \text{ is support vector}} \alpha_i \mathbf{a}(i) \cdot \mathbf{a}$$

Ecuación 1: Ecuación de función kernel de Support Vector Machine

En la Ecuación 1 el producto punto puede ser reemplazado por una función de Kernel o una función de decisión para problemas no lineales. Siendo x el número de fallos totales, es decir el número de veces que α_i está mal clasificado, el resultado de la sumatoria es el producto punto de $\mathbf{a}(i) \cdot \mathbf{a}$, el algoritmo modifica directamente el valor de x .

El funcionamiento habitual de la técnica es definir por un parámetro de error un tubo alrededor de una función lineal cuyo error es ignorado, formando un cilindro en los puntos de entrenamiento y el resultado de la función es el tubo más plano que los encierre; el algoritmo depende de un parámetro de complejidad C : mientras mayor sea el valor de C , de mejor forma la función se ajustará a la distribución de la función numérica.

2.6. Resumen del capítulo

En este capítulo se resumen algunos de los proyectos desarrollados con similar con total o parcial enfoque a este trabajo, se da un breve enfoque de las variables climatológicas que influyen en la

producción agrícola, específicamente producción de rosas. En cuanto al desarrollo de software se da un breve resumen de la situación del desarrollo de aplicaciones nativas tanto en iOS como Android y su principal diferencia frente al desarrollo con *frameworks* de *crossover*. Finalmente, las últimas dos secciones dan una rápida recopilación de minería de datos y las técnicas que son utilizadas para el desarrollo de este trabajo.

CAPÍTULO III

3. Arquitectura y tecnologías utilizadas

El presente capítulo dispone la arquitectura general del proyecto y enfatiza los puntos de interés para el desarrollo de esta investigación. También expone el uso de las herramientas, una breve descripción de las ventajas de los IDEs y *frameworks* seleccionados para la implementación de la solución (WSN, aplicación y algoritmo predictivo), además la motivación y su aporte en el alcance del mismo.

3.1. Arquitectura general

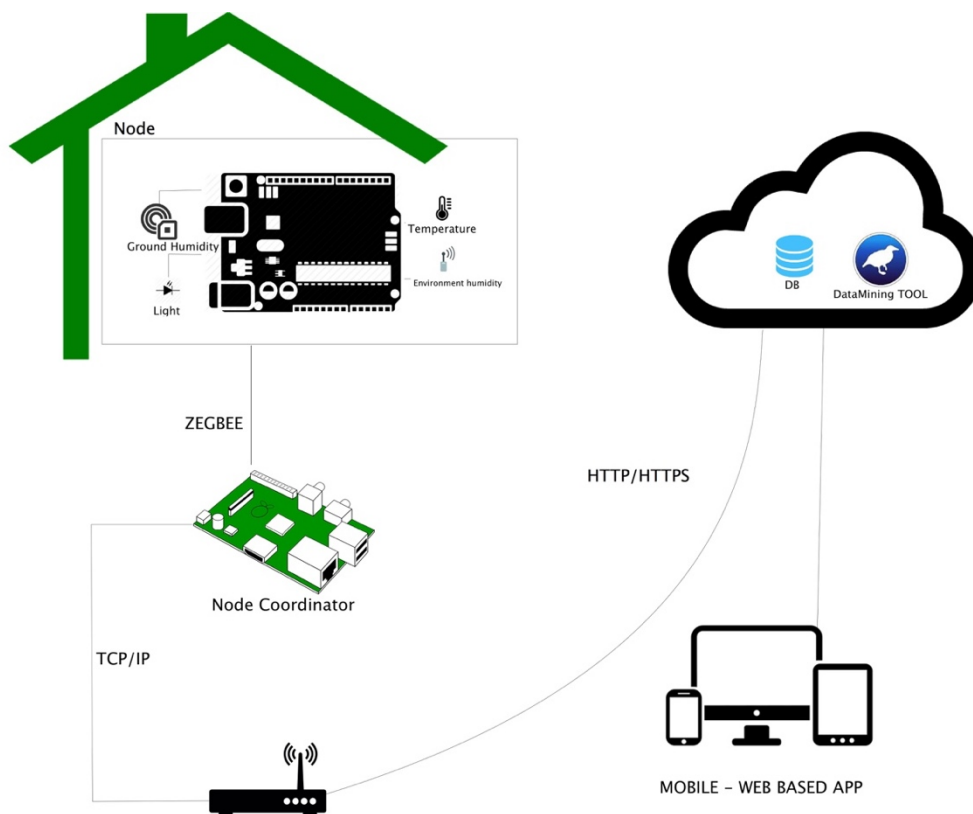


Figura 8: Arquitectura General del Proyecto

En la Figura 8, se muestra la arquitectura general del proyecto. La primera instancia del proyecto constituye en la elaboración de la WSN dispuesta en un invernadero pasivo. La implementación de la WSN, se compone de dos tipos de nodos un nodo enrutador encargado de coordinar transformaciones sobre los datos y uno o varios nodos de datos encargados de recolectar lecturas de los sensores.

A los nodos recolectores (*nodes*) se les proporciona conexión a tres tipos de sensores que permiten hacer cuatro tipos de lecturas de las condiciones ambientales - temperatura, humedad relativa de ambiente (estos dos son leídos en el mismo sensor), humedad de suelo y luminosidad. Todos los sensores están conectados a una placa computadora compuesta de al menos un microcontrolador.

La placa de circuitos se comunica a través del protocolo inalámbrico ZegBee hacia un nodo coordinador. En este nodo está implementado una secuencia de operación sobre la estructuración de los datos y como han de ser dispuestos para ser enviados a la Nube. Este microcontrolador está dotado de una dirección IP y acceso a la red gracias a un enrutador. A través del protocolo HTTP, los datos recolectados son liberados a la nube a un repositorio de datos.

La segunda parte de este trabajo consiste en la implementación de una aplicación que permite que el utilizador sea capaz de realizar un monitoreo sobre los datos y acceder a diferentes acciones para organizar y visualizar los datos. Mediante el uso de un de un protocolo de bajo nivel como HTTP o sockets, los datos son desplegados en el aplicativo capaz de ejecutarse en plataformas web y móvil.

La tercera y última parte de este trabajo es la comunicación con el API de predicción desarrollada con técnicas de *machine learning*, y se encuentra disponible en otro servidor en la nube. Este API ejecuta el cálculo sobre los datos y se comunica con la aplicación para que los datos resultantes de la tendencia de la predicción puedan ser mostrados al usuario.

3.2. Herramientas y frameworks

Las herramientas consideradas para el desarrollo de este prototipo son licenciadas bajo GNU y otras provistas por el Instituto Politécnico de Leiria, con una licencia de carácter educativo y sin fines de lucro.

3.2.1. IntelliJ IDEA 2016.2.3

El IDE IntelliJ IDEA, publicado y distribuido por JetBrains (JetBrains, 2016), es uno de las suites más completas y con mayor número de utilidades y herramientas para agilizar el proceso de desarrollo JAVA. Aunque es una suite licenciada, IPL provee de una licencia educativa.

Además de brindar numerosas prestaciones gracias a su API de inferencia en rutinas simples y cotidianas como refactorización y tratamiento de datos a través de expresiones (RegExp), implementa ciertas funcionalidades orientadas a un desarrollo basado en pruebas TDD con inspectores en JUnit (API de pruebas en Java); también elimina la latencia el momento de escritura sobre el IDE y está integrado de manera automática con los repositorios de fuentes y de versión más usuales y de distribución libre como GitHub y Gradle. Finalmente, optimiza el JVM (Java Virtual Machine), de forma que la construcción en memoria en el proceso de depuración (*debugging*) utilice la menor cantidad de recursos posibles acelerando el proceso de desarrollo.

La razón de su uso en la implementación del aplicativo es la facilidad que brinda durante el proceso de desarrollo, ajustándose a los objetivos definidos para este proyecto. Con IntelliJ IDEA se desarrolla el API de Predicción, integrando las funcionalidades que WEKA provee y orientándose al procesamiento de los datos crudos recogidos por sensores y exponiendo la predicción como un web-service (WS), para el uso y renderización en el aplicativo móvil.

3.2.2. WebStorm 2016.2.3

El IDE WebStorm, publicado y distribuido por JetBrains (JetBrains, 2016), es uno de las suites más completas para JavaScript y con mayor número de utilidades, compiladores y herramientas para agilizar el proceso de desarrollo en web. Como pertenece a JetBrains, también es una suite licenciada: la misma licencia educativa de IPL funciona en éste IDE.

El lenguaje JavaScript, ha ido evolucionando durante los últimos años de ser un lenguaje de cohesión visual de los elementos HTML en un navegador a un entorno depurado de desarrollo. Tal ha sido su auge que es posible realizar un sistema o aplicativo integro con la implementación de uno o varios *frameworks* que los soportan las últimas versiones.

Aunque no es necesario un IDE para desarrollar en JavaScript, al ser éste un lenguaje interpretado en tiempo de ejecución, WebStorm provee de funcionalidades que evitan el proceso basado en prueba-error y la experiencia del desarrollador. Las características principales son el soporte de Flow, API que permite detectar errores de tipos de datos, analiza y completa ayudando a tareas de refactorización; permite una rápida migración y sugerencias con la última especificación de JavaScript (ECMAScript6). Finalmente, brinda soporte a una gran cantidad de *frameworks* actuales de JavaScript, lo que reduce la curva de aprendizaje al cada uno manejar sus propias notaciones.

El uso de WebStorm en el presente proyecto es la implementación del aplicativo móvil basado en la integración con el *framework* de desarrollo de *crossover* Meteor, gracias a la integración con los *frameworks* (Blaze, React, Handlebars y CordovaJS) necesarios que requiere el motor de Meteor ofreciendo plantillas y sugerencias.

3.2.3. PyCharm 2016.2.3

El IDE PyCharm, publicado y distribuido por JetBrains (JetBrains s.r.o, 2016), es una de las suites más completas para Python, dispone de un gran número de utilidades, compiladores y herramientas, para la versión del año 2017 (presente), cuenta con un conjunto de librerías que agilizan el desarrollo orientado a pruebas. Como pertenece a JetBrains, es una suite licenciada, la misma licencia educativa de IPL proporciona la licencia funcional para este IDE.

Python es un lenguaje de programación definido como funcional, está diseñado específicamente para simplificar la complejidad en la sintaxis de los lenguajes tradicionales. Además, ofrece una cantidad de dominios de aplicación como desarrollo para campos como: Web, GUI, Administración de sistemas y Ciencia de datos. Este último campo es la base sobre la que se posiciona la preparación de los datos crudos para ser procesados.

Python, actualmente cuenta con dos versiones oficiales, 2.7.13 y 3.6.1. El último *release* de la versión 2 de Python fue entregado en 2010, sin embargo, aún existen muchos *frameworks* (e.g., Pandas) que fueron desarrollados para ésta versión y siguen teniendo soporte de la comunidad.

3.2.4. Framework Meteor 1.4

Meteor es un *framework* de distribución libre implementada en JavaScript *full-stack* de código abierto que permite desarrollar aplicaciones web y móviles. Meteor está provista de un conjunto de tecnologías clave para construir aplicaciones cliente conectadas reactivamente, complementados con paquetes de las tecnologías de servidor como Node.js y JavaScript (Meteor Developers, 2016).

Entre las características más representativas está la capacidad que ofrece al desarrollar e implementar todos los ambientes utilizando un solo lenguaje como JavaScript. También utiliza el concepto de *data on the wire*, en la que el servidor únicamente maneja data y no HTML, por lo cual el cliente solo debe renderizar los resultados. Otra característica es el comportamiento reactivo que permite que el UI refleje el verdadero estado de los datos con un mínimo esfuerzo en desarrollo.

El *framework* está provisto de diferentes paquetes como Apache Server, el último estándar de JavaScript con EcmaScript 6, Outh para autenticaciones y librerías de uso convencional como JQuery, Underscore, Webapp, entre otras. La funcionalidad de aprovisionamiento para el uso de librerías es delegado a NPM, que es, un administrador de paquetes y dependencias para Node.js, diseñado específicamente para proyectos de JavaScript (Npm Inc, 2017).

Los principios de Meteor son: Datos como hilos, un solo lenguaje, base de datos en cualquier lugar, compensación de latencia, reactividad en toda la pila, simplicidad relacionada a productividad (Meteor Developers, 2016). En el lado del cliente MeteorJS usa Handlebars para plantillas HTML, puede ser utilizado con ReactJS o AngularJS como *framework* del patrón *Model-View-Controller (MVC)*. Cualquiera de estos dos frameworks tienen la capacidad de ser reactivos, es decir cualquier modificación es visualizada en tiempo real sin realizar peticiones innecesarias.

Dentro de las principales consideraciones para diseñar aplicaciones con Meteor, se encuentran condiciones como la única base de datos soportada es MongoDB, lo que implica que no existe soporte a SQL (Wodehouse, 2017), por lo que requiere de mucho esfuerzo acoplar Meteor a una solución ya existente. Si la aplicación tiende a crecer exponencialmente es necesario evaluar parámetros como número de usuarios y tráfico de datos.

La razón del uso del framework Meteor es por las altas prestaciones que dispone; además de su integración nativa con *CordovaJS*, que permite el desarrollo de aplicaciones móviles utilizando las librerías de Chromium propia de los dispositivos móviles, el aplicativo *Greenhouse Prediction* es implementado con las prestaciones expuestas.

3.2.5. Weka

Weka (Machine Learning Group at the University of Waikato, 2017), es una herramienta desarrollada por la universidad de Waikato en Nueva Zelanda, un software de distribución libre potente que contiene instrumentos de pre procesamiento de datos, clasificación, regresión, reglas de asociación y visualización; además de permitir la implementación de nuevas máquinas de aprendizaje.

El sistema Weka está escrito en Java bajo de términos de licencia GNU, razón por la que se ejecuta bajo casi cualquier plataforma. El Explorer (herramienta visual) ofrece una interfaz uniforme para aplicar algoritmos de análisis (Witten, Frank, & Hall, 2011), brindando soporte a todo el proceso experimental de Data-Mining; también es posible acceder a los códigos fuentes de la implementación ofreciendo la libertad de customizar la solución.

Weka está provista de una extensa colección de técnicas de procesamiento de datos y modelado, que permiten realizar de forma simple la aplicación de diferentes tareas básicas de minería o de aprendizaje de datos. En sus primeras implementaciones WEKA fue perfeccionado para analizar datos relacionados al dominio de Agricultura, aunque posteriormente fue extendido y usado para diferentes dominios.

El uso en este proyecto, principalmente se enfoca en las bondades que esta herramienta ofrece con la gran cantidad de algoritmos de análisis y la capacidad de trabajar con grandes volúmenes

de datos. Su implementación en el algoritmo de procesamiento de datos es usada como *core* del API de predicción (Prediction API).

3.2.6. Pandas

Pandas es una librería de Python de código abierto que provee herramientas para preparación y análisis de datos. El *framework* está escrito de forma que permita trabajar de una forma sencilla con grandes volúmenes de datos.

Entre sus principales características incluyen la capacidad de tabular tipos de datos heterogéneos como si se tratara de una hoja de cálculo o una consulta a una base de datos; para arreglos uni-dimensionales y bi-dimensionales con *Series* y *Dataframe*, respectivamente.

Pandas tiene implementaciones disponibles para Python 2.7.1, 3.4 y 3.5. Dentro de sus funcionalidades destacadas permite un manejo eficiente sobre el conjunto de datos como la manipulación de datos faltantes, en el caso de no existir las reemplaza con NaN; herramientas para lectura y escritura en diferentes formatos como CSV, Xlsx o SQL; alto rendimiento para combinar y unir conjuntos de datos; eficiente funcionamiento con series temporales con funcionalidades como: conversiones de frecuencia, cambios de formateo de fechas e indexaciones de alto a nivel a modelos más simples; también permite la modificación de cada columna en los conjuntos de datos para ser uni o bi – direccionales.

3.3. Resumen del capítulo

En este capítulo se presentó la Arquitectura General del proyecto y las decisiones de cómo esta problemática fue afrontada. También las herramientas que se utilizaron para cumplir con el desarrollo de los objetivos del proyecto y los *frameworks* que fueron utilizados en la implementación de estos, así como también la justificación de su uso.

CAPÍTULO IV

4. Implementación de la Wireless Sensor Network

El capítulo a continuación detalla el proceso de diseño e implementación de la Wireless Sensor Network (WSN), una red de nodos sensores compuesta por un nodo coordinador, nodo router y dos nodos de datos. La finalidad de contar con una red de datos propia es que sea ajustada al alcance de este trabajo y su enfoque a pequeños invernaderos pasivos, siendo una implementación de bajo costo (*low cost*).

4.1. Materiales

La selección de materiales fue definida según las condiciones ambientales que son cruciales para el desarrollo de la planta y el crecimiento de una rosa. Los materiales de cada nodo son: un Xbee S2, una placa Arduino, una Raspberry PI, un módulo de tiempo y tres sensores de humedad de suelo, uno de luminosidad, dos de temperatura y humedad ambiental.

En la Figura 9, se muestra el componente Xbee S2, que mejora en relación a sus previas versiones, la potencia de salida y protocolo de datos y permite crear redes malla complejas basadas en el firmware de malla a través Xbee ZB Zigbee. La finalidad de estos módulos es permitir una comunicación fiable y simple entre microcontroladores, ordenadores y sistemas. Para la comunicación entre los nodos de datos y nodos controladores se utiliza el protocolo ZigBee por la simplicidad que aporta el momento de mandar tramas cortas de datos y la optimización de la vida útil de sus baterías. Este protocolo es ideal para redes WPAN (*Wireless Personal Area Network*) y cumple con el estándar ISO 801.15.4.



Figura 9: Componente Xbee S2

En la Figura 10 se muestra el sensor de temperatura y humedad de ambiente, el sensor DHT22, que ofrece una mayor precisión en su lectura y un mejor rango de alcance, frente a sus competidores y sus previas versiones como DHT11. Dentro de sus características el valor de consumo de energía no es alto y oscila entre los 3.3v – 5.5v. Aporta con resolución decimal con exactitud de dos caracteres y una de las condicionantes es que su tiempo de muestreo es de aproximadamente dos segundos, siendo este tiempo el valor que delimitara la periodicidad de la trama. Sus valores son en temperatura en °C y porcentuales para la humedad de ambiente.

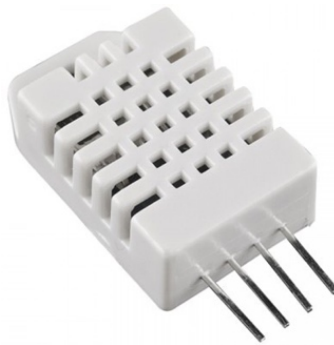


Figura 10: Sensor de temperatura y humedad de ambiente DHT11

El sensor FC-28, en la Figura 11, es el escogido para la medición de la humedad de suelo, es comúnmente usado en soluciones de sistemas de riego. Este sensor detecta cuando es necesario activar o desactivar el sistema de bombeo. Los valores dan su valor más bajo cuando el sensor es completamente sumergido en agua y el más alto cuando es completamente seco.

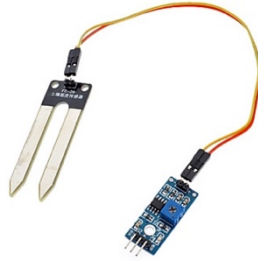


Figura 11: Sensor de humedad de suelo FC-28

En la Figura 12 se muestra el sensor elegido para analizar la intensidad de luz del ambiente, el BH1750, que es una versión mejorada de un sensor de luz típico basado en un LDR (*Light-Dependent Resistor*), el cual simplemente entrega un valor analógico. Este sensor es usado para medir la cantidad de luz por metro cuadrado que hay en el ambiente; los valores son expresados en LUX.



Figura 12: Sensor de luminosidad BH1750

El módulo de reloj seleccionado para tener un control de cuando los datos son tomados es el DS3231. Entre sus características, su exactitud oscila entre $\pm 2\text{ppm}$ a temperatura de -40°C a $+80^{\circ}\text{C}$, cuenta con memoria EEPROM y una batería de respaldo. El reloj incorpora un sensor de temperatura con un error de $\pm 3^{\circ}\text{C}$ (que no es usado en este proyecto).

La Tabla 1 resume el rango de valores de cada sensor y su unidad de medida. Con los datos que son recolectados entre estos valores cada dos segundos se construye una trama de datos que es enviada de cada uno de los nodos de datos al nodo controlador. La trama es un *string* definido de la siguiente forma: **#Trama_Nodo_Hora_HuSu_Lux_HuAm_Temp**, compuesta de izquierda a derecha: número de la trama, número del nodo, hora, lecturas de humedad de suelo, lecturas de luminosidad, lecturas de humedad ambiental y lecturas de temperatura

PARÁMETRO	UNIDADES	MÍNIMO	MÁXIMO
NÚMERO DE TRAMA	Entero	1	n
NODO	Entero	01	n
HORA	HHMMSS	000001	235959
HUMEDAD DEL SUELO		0	1023
SENSOR LUZ	LUX	1	65535
HUMEDAD DEL AMBIENTE	% (\pm 5% error)	0%	99.9%
TEMPERATURA	°C	-40°	80°

Tabla 1: Rango de valores de los sensores

4.2. Diseño de los Nodos

La WSN fue diseñada como una red tipo malla a base de los tres tipos de nodos siguientes: nodo de datos, nodo router y nodo coordinador. El nodo de datos en su primera versión, fue construido con una placa Arduino mega y con un dispositivo Xbee S2 pro, que permite la transmisión de los datos utilizando un protocolo de cifrado, a través del nodo coordinador y el nodo router, que son los medios encargados de la transmisión de la información recolectada y la compactación de la trama de información. Las variables de temperatura y humedad ambiental, humedad de suelo y luminosidad consideradas para esta experimentación requieren que el prototipo utilice sensores para estas características, considerando sus factores mínimos y máximos detallados a continuación: Este prototipo consta de dos ejemplares de sensor por cada tipo abarcando un radio de cobertura experimental de 2.8 m a la redonda, medida que fue adoptada por la cobertura que se obtiene de los sensores de humedad de suelo que es de aproximadamente 60 cm. Mientras que los demás sensores como el de luz, temperatura y humedad relativa presentan un espectro mayor, se mantiene el de menor medida para que los datos sean coherentes en ambos nodos recolectores.



Figura 13: Implementación de WSN en Invernadero de IASA

Para el envío de los datos se recolectó, unificó y estandarizó una trama conformada por una cabecera y un cuerpo de mensaje. Esta información es organizada, estandarizada, compactada y enviada al nodo router. En éste, se reciben todas las cadenas de los diferentes nodos emisores y se remiten al nodo coordinador de información. En éste dispositivo la cadena es receptada, desempaquetada, procesada y separada. Al concluir éste proceso, el nodo realiza una petición HTTP al servidor de base de datos (MongoDB) externo por medio de un servicio web REST, que contiene los métodos necesarios para realizar operaciones de base de datos NOSQL para interactuar con la información. Cada uno de los nodos ejecuta un *script* de *bin/bash* en el caso de que este inactivo por fallas eléctricas sea capaz de iniciar la comunicación por sí mismo. Finalmente, cada uno de los nodos recolectores de datos fue construido como se muestra en la Figura 14.

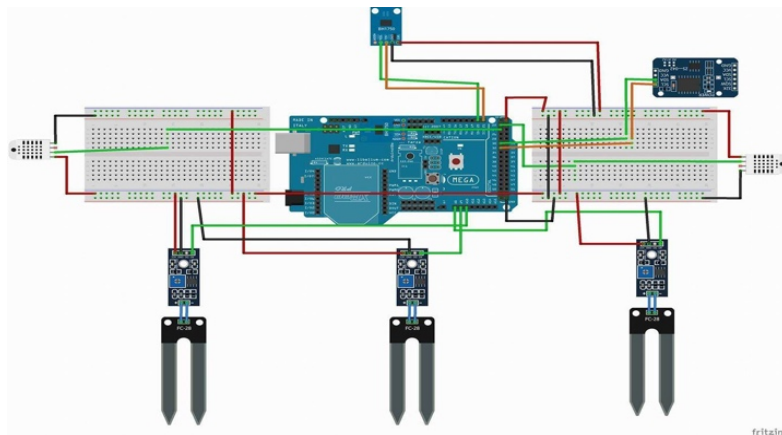


Figura 14: Implementación de Nodo Recolector

4.3. Experiencias

En este capítulo se expone las experiencias que se fueron produciendo durante la implementación del nodo. Además, se enfoca en presentar los problemas que fueron sucediendo y como el equipo decidió interactuar para resolverlas, como también las prioridades que fueron surgiendo tanto en la etapa implementación física de los nodos como las diferentes interfaces de comunicación a la nube.

La implementación del nodo enrutador de mensajes fue la primera en ser implementada con la finalidad de constatar que la comunicación era posible. En las primeras versiones del nodo se confirmó la operatividad de la prueba de concepto de que el nodo era capaz de transmitir datos de manera constante a un servidor. Posteriormente, fueran siendo acopladas nuevas implementaciones para accionar sobre los datos. La codificación del *script* no presentaba un desafío muy grande, debido a que era un proceso de algoritmia simple sobre tipos de datos *string*.

4.3.1. Conectividad

Uno de los mayores inconvenientes se fue presentando por temas de conexión y proxis, como reservar una IP estática y publica en la red de la Universidad de las Fuerzas Armadas. La gran cantidad de datos que eran transmitidos desde el nodo al servidor ocasionaba que los sistemas de monitoreo de redes alertaran de vulnerabilidades, para lo que se pensó inicialmente agrupar datos

en el nodo contando las recurrencias de una trama con datos cercanos. Por ejemplo, si la medida era 27.01°C, agruparla si la medida previa era 27 °C, aunque finalmente esta idea fue desechada debido a que la pérdida de datos comprometía la confiabilidad de cualquier resultado obtenido en un posterior análisis para minería o aprendizaje de datos.

4.3.2. Autosuficiencia

El segundo paso con el que se decidió proceder tomó lugar durante la implementación del nodo de datos, presentando muchos retos para el equipo de trabajo. Uno de los desafíos fue proveer al nodo de la capacidad de ser autosuficiente en términos de energía, debido a dos situaciones que fueron identificadas: la primera, la distancia que podría tener en las diferentes zonas del invernadero a la toma de corriente implicaba largas extensiones de cables y, la segunda, que durante la época de lluvias es común que haya fallas de energía sobre todo en zonas rurales como el invernadero del IASA y el tiempo en el que la planta de energía entra en funcionamiento podría ser un problema de no ser lo suficientemente rápida.

Para esto, se pensó en un banco de energía solar, sin embargo, el banco no podía dotar al nodo de la energía suficiente por el consumo de energía diario, una vez que requería que el sol sea reflejado perpendicularmente al banco durante todas las horas de luz que tenga el día, en Ecuador 12 horas. Se contactó un equipo de investigación canadiense en la rama de electrónica, mismo que se encontraba trabajando en un prototipo de estas características. No obstante, aunque el equipo de aquel prototipo prestó interés, no fue posible la integración con esta tecnología por temas logísticos.

4.3.3. Cobertura

Inicialmente las primeras versiones de la implementación del nodo de datos fueron probadas en un laboratorio y utilizaba un sensor de cada tipo. Sin embargo, la cobertura del espectro de cada uno de los sensores no era muy alta y este factor incurriría, en parte, a que el nodo sea capaz de cubrir menos de un metro cuadrado de terreno aproximadamente solamente, según las experiencias obtenidas.

Por otro lado, el uso de un solo sensor de cada lectura suponía una sub-utilización de la capacidad de la placa Arduino y se realizaron nuevas conexiones con nuevos sensores. Con base en las pruebas realizadas se determinó que, con un sensor de luminosidad, 2 de temperatura y humedad ambiental y 3 de humedad de suelo se lograba un equilibrio en el espectro de cobertura y mantenía las señales desacopladas. El resultado de estas pruebas fue una cobertura de tres metros aproximadamente.

4.3.4. Componentes innecesarios

Se decidió incorporar un módulo de tiempo (reloj) para el nodo de datos. Aunque este incluía un sensor de temperatura incorporado, se decidió omitirlo, debido a que daba una menor exactitud en comparación a los sensores escogidos. Finalmente, todo el módulo de tiempo fue desechado en versiones posteriores ya que el *timestamp*, medida en UNIX Epoch Time (Computer Hope, 2017), podía ser incorporada en el nodo enrutador el momento mismo de la conversión de la trama.

4.3.5. Mal Funcionamiento

Durante la construcción del nodo de datos, varios de los sensores no funcionaban adecuadamente a lo que se pretendía inicialmente por lo que fueron cambiando para versiones superiores o bien de otro proveedor. Además, el *datasheet* de algunos sensores suponían el uso de *pin*s iguales, por lo que si se usaban otros *pin*s las lecturas podían verse comprometidas o sobrepuestas. Para mitigar este impacto se precisó del uso de dos placas de pruebas adicionales (*proto-board*), como se muestra en la Figura 14.

4.3.6. Factor humano

El factor humano, fue otro de los más complejos problemas que el equipo fue afrontando durante casi todas las etapas de la recolección. Conociendo que el IASA es una extensión de Universidad de las Fuerzas Armadas, su invernadero fue diseñado con propósitos de enseñanza en el área agrícola y agropecuaria, lo que significa que estudiantes tienen acceso al invernadero. En muchas ocasiones movían el nodo bruscamente lo que incidía en desconexiones, siendo estas eléctricas o de algún cable de conexión de un sensor al nodo. Con el fin de mitigar este problema, se procuró

que el nodo y sus elementos físicos estén escondidos o sujetos con cables o adhesivo, sin embargo, esta situación fue recurrente y ocasionó pérdida de datos.

4.3.7. Resultados

Con lo expuesto anteriormente, la primera versión del nodo de datos funcionaba correctamente, salvo los problemas de energía, ya sea por fallas eléctricas durante los primeros meses o la prueba en campo del banco de energía solar. Conjuntamente a los inconvenientes en el acceso a la red que en el IASA era restringida durante ciertas horas del día debido a la cantidad de usuarios, el departamento de gestión de redes desligaba todos los dispositivos para volver a ser conectados. En etapas iniciales esto suponía un problema ya que requería que alguien se acercara al nodo físicamente y volviera a conectar a la red inalámbrica provocando que la recolección no fuera continua en muchos tramos.

La primera versión del nodo fue probada alrededor de un mes y medio a dos meses, con la finalidad de que el equipo pueda monitorear y ver los problemas reales del equipamiento en el campo. Una vez que se consiguió mitigar problemas de mayor impacto y se redujeron los riesgos de futuros accidentes, se decidió proveer de un segundo nodo de datos con el objetivo que contrarrestara y validara las lecturas del primer nodo, ya que en el caso de que un sensor se dañara era difícil detectar si la medida era errada. El segundo nodo se ubicó a una distancia de dos metros aproximadamente del primero, obteniendo en ciertos puntos varianzas no muy alejadas el uno del otro, cantidades decimales que muchas ocasiones se debían a que una parte del invernadero podría estar bajo sombra o con el sol reflejado directamente sobre los sensores de temperatura y de luz especialmente.

4.4. Resumen del capítulo

En este capítulo se da una recapitulación de cómo fue implementada la WSN, los materiales que fueron utilizados con hincapié en las bondades de los sensores que fueron utilizados. Posteriormente, las decisiones de diseño que fueron adoptadas para la construcción de los nodos (coordinador y recolectores) y el *datasheet* para futuras replicas del nodo. Finalmente, se

resumen las experiencias y problemáticas que fueron suscitando durante las diferentes etapas de ensamblaje de la WSN y las decisiones para mitigar riesgos.

CAPÍTULO V

5. Aplicación móvil

El presente capítulo dispone la definición de la metodología de desarrollo y la descripción de cada uno de los modelos que forman parte de las diferentes fases del proceso. Del mismo modo, se describe como fue desarrollada la aplicación, que llamamos de *Greenhouse Prediction Application* (GPA), sus características principales, arquitectura, problemas y aprendizajes.

5.1. Metodología de desarrollo

La metodología UWE (UML – Based Web Engineering), que fue desarrollada por Nora Koch en la Universidad de Múnich en Alemania, procura encontrar una forma estándar de análisis y diseño de sistemas web (Rossi, Pastor, Schwabe, & Olsina). El empleo de esta metodología se basa en extender el lenguaje UML y ser ajustado a los componentes web. Su funcionalidad puede ser a priori adecuada para el desarrollo de una aplicación móvil (*crossover* basado en librerías web) y adaptable como una metodología ágil. Las etapas o fases que propone de forma generalizada son captura, análisis y especificación de requerimientos, diseño del sistema, implementación y pruebas.

5.1.1. Captura, análisis y especificación de requerimientos

Refiere al proceso de comprensión entre la funcionalidad que se pretende obtener y la que es desarrollada. Enfatiza la funcionalidad de una forma conjunta y pretende obtener una visión panorámica de lo que se pretende conseguir.

- **Modelo de casos de uso:** expresa de forma general una idea de lo que el usuario de la aplicación pretende y de las funcionalidades que el sistema persigue. Este es un paso necesario y usual entre metodologías de desarrollo tradicionales. El valor que aportan las tecnologías ágiles podría ayudar a comprender el contexto del negocio y el dominio de la aplicación de manera que no sean excluyentes con el desarrollo.

5.1.2. Diseño del sistema

El diseño del sistema está relacionado a la comprensión conceptual del sistema, de la estructuración de su implementación, la navegación y su presentación al usuario final.

- **Modelo conceptual:** expresa el dominio de la aplicación basado en la captura de requerimientos tomado de los casos de uso.
- **Modelo entidad-relación** (Base de datos): expresa el dominio de la aplicación en cuanto al almacenamiento de los datos sensibles del funcionamiento general y específico de la aplicación.
- **Modelo navegacional:** propone un conjunto de guías generales, en la que se da el espacio y los componentes que interactúan en la navegación de la aplicación.
- **Modelo de presentación:** contiene el diseño de GUIs abstractas y la interacción de usuario con la aplicación que generalmente pueden ser expresados en *mockups*.
- **Implementación y pruebas:** La implementación del aplicativo estará orientado a prácticas de programación ágil, proceso que está orientado al uso de TDD (*Test Driven Development*) y BDD (*Behaviour Driven Development*). Siendo una técnica que es practicada durante todas las etapas del desarrollo.

5.2. Modelos y diseño

En la siguiente sección se presentan los modelos correspondientes a la metodología UWE, generados durante el análisis y el diseño de la aplicación GPA, expuesta en la sección 5.1. Cada una de estas etapas puede ser desarrollada con la elaboración de modelos tales como los que se describen a continuación:

5.2.1. Modelo de casos de Uso

Las principales funcionalidades de la aplicación están descritas en la Figura 15. Se muestra que existen dos tipos de utilizadores para la aplicación: Administrador y Agricultor. La aplicación se encuentra dividida por cuatro módulos agrupados según sus funcionalidades. El primer módulo, de Inicio de sesión, describe dos funcionalidades como es el inicio y fin de sesión, y es para ambos actores; El segundo módulo de Administración es únicamente accedido por el

administrador; El tercer módulo es el de Monitoreo, únicamente accedido por el agricultor y, por último, el módulo de Predicción, cuya función principal es calcular la predicción basado en el modelo matemático que reside externa a la aplicación en un API desarrollado.

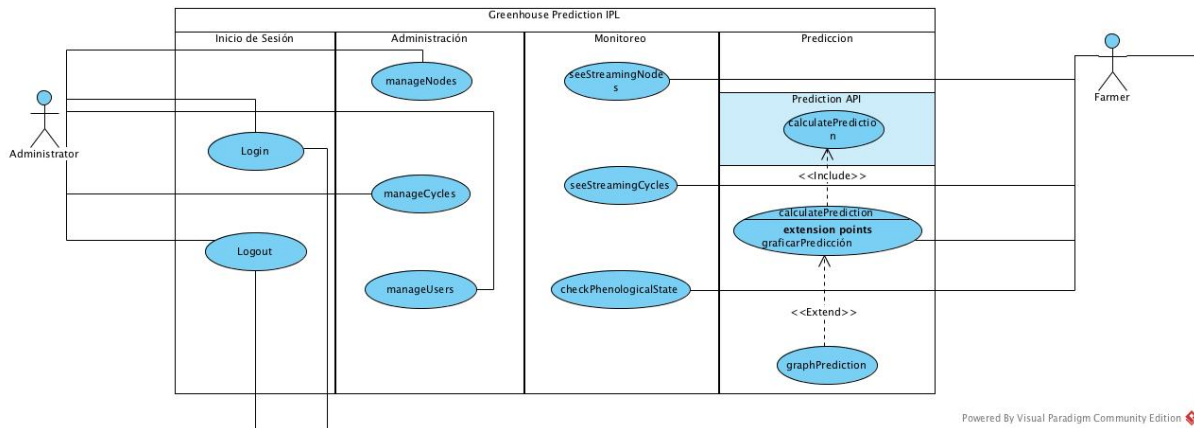


Figura 15: Modelo de casos de Uso

5.2.2. Modelo conceptual

En la Figura 16, se presenta el modelo conceptual de la aplicación móvil GPA, considerado como dominio del problema de la aplicación. Este modelo identifica como los ítems más destacados el agricultor, el cual se relaciona con el invernadero, ciclos que se componen de nodos y estos de sensores, como eje principal los datos recolectados que es usada por las funcionalidades de monitoreo y el API de predicción para arrojar el resultado buscado.

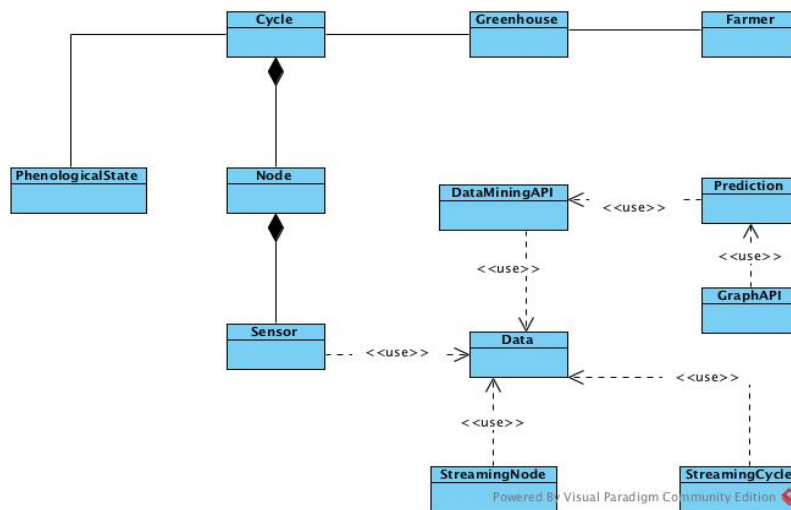


Figura 16: Modelo conceptual de dominio

El modelo conceptual brinda un panorama general del dominio de la aplicación para lo cual es necesario comprender cada una de las entidades descritas en el modelo: un agricultor (*farmer*) es propietario o empleado y se relaciona con un invernadero (*greenhouse*).

En un invernadero es usual que se lleven varios cultivos de diferentes plantas que, para el enfoque de este proyecto, es un invernadero pasivo. En 2.2 (Cultivo de rosas), se describió que cada una de estas tiene un ciclo (*cycle*) de crecimiento y un ciclo está relacionado a un estado fenológico (*phenological state*). Un nodo (*node*) es el equipo compuesto de varios sensores y estos se encargan de tomar lecturas de las condiciones ambientales del cultivo dentro del invernadero.

Cada uno de los datos leídos es considerado un registro de la entidad de datos (*data*). Estos datos son los que se presentan en el monitoreo de la aplicación con el uso de *streaming* de cada nodo y pasarán al análisis de predicción y posteriormente a ser presentados al usuario.

5.2.3. Modelo Entidad - Relación

En la Figura 17, se muestra la distribución de las colecciones. No existen relaciones entre las entidades ya que se ha diseñado el problema con una solución de un motor de base de datos No-SQL que trabaja con índices, lo que mejora el rendimiento. Las colecciones para el aplicativo agrupan y referencian a las entidades de dominio.

El uso de este modelo describe las entidades de la capa de *model* en la aplicación móvil o web. La base de datos en este desarrollo no sugiere ninguna restricción en la validez de la existencia de un registro, debido a que las colecciones de mayor importancia son únicamente manipuladas directamente por el servicio web, expuesto por el servidor de base de datos, que es consumido por el nodo coordinador.

El API de predicción no almacena datos por sí mismo y solo realiza el cálculo con algoritmos de minería de datos o técnicas de aprendizaje.

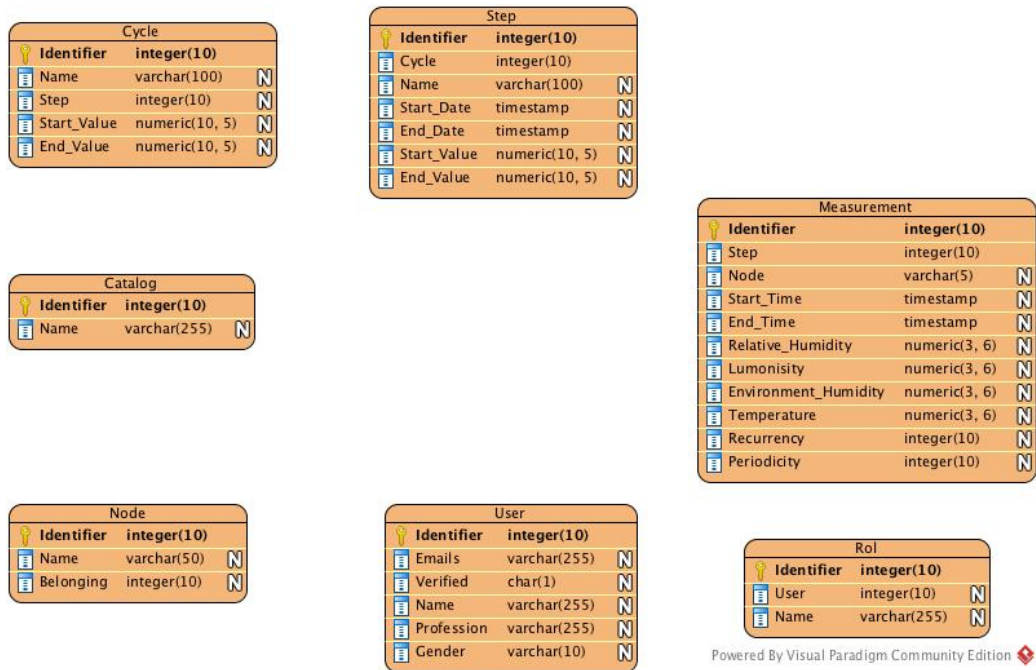


Figura 17: Modelo Entidad-Relación No-SQL

5.2.4. Modelo Navegacional

En la Figura 18, se presenta el modelo navegacional, el cual se diseña con la extensión UML, y se muestra el flujo de navegación entre las pantallas de la aplicación.

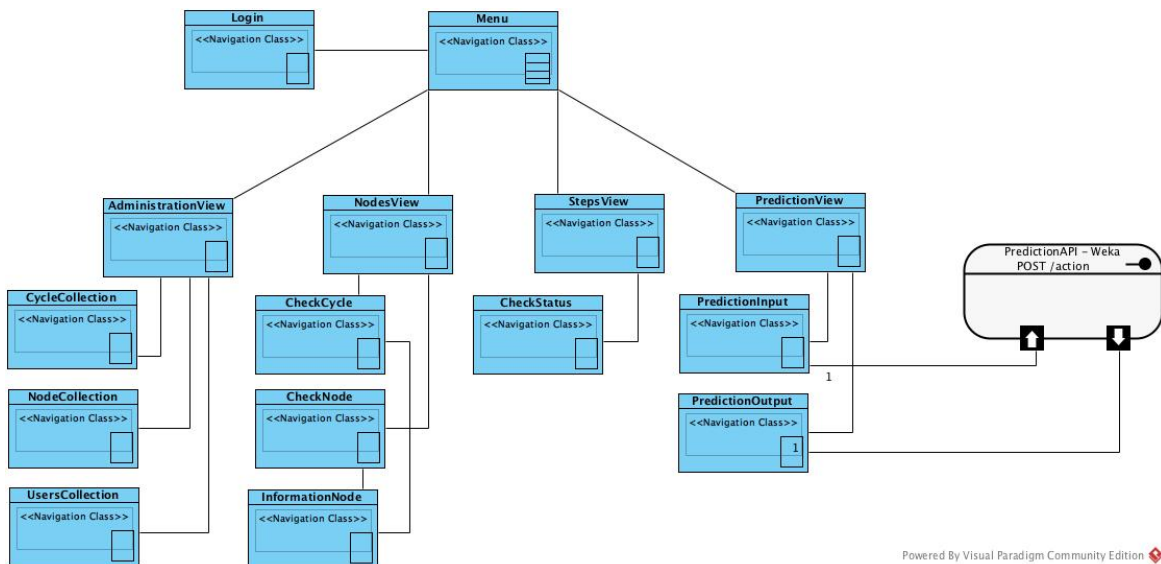


Figura 18: Modelo Navegacional

El modelo de navegación muestra la estructura del aplicativo y como este está distribuido en su implementación. El punto de ingreso al sistema es una interfaz de usuario para iniciar sesión, posteriormente accediendo a un menú en el cual existen cuatro secciones que corresponden a cuatro módulos de la aplicación. El primer módulo es un módulo de administración en el cual únicamente el administrador puede acceder el momento de crear o modificar un ciclo, un nodo o verificar que usuarios existen y están habilitados. El segundo módulo está relacionado con los nodos y permite constatar cual es la información que se encuentra en cada uno de los nodos para monitorear el estado de los mismos; Cada uno de los parámetros es modificable. El tercer módulo permite al agricultor visualizar de forma el estado de cada condición climática en el invernadero en un periodo de 24 horas del día seleccionado. El cuarto y último módulo permite a la aplicación comunicarse con el API de predicción tanto para solicitar los datos sobre los que se quieren realizar la predicción, como para obtener el resultado de la predicción de lo que sucedería en un estimado intervalo de tiempo, bajo las condiciones simuladas, y de esta forma que el agricultor consideré las posibles acciones a ejecutar sobre su cultivo.

5.2.5. Modelo Interfaz Abstracta (UI)

El diseño de interfaz abstracta, presenta un bosquejo aproximado a cómo es diseñada la aplicación. En el conjunto de figuras a continuación, se muestra los *mockups* diseñados para la aplicación móvil y web.

En la figura siguiente, se muestra la pantalla de inicio de sesión (*Log-in*). La pantalla es simple y permite la acción del usuario para ser autenticado. Conjuntamente se incluye la funcionalidad para registrarse en el sistema, utilizando correos electrónicos para confirmar. Información relevante para la aplicación puede ser incluida en esta GUI.



Figura 19: Pantalla de bienvenida e inicio de sesión

En la Figura 20, se muestra el menú principal de la aplicación mismo que será un *side-bar* menú como opción por defecto. Sin embargo, si la plataforma sea de menor resolución pasa a ser un menú en la parte inferior. Estas opciones son el punto de acceso a los cuatro módulos descritos en la Sección 5.2.1

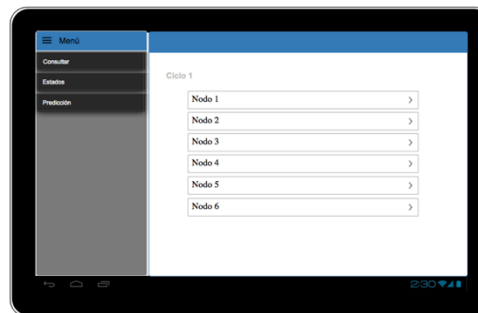


Figura 20: Menú principal

En la Figura 21, se muestra la información de los datos recolectados por un nodo. Estos datos son mostrados en un sistema de semaforización para advertir al usuario de la lectura de la sección y, de esta forma, advertir en tiempo real el estado de su invernadero. Si el usuario no estuviese observando, cuando una anomalía ocurre, aparece una notificación para informarle.

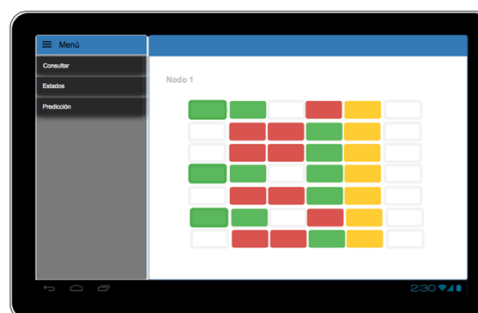


Figura 21: Pantalla de monitoreo de condiciones ambientales

En la Figura 22, se muestra información acerca del estado del parámetro capturado del nodo seleccionado y su estado en general y las posibles acciones a ejecutar en el caso de que alguna anomalía persistiera.



Figura 22: Pantalla de mensaje sobre el monitoreo de condiciones ambientales

En la Figura 23, se muestra un GUI con el resultado de la predicción, disponible en el API desplegada como servicio web, mismo que usa un API de gráficos para mostrar información resumida de la tendencia de la predicción. Esta información puede ser beneficiosa para el usuario ya que puede ver de forma resumida las condiciones de cada nodo dentro del invernadero y tomar acciones al respecto.



Figura 23: Predicción de condiciones ambientales

5.3. Desarrollo de la aplicación

La aplicación fue desarrollada usando el *full-stack framework* MeteorJS en el IDE WebStorm. El proyecto mantiene el nombre genérico de *greenhouses*, y sigue con la convención de ECMAScript6 y las guías de estilo para JavaScript definidas por (airbnb, 2017). Sin embargo, es necesario considerar que JavaScript como lenguaje ha evolucionado y las prácticas de codificación en este proyecto son las que siguen la aproximación más razonable.

5.3.1. Arquitectura de la aplicación

La aplicación está estructurada siguiendo la arquitectura clásica de un cliente – servidor y basada en micro-servicios, más ciertas especificaciones propias de MeteorJs. El servidor es una implementación que sobrepone los recursos en un servidor *express* de *NodeJs* (Node.js Foundation, 2017).

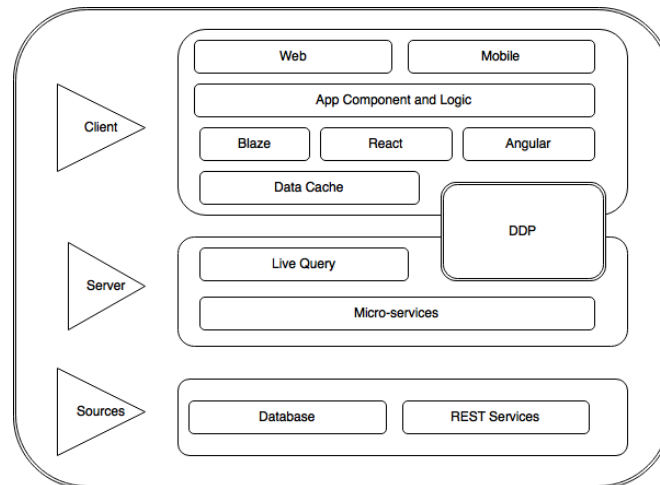


Figura 24: Arquitectura de aplicación móvil y web

En la Figura 24, se muestra la arquitectura de la aplicación, la interacción entre sus bloques funcionales y como se encuentra estructurado por tres capas *Sources*, *Server* y *Client*.

En la capa de *Sources*, se encuentran los recursos que han de ser consumidos por la aplicación y cómo funciona la conexión desde la aplicación. En esta capa se encuentran los *plugins* necesarios para interactuar con servicios web del tipo REST y la base de datos. En el caso de Meteor la única base de datos soportada es MongoDB (mayor detalle en 3.2.4).

En la capa *Server*, se definen tres principales responsabilidades: un módulo de micro-servicios en la que se encuentran todas las posibles interacciones sobre los recursos de la capa de *Sources*, el concepto que MeteorJs para manipular las colecciones de la base de datos es un modelo de publicaciones en la capa del servidor que serán expuestas para el uso exclusivo del cliente.

Con la finalidad de simplificar el concepto de publicaciones, existe un *framework* de *Live Query* que se encarga principalmente de manipular y procesar los datos en la solicitud del servidor a la

base de datos, antes de que los datos alcancen la capa de cliente. El último componente de esta capa es *Datagram Delivery Protocol* (DDP), este protocolo se encarga de gestionar como los datos han de ser expuestos y del cómo se ensamblan los datos en una trama.

La capa superior *Client*, comienza con el consumo de las subscripciones de datos a través de DDP y luego son inmediatamente almacenados en el cache de los datos. El siguiente componente se encarga de la organización y gestión de las vistas en la aplicación opcionalmente puede usar Blaze como framework de *templating* y pueden usar uno de los dos *frameworks* de MVC: AngularJs y ReactJs.

El último componente extrae los componentes de la aplicación que son visualizados por el usuario y la lógica de la aplicación como navegación, seguridad y estados. Posteriormente, aquel artefacto permite el despliegue de la aplicación en las plataformas web, iOS o Android, gracias a CordovaJs.

5.3.2. Funcionalidad de la aplicación

En la Figura 25 se muestra el menú principal de la aplicación en el que se muestra: los logotipos de las universidades que trabajan en conjunto para la completitud de este proyecto, información relevante de la aplicación y, en la parte superior, el panel para inicio de sesión o registro de usuario.

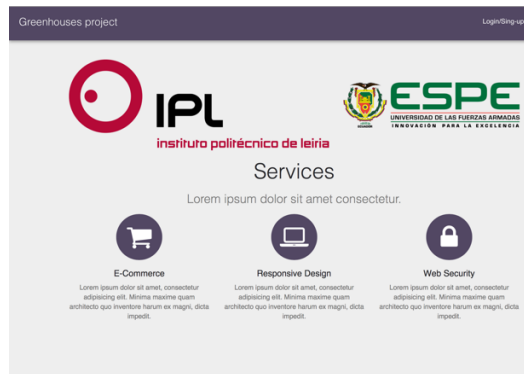


Figura 25: Implementación de la pantalla de bienvenida

La Figura 26 se muestra el panel que se despliega para iniciar sesión y el registro de cuenta. Estos solicitan los datos necesarios para la acción que se requiera hacer. Cuando hay registros de

nuevas cuentas es necesario que el correo sea válido ya que solicita una confirmación para utilizar la aplicación.

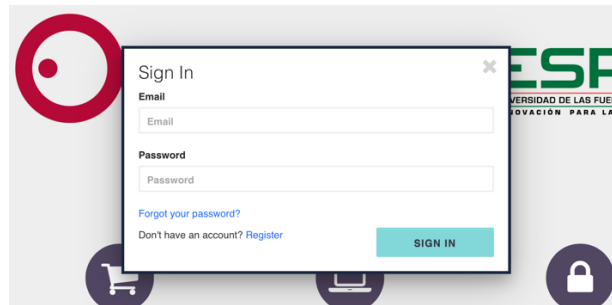


Figura 26: Implementación de la pantalla de inicio de sesión

En la Figura 27 se muestra el menú principal de la aplicación y un mensaje de bienvenida en el panel en la parte derecha, si es accedido por la web, o en la parte inferior, si es accedido por medio de la aplicación móvil. En el menú existen cuatro opciones accesibles: la administración de usuarios, comprobación de los nodos, predicción y parametrización.

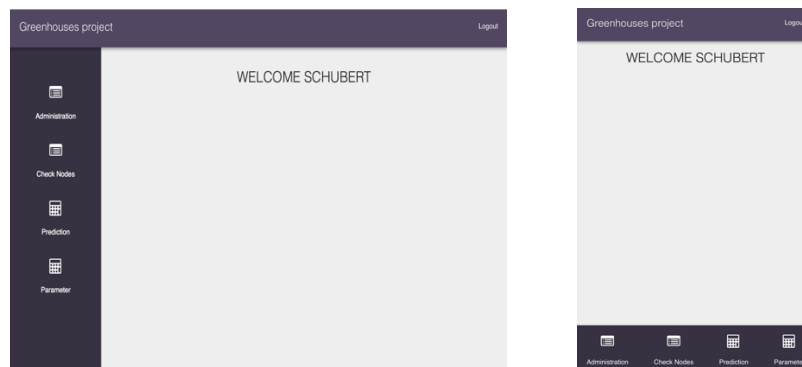




Figura 27: Implementación del menú principal

En la Figura 28 se observa la administración de los nodos. Actualmente existen dos nodos. En la parte superior del listado de nodos se encuentra un botón de añadir para añadir uno nuevo. En cada representación del nodo se pueden observar dos acciones disponibles en la sección derecha superior, la primera representada con el ícono  y la segunda con el ícono de .

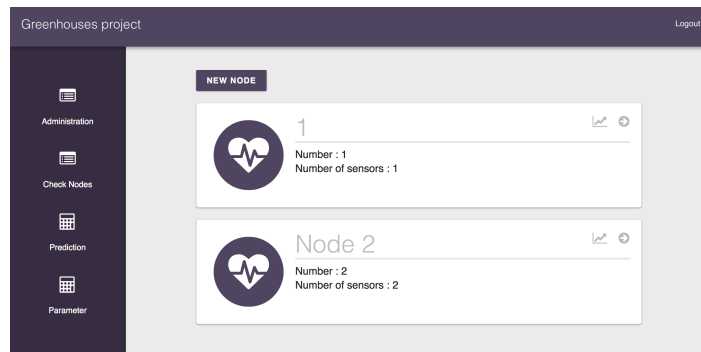


Figura 28: Implementación de administración de nodos

La acción del botón ↻, se presenta en la Figura 29, en la que se puede ver el monitoreo de cada una de las condiciones climatológicas en el nodo seleccionado. Cada una de las variables se encuentra con colores indicativos de su estado. Cuando el usuario enfoca a un registro, los colores están acorde a los parámetros óptimos y aceptables establecidos. También muestra una alerta resumiendo el detalle de la lectura y la medida promedio actual.

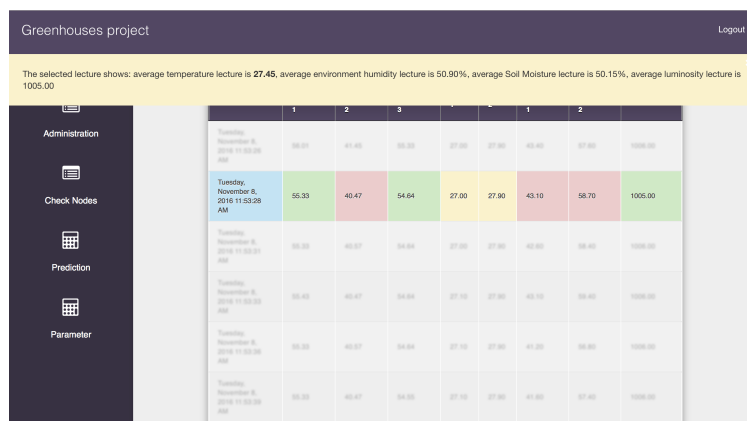


Figura 29: Implementación de monitoreo de los nodos

En la Figura 30, se puede ver una gráfica que muestra la tendencia de las condiciones ambientales en el número de sensores que componga el nodo para cada variable. Esta corresponde a la acción de 📈, durante todo un día seleccionado. Se solicita una fecha y se puede observar la disposición de las variables durante las 24 horas del día. En la parte inferior se puede seleccionar la variable que se desea visualizar y cada trazo acompañado de los parámetros óptimos y aceptables.

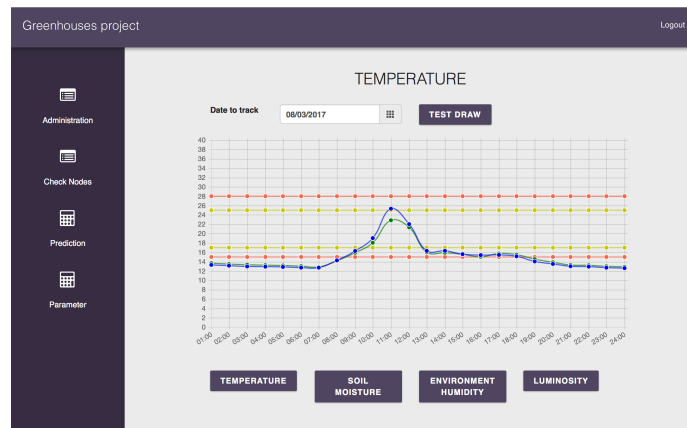


Figura 30: Implementación de monitoreo gráfico

En la Figura 31 se muestra el menú de predicción que toma el valor de la medida actual y el valor de las últimas diez medias horas del día para realizar la predicción de la tendencia en la próxima media hora. Junto a la medida se observa los parámetros aceptables y óptimos, la tendencia de esta variable en las últimas horas y el valor predicho. En la parte inferior se encuentra un panel con las variables ambientales a predecir.

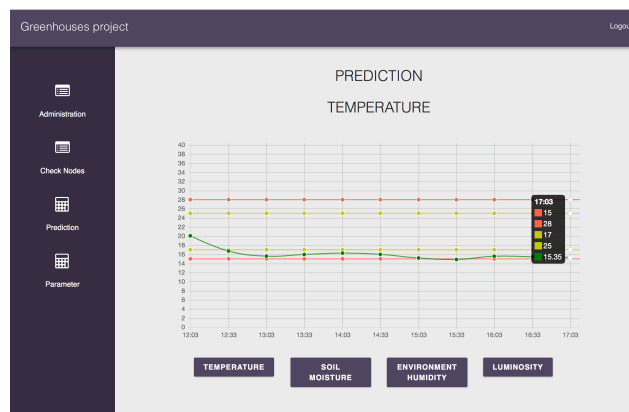


Figura 31: Implementación de pantalla de predicción

En la Figura 32, se muestra un panel con los parámetros ideales y aceptables: ideal superior, ideal inferior, límite superior y límite inferior, para cada una de las condiciones ambientales. En la parte superior del panel se encuentra un botón . Este ícono permite la edición de las variables y, eventualmente, si la aplicación fuera a ser utilizada en otro dominio.

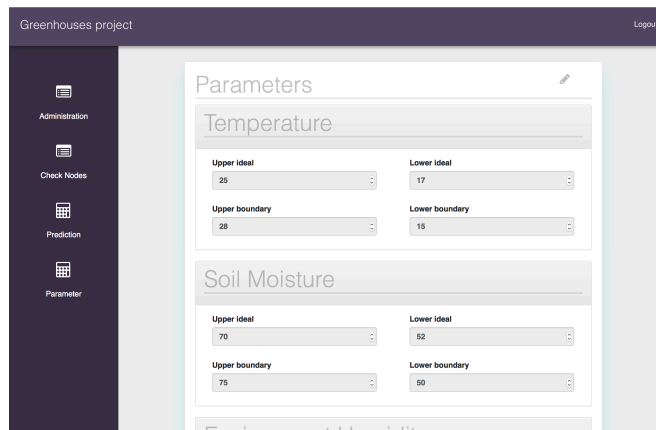


Figura 32: Implementación de pantalla de parametrización de umbrales

5.4. Pruebas y Experiencias

Durante el desarrollo de la aplicación se realizaron varias pruebas para validar su correcto funcionamiento, sin embargo, no hay una estrategia de pruebas implementada que valide una completa cobertura de los casos. Por otra parte, el código fuente de la aplicación y del API de predicción fue inspeccionado con la herramienta de análisis de código Codebeat con la finalidad de asegurar la limpieza y calidad de codificación (Codebeat, 2017). Esta herramienta calificó con 3.19 y 3.99 (la ponderación es sobre 4 puntos), a la aplicación GPA y API Prediction respectivamente (Figura 33).

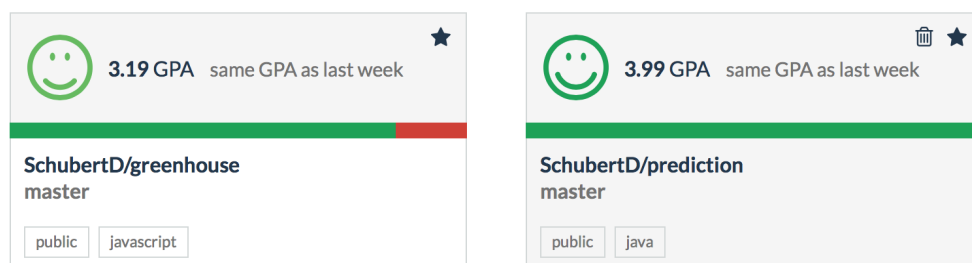


Figura 33: Análisis de código de aplicación y API

En las experiencias, se puede rescatar que debido a que, para el uso de Meteor, se necesita de otro tipo de paquetes específicos diferentes a los que se usan en aplicaciones de NodeJs. Los paquetes deben pasar por un *wrapper* para ser usados en la estructura de la aplicación. Aunque muchos de los paquetes pueden ser utilizados de todas formas en las aplicaciones, su funcionamiento ya no está adherido al ciclo de vida del *framework*.

Muchos de los paquetes que permiten el manejo de las entidades de la base de datos están ya optimizados para trabajar con Meteor, lo que facilita mucho el uso de colecciones en formularios de vista. Aunque la arquitectura singular de Meteor intenta ser intuitiva y fácil de usar, podría ser un problema si no se está acostumbrado a la misma.

La elaboración de la tabla de monitoreo fue relativamente sencillo de construir. Sin embargo, el principal desafío estaba en que la cantidad de datos que podría fluctuar en la aplicación dificultaba la lógica necesaria para ser visualizada trayendo demasiados datos a memoria. Para mitigar este posible riesgo de que la batería en dispositivos móviles sea drenada, se procesó una forma de traer en suscripciones bajo demanda para que la paginación sea procesada en el servidor.

El uso de un *framework* para utilizar *charts* y gráficas fue desafiante, debido a que fuerza a utilizar los eventos específicos de JQuery (JQuery Foundation, 2017), para ser renderizado y permitir la interacción con la gráfica. El impacto de esta desventaja del *framework* fue reflejado más en el tiempo de desarrollo por las diferentes pruebas que se debía realizar y la falta de documentación para el uso entre estas dos tecnologías, ya que algunas de las funcionalidades del *framework* de gráficas no funcionaban correctamente.

5.5. Resumen del capítulo

En este capítulo se muestra la metodología de desarrollo de software adoptando una variación de UWE y simplificando sus artefactos entregables para orientarse a *Agile*. Posteriormente, se explica la arquitectura de la aplicación con MeteorJs y como se adapta en el desarrollo. Finalmente se muestra las funcionalidades de la aplicación y muestra capturas de pantalla del funcionamiento de la misma.

CAPÍTULO VI

6. Modelo predictivo

La fase final del proyecto consistió en elaborar un modelo predictivo capaz de anticipar las medidas estimadas de las condiciones ambientales dentro del invernadero durante los próximos n minutos. Para la consecución de este hito fue necesario implementar la WSN descrita en el Capítulo 0, para conseguir lecturas de los datos a través de los sensores.

6.1. Definición del problema

En el contexto que se ha delimitado para la ejecución del presente proyecto, se establece contar con una herramienta capaz de brindar información relevante de las condiciones actuales y futuras en un invernadero, esto con la finalidad de que un agricultor pueda tomar decisiones basadas en tales predicciones.

En cuanto, al objetivo del proceso de minería se busca implementar un modelo capaz de predecir las futuras condiciones ambientales de variables como temperatura, humedad de suelo, humedad relativa de ambiente y luminosidad con base a los valores recientes de esas variables. El modelo es considerado exitoso si los valores resultantes son representativamente cercanos a las mediciones futuras del invernadero.

Finalmente, la herramienta de *Data Mining* seleccionada es Weka, descrita en la sección 3.2.5, y las técnicas seleccionadas son: Regresión Lineal, Redes Neuronales y *Support Vector Machines*; descritas en las secciones 2.5.1, 2.5.2 y 2.5.3; respectivamente.

6.2. Exploración y Preparación de los datos

Esta sección es la combinación de dos fases del *Data Mining* en el presente proyecto, debido a que durante el funcionamiento del nodo existieron problemas de pérdida de información, lo que mostraba espacios vacíos en la línea temporal de los datos y no permitiendo su continuidad. Por esta razón, se precisó de una previa limpieza de los datos antes de ser explorados.

6.2.1. Datos crudos

Los datos fueron almacenados en un repositorio *MongoDB*, disponible en un dominio provisto en el laboratorio de aplicaciones móviles de la Universidad de las Fuerzas Armadas. La colección de los datos recolectados fue llamada *NodeData* y se compone de los siguientes campos:

- Nombre de la clase (*className*) y nodo (*node*), son datos del tipo *string* y hacen referencia a la ubicación y a la referencia de la ubicación de los nodos recolectores;
- Tiempo de inicio (*start time*) y tiempo de fin, son datos de tipo *Big Integer* y almacenan los tiempos de inicio y finalización de la lectura de los sensores en formato UNIX – *Epochtime*;
- El valor de periodicidad es del tipo *Integer* y almacena el número de segundos de separación entre las medidas de los sensores (actualmente 2 segundos);
- Las lecturas de los sensores son tipos de datos *Float* con precisión decimal de un dígito, almacenan las lecturas de los sensores de condiciones ambientales: un sensor de luminosidad, 3 sensores de humedad de suelo, 2 sensores de temperatura y humedad de ambiente.

En la Figura 34, se muestra una vista de los registros de la tabla *NodeData* en la base de datos. Los valores que aquí se muestran son datos crudos leídos directamente desde el sensor y sin ningún tratamiento. Los valores corresponden a la fecha del 18 de noviembre de 2017, entre las 11:53:26 a 11:53:54.

id	className	node	start_time	end_time	TEMP1	TEMP2	ENVH_1	ENVH_2	SM_1	SM_2	SM_3	LUM	periocity
ObjectId(5822030)	com.espe.edu.	1	1478624006	1478624006	27	27.9	43.4	57.6	573	424	566	1006	2
ObjectId(5822030)	com.espe.edu.	1	1478624008	1478624008	27	27.9	43.1	58.7	566	414	559	1005	2
ObjectId(5822031)	com.espe.edu.	1	1478624011	1478624011	27	27.9	42.6	58.4	566	415	559	1006	2
ObjectId(5822031)	com.espe.edu.	1	1478624013	1478624013	27.1	27.9	43.1	59.4	567	414	559	1006	2
ObjectId(5822031)	com.espe.edu.	1	1478624016	1478624016	27.1	27.9	41.2	56.8	566	415	559	1006	2
ObjectId(5822031)	com.espe.edu.	1	1478624019	1478624019	27.1	27.9	41.6	57.4	566	414	558	1006	2
ObjectId(5822031)	com.espe.edu.	1	1478624021	1478624026	27.2	28	41.6	56.8	566	414	559	1006	2
ObjectId(5822032)	com.espe.edu.	1	1478624029	1478624029	27.2	28.1	41.5	57.4	567	415	558	1006	2
ObjectId(5822032)	com.espe.edu.	1	1478624031	1478624034	27.2	28.1	41.7	56.6	566	414	558	1006	2
...

Figura 34: Muestra de datos crudos leída por los sensores

6.2.2. Limpieza de los datos

En la Figura 35, se muestra el flujo que sigue el script de preparación de datos, que toma como entrada la lectura de los datos crudos. Los datos son posteriormente agrupados por la cantidad de

minutos y, en este proceso, se remueven posibles valores vacíos. El proceso continúa agrupando los datos en el número de h horas en mediciones de m minutos, considerando las una y todas condiciones climatológicas para cada caso. Finalmente, la salida del script son archivos en formato CSV. En las secciones siguientes se explica todo este proceso con más detalle, así como la etapa de exploración de datos.

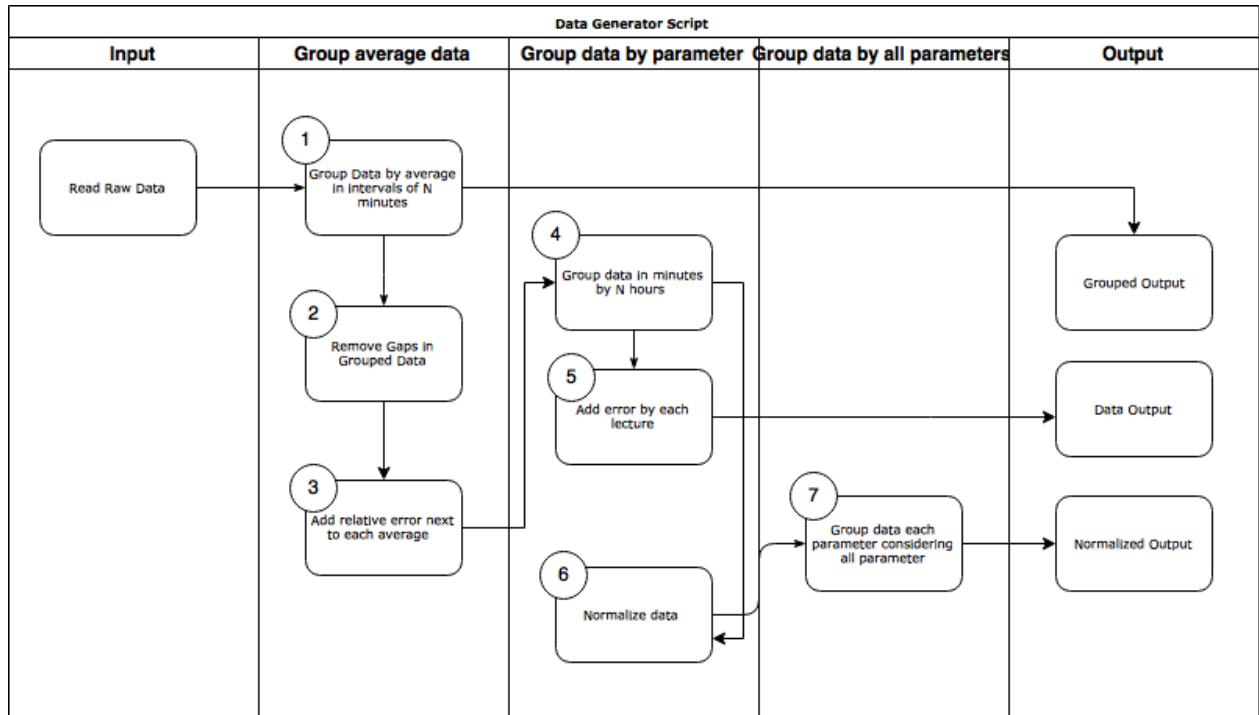


Figura 35: Diagrama de flujo del proceso de preparación de datos (Script Python)

En el primer bloque, Agrupación media de datos, se ejecuta el procedimiento de limpieza de datos. El primer proceso **(1)** corresponde a la agrupación de los datos por el número de minutos (m minutos) y promediar los resultados. Posteriormente, el segundo proceso **(2)** remueve los valores vacíos (valores con N/A). Finalmente, el tercer proceso **(3)** consiste en calcular el valor de desviación estándar de cada variable ambiental y añadirlo a cada registro. El resultado del archivo de salida CSV sería el mostrado en la Figura 36, parametrizado con $m=30$, los valores de inicio y fin de 11 de noviembre de 2016 desde 16:30 hasta las 21:00.

start_time	TEMP	ERR_TEMP	ENV_HUM	ERR_ENV_HUM	S_MOIS	ERR_S_MOIS	LUM	ERR_LUM	end_time
11/8/16 16:30	27.80204918	0.514101406	48.53483607	12.21034308	511.1338798	84.078827	1004.131148	0	11/8/16 17:00
11/8/16 17:00	27.23046472	1.009422314	46.69475043	11.32721777	500.1485944	78.93016739	1005.662651	0	11/8/16 17:30
11/8/16 17:30	27.87794613	1.306838088	46.70909091	9.86354439	486.8237935	75.08847243	1004.69697	0	11/8/16 18:00
11/8/16 18:00	28.10560498	0.557758783	45.23202847	11.70903403	475.6381969	72.7285963	1003.946619	0	11/8/16 18:30
11/8/16 18:30	27.32905983	0.757873422	45.74666667	14.76487308	468.7566952	71.20492235	995.1094017	0	11/8/16 19:00
11/8/16 19:00	25.35408526	1.087286573	49.46092362	16.75252769	467.5559503	70.85586367	985.5328597	0	11/8/16 19:30
11/8/16 19:30	19.66012891	0.51893564	68.24309392	20.94051806	469.1737262	71.49703296	917.0423573	0	11/8/16 20:00
11/8/16 20:00	18.53485772	0.445103598	75.36615854	23.27746659	470.0121951	72.81397253	926.0142276	0	11/8/16 20:30
11/8/16 20:30	17.61710526	0.3814655	78.48125	22.96934034	471.4890351	73.78759967	914.4605263	0	11/8/16 21:00
...

Figura 36: Muestra de datos organizados por $m=30$

6.2.3. Exploración de datos

Describimos ahora lo que fue hecho con relación a la exploración de los datos, etapa que se ha hecho después de la limpieza de los mismos. Esta etapa consistió en evaluar los parámetros que permitan ver la tendencia y la correlación entre los mismos para las diferentes condiciones ambientales. Para esta fase se consideró el set de datos parametrizado con $m=30$.

Correlación				
TEMP	1			
ENV_HUM	-0.927757901	1		
SOIL_MOIS	-0.094106142	0.149400682	1	
LUM	0.64681014	-0.636856516	-0.002217859	1
	TEMP	ENV_HUM	SOIL_MOIS	LUM

Tabla 2: Tabla de correlación entre variables ambientales

En la Tabla 2, se puede observar el coeficiente de correlación entre todas las variables objeto del estudio. Estos resultados muestran un estrecho margen de correlación indirectamente proporcional entre la temperatura y la humedad de ambiente que es superior al 90%; mientras que el resto de variables ambientales no tienen un valor de correlación fuerte, lo que implica que son variables independientes. Para este estudio, se decidió tener en consideración para la predicción de una variable dos escenarios: registros anteriores de la misma variable y de todas las variables.

Se realizaron gráficos de la evolución de las variables a lo largo del tiempo que naturalmente mostraron que el valor de las variables varía de forma más o menos cíclica día a día. Por otro

lado, podría tener sentido intentar entender cómo varían los valores de estas variables a lo largo del año para averiguar la necesidad de utilizar modelos predictivos para diferentes partes del año.

A pesar de que se han hecho gráficos de la evolución de las variables ambientales a lo largo del tiempo y de que se han calculado medidas como promedios y desvíos estándar de las mismas, el hecho de no haber sido posible recopilar datos para un año entero y de ser un hecho conocido que las condiciones ambientales en Ecuador se mantienen sensiblemente las mismas a lo largo del año, llevó a la opción de utilizar sólo un modelo predictivo.

6.2.4. Preparación de datos

La preparación de datos (siendo esta fase iterativa), consiste en fabricar los posibles escenarios de prueba, siendo, para este trabajo, cuatro escenarios y cada uno dividido en 8 conjuntos de datos (*datasets*), dando un total de combinación 64 escenarios posibles para cada configuración del script de la Figura 35 en función de los parámetros minutos (*m*) y horas (*h*). Los escenarios posibles se muestran en la Tabla 3:

CONDICIÓN	VALOR	DESCRIPCIÓN
NÚMERO DE HORAS	2 – 6 horas	Número de horas hacia atrás seleccionadas para la prueba que se tienen en cuenta para prever el valor de lo próximo intervalo de tiempo.
TIPO DE DATOS	Crudos y Normalizados	Los datos pueden ser crudos (leídos directamente del sensor) y normalizados (datos comprendidos entre 0 - 1)
INTERVALO DE MINUTOS	De 1 a 60 minutos	Número de minutos del cual se tomará los intervalos de tiempo.
COMPETENCIA DE DATOS	Considerar en función de la misma variable, Considerar en función de todas las variables	Para la predicción de una variable se puede considerar solamente sus valores anteriores o lo de todas las variables que componen el problema.
USO DE DESVIACIÓN ESTÁNDAR	True o False	La predicción de la variable puede considerar el uso o no de la desviación estándar resultante de la agrupación de los datos

Tabla 3: Factores de combinación de escenarios

El cuarto proceso (4) del script grupa los datos mostrados en la Tabla 3, por un número h de horas. Posteriormente, el quinto proceso (5) añade el cálculo de la desviación estándar para cada lectura registrada del resultado de los tres primeros procesos. Finalmente, en el sexto proceso (6), se convierte los datos procesados a datos normalizados. Los procesos (5) y (6) pueden realizarse en paralelo y no son dependientes el uno del otro.

Los resultados de los procesos (4) y (5), para la variable ambiental de **temperatura** con una configuración de $m=30$ (30 minutos), $h=3$ (3 horas atrás) y considerando la desviación estándar, son los correspondientes a la Tabla 4, cuando se considera solamente la misma variable ambiental, y los de la Tabla 5, cuando se consideran todas las variables ambientales para la predicción. En estas tablas, la penúltima columna más a derecha es la correspondiente al valor que se pretende prever e las restantes a la izquierda representan lo histórico reciente de las condiciones ambientales en el invernadero.

TEMP-120	ERR_TEMP_90	TEMP-902	ERR_TEMP_903	TEMP-60	ERR_TEMP_60	TEMP-30	ERR_TEMP_30	TEMP_PREVISTA	HOUR
27.8020492	0.514101406	27.23046472	1.009422314	27.8779461	0.306838088	28.105605	0.557758783	27.32905983	16.5
27.2304647	1.009422314	27.87794613	1.306838088	28.105605	0.557758783	27.3290598	0.757873422	25.35408526	17
27.8779461	1.306838088	28.10560498	0.557758783	27.3290598	0.757873422	25.3540853	1.087286573	19.66012891	17.5
28.105605	0.557758783	27.32905983	0.757873422	25.3540853	1.087286573	19.6601289	0.51893564	18.53485772	18
27.3290598	0.757873422	25.35408526	1.087286573	19.6601289	0.51893564	18.5348577	0.445103598	17.61710526	18.5
17.9492021	0.091397313	17.78647661	0.174398997	18.972029	0.481857404	21.111039	0.060303046	20.31196455	18.5
17.7864766	0.174398997	18.97202899	0.481857404	21.111039	0.060303046	20.3119645	0.295585257	18.37760646	19
...

Tabla 4: Preparación datos para la variable temperatura $m=30$, $h=3$, misma variable

ENV-120	...	ENV_30	ERR_ENV_30	SM-120	...	SM_30	ERR_SM_30	LUM-120	...	LUM_30	ERR_LUM_30	TEMP-120	...	ERR_TEMP_30	TEMP_PREVISTA	HOUR
48.5348361	...	45.2320285	11.709034	511.13388	...	475.638197	72.7285963	1004.13115	...	1003.94662	0	27.80205	...	0.557758783	27.32905983	16.5
46.6947504	...	45.7466667	14.7648731	500.148594	...	468.756695	71.2049223	1005.86265	...	995.109402	0	27.23046	...	0.757873422	25.35408526	17
46.7090909	...	49.4609236	16.7525277	486.823793	...	467.55595	70.8558637	1004.89697	...	985.53286	0	27.87795	...	1.087286573	19.66012891	17.5
45.2320285	...	68.2430939	20.9405181	475.638197	...	469.173726	71.497033	1003.94662	...	917.042357	0	28.1056	...	0.51893564	18.53485772	18
45.7466667	...	75.3661585	23.2774666	468.756695	...	470.012195	72.8139725	995.109402	...	926.014228	0	27.32906	...	0.445103598	17.61710526	18.5
73.9178191	...	60.621645	16.856691	534.734043	...	500.330928	85.0470022	965.085106	...	996.874459	0	17.9492	...	0.060303046	20.31196455	18.5
75.0225877	...	63.8688331	16.8907652	523.867934	...	494.908912	83.0461387	969.073099	...	970.923191	0	17.78648	...	0.295585257	18.37760646	19
70.4374638	...	77.5961821	22.5464268	509.601449	...	493.895742	82.5122272	988.243478	...	931.826725	0	18.97203	...	0.304751601	17.21728673	19.5
60.621645	...	82.8611028	22.2105296	500.330928	...	493.717834	81.8963506	996.874459	...	901.947839	0	21.11104	...	0.262398791	16.35529851	20
...

Tabla 5: Preparación datos para la variable temperatura $m=30$, $h=3$, todas las variables

Por otra parte, los resultados de los procesos (4) y (6), para la variable ambiental de **luminosidad** con una configuración de $m=15$ (15 minutos), $h=2$ (2 horas atrás) y sin desviación estándar, corresponden a la Tabla 6, cuando se considera solamente la misma variable ambiental, y los de la Tabla 7, cuando se consideran todas las variables ambientales para la predicción.

LUM - 120	...	LUM - 45	LUM - 30	LUM - 15	LUM PREVISTA	HOUR
1004.13115	...	1005.36331	1002.55986	997.56	992.5298246	16.75
1004.2974	...	1002.55986	997.56	992.529825	998.1577061	17
1006.83974	...	997.56	992.529825	998.157706	973.1302817	17.25
1005.52542	...	992.529825	998.157706	973.130282	918.5220588	17.5
1003.8796	...	998.157706	973.130282	918.522059	915.5571956	17.75
1005.36331	...	973.130282	918.522059	915.557196	919.7509728	18
1002.55986	...	918.522059	915.557196	919.750973	932.8638298	18.25
997.56	...	915.557196	919.750973	932.86383	914.4605263	18.5
965.085106	...	1000.29683	993.442197	986.976048	955.2915452	18.75
...

Tabla 6: Preparación datos para la variable luminosidad $m=15$, $h=2$, misma variable

TEMP-120	...	TEMP-15	EH - 120	...	EH-15	SM-120	...	SM-15	LUM - 120	...	LUM_PRED	HOUR
27.8020492	...	27.7947183	48.5348361	...	44.9179577	511.13388	...	472.776995	1004.13115	...	992.529825	16.75
26.5749071	...	29.1766667	47.5605948	...	43.7948333	504.519207	...	468.957778	1004.2974	...	998.157706	17
27.7956731	...	25.3842105	45.9482372	...	47.8012281	496.380342	...	468.545029	1006.83974	...	973.130282	17.25
29.1471186	...	27.1761649	46.0522034	...	43.9290323	489.950282	...	467.51374	1005.52542	...	918.522059	17.5
26.6257525	...	23.5640845	47.3571906	...	54.8954225	483.73913	...	467.597418	1003.8796	...	915.557196	17.75
28.4232014	...	19.9992647	45.5528777	...	65.4454044	478.561151	...	468.844363	1005.36331	...	919.750973	18
27.7947183	...	19.3197417	44.9179577	...	71.051107	472.776995	...	469.504305	1002.55986	...	932.86383	18.25
29.1766667	...	18.7844358	43.7948333	...	74.1978599	468.957778	...	469.564202	997.56	...	914.460526	18.5
17.9492021	...	21.3215318	73.9178191	...	59.5956647	534.734043	...	498.38632	965.085106	...	955.291545	18.75
...

Tabla 7: Preparación datos para la variable temperatura $m=30$, $h=2$, misma variable

Cabe destacar que la preparación de datos de un modelo predictivo para una variable ambiental en los próximos m minutos, se ha de considerar el valor de las últimas h horas de la misma o de todas las variables ambientales de las últimas h horas divididos en períodos de m minutos. En la Tabla 8, se muestra cual sería la disposición de la variable de temperatura, cuando el *script* es configurado para 3 horas con intervalos de 30 minutos. Las diferentes tramas de valores son agrupadas de tal forma en que se forma una diagonal, donde se busca predecir la temperatura a la hora 7:30, los datos considerados para la predicción corresponden a la fila de valor X.

	3 horas atrás		2 horas atrás		1 hora atrás		Valor previsto
	T-180	T-150	T-120	T-90	T-60	T-30	T
5:00	<i>f</i>	<i>g</i>	<i>h</i>	<i>i</i>	<i>j</i>	<i>k</i>	<i>l</i>
6:30	<i>g</i>	<i>h</i>	<i>i</i>	<i>j</i>	<i>k</i>	<i>l</i>	<i>m</i>
7:00	<i>h</i>	<i>i</i>	<i>j</i>	<i>k</i>	<i>l</i>	<i>m</i>	<i>n</i>
7:30	<i>i</i>	<i>j</i>	<i>k</i>	<i>l</i>	<i>m</i>	<i>n</i>	X

Tabla 8: Preparación de los datos en el *script*

6.3. Modelamiento

La implementación del modelo fue un proceso exploratorio y experimental durante el cual se correspondió cada uno de los algoritmos de aprendizaje con cada uno de los conjuntos de datos provistos por el procedimiento descrito en la Sección 6.2.

Las diferentes pruebas que se fueron realizando consistían mayormente en ir probando diferentes combinaciones de datos con la finalidad de encontrar el mejor escenario con menor porcentaje de error. Los escenarios probados fueron en función al número de datos disponibles. Cada una de estas combinaciones da un total de 64 conjuntos de pruebas y cada uno de estos conjuntos fue sometido a cada uno de los tres algoritmos de aprendizaje seleccionados.

En el algoritmo de regresión lineal no hay ninguna configuración de parámetros que pueda alterar los resultados de esta prueba. En la Figura 37, se muestran el modelo y los respectivos resultados, aplicado al escenario para este algoritmo, parametrizado para la variable de temperatura con: $m=30$, $h=3$, sin desviación estándar, datos no normalizados y considerando únicamente la misma variable de temperatura.

```
Linear Regression Model
12 =                               === Summary ===
-0.1009 * 0 +                       Correlation coefficient           0.9811
-0.0386 * 2 +                       Mean absolute error             0.8804
 0.0361 * 4 +                       Root mean squared error        1.3791
-0.295 * 8 +                        Relative absolute error        14.2758 %
 1.2911 * 10 +                      Root relative squared error    19.4368 %
 0.0216 * hour +                   Total Number of Instances      108
 1.3927
```

Figura 37: Modelo con algoritmo de regresión lineal para temperatura, $m=30$ y $h=3$

En algoritmos como el de SVM, existe un parámetro de configuración (complejidad de variable C) que influencia su funcionamiento. En la Figura 38, se muestra el modelo de SVM y sus resultados, el escenario para la variable de luminosidad con: $m=30$, $h=5$, sin desviación estándar, datos no normalizados y considerando únicamente la misma variable y una complejidad de $C=30$.

```

SM0reg
weights (not support vectors):
- 0.0022 * (normalized) 0
+ 0.0057 * (normalized) 2
- 0.0084 * (normalized) 4
+ 0.0157 * (normalized) 6
- 0.0711 * (normalized) 8
+ 0.2626 * (normalized) 10
- 0.9392 * (normalized) 12
+ 1.7356 * (normalized) 14
- 0.0018 * (normalized) hour
+ 0.0011
Correlation coefficient 0.967
Mean absolute error 44.1717
Root mean squared error 120.6528
Relative absolute error 9.619 %
Root relative squared error 25.8702 %
Total Number of Instances 102

```

Figura 38: Modelo con algoritmo de regresión lineal para temperatura, $m = 30$, $h = 5$ y $C = 23$

En el algoritmo de Redes Neuronales los atributos que pueden significar un mejor acercamiento al resultado ideal, corresponde al tamaño de las capas ocultas (número de neuronas) de la RNA. La Figura 39, muestra el modelo obtenido y sus resultados aplicando el algoritmo de Redes Neuronales para la variable Humedad de Ambiente con la configuración: $m=30$, $h=6$, con desviación estándar, datos normalizados y considerando todas las variables ambientales y el parámetro de *hidden layers*=10.

```

Linear Node 0
Inputs Weights
Threshold -0.14625797157867274
Node 1 -0.9523096208123439
Node 2 0.804051214311823234
Node 3 -0.004408783875593859
Node 4 -0.6948458240445345
Node 5 0.7657631167376621
Node 6 1.0488979375380577
Node 7 -0.7738676789574413
Node 8 0.6412589293865685
Node 9 -0.4077018978191156
Node 10 0.5844678673662218
==== Summary ====
Correlation coefficient 0.9545
Mean absolute error 0.0439
Root mean squared error 0.0669
Relative absolute error 25.5083 %
Root relative squared error 30.4376 %
Total Number of Instances 100

Sigmoid Node 1
Inputs Weights
Threshold -0.5231790172888832
Attrib temp - 0 -0.5848571756943555
Attrib temp - 1 -0.10932824298918026
Attrib temp - 2 -0.3571633967912368
Attrib temp - 3 0.056394947581139734
Attrib temp - 4 -0.41554283768559247
Attrib temp - 5 -0.7410587872982591
Attrib temp - 6 -0.1328908833002518
Attrib temp - 7 0.46864122746808533
Attrib temp - 8 0.21347042439347694
Attrib temp - 9 0.09984354489678842

```

Figura 39: Modelo con algoritmo de redes neuronales para humedad de ambiente, $m = 30$, $h = 6$ y $HL = 10$

El proceso descrito en esta sección es iterativo y consiste en configurar las pruebas sobre los escenarios disponibles en búsqueda de los mejores escenarios y resultados.

6.4. Evaluación

En esta sección se muestra una recopilación de los mejores resultados obtenidos durante la implementación del modelo para $m=30$. Al ser la temperatura el parámetro de mayor impacto, se analizan con más detalle los resultados de este. Para cada uno se comprueba si la presencia de

la desviación estándar mejora la precisión del modelo. Al final se presentan los mejores resultados para las otras variables ambientales.

6.4.1. Regresión Lineal

Para el algoritmo de regresión lineal se obtuvieron resultados progresivamente mejores para los diferentes escenarios de tiempo. En la Tabla 9, se resumen los resultados obtenidos de dos a seis horas usando desviación estándar. Como se puede comprobar, los valores tienden a mejorar cuando se considera mayor número de horas con un pico en las 5 horas.

	2H	3H	4H	5H	6H
CORRELATION COEFFICIENT	0.9804	0.9807	0.9843	0.9875	0.9762
MEAN ABSOLUTE ERROR	0.8488	0.7821	0.7828	0.746	0.9702
ROOT MEAN SQUARED ERROR	1.3948	1.3937	1.2288	1.1212	1.5857
RELATIVE ABSOLUTE ERROR	13.83%	12.68%	12.78%	12.10 %	15.10%
ROOT RELATIVE SQUARED ERROR	19.74%	19.64%	17.49%	15.81 %	21.61%
TOTAL NUMBER OF INSTANCES	111	108	105	102	100

Tabla 9: Resultados de Regresión lineal para 2h a 6h con error de desviación estándar

En la Tabla 10 se muestran los resultados de regresión lineal sin el uso del coeficiente de desviación estándar. De la misma forma, la tendencia mejora cuantas más horas se considere previamente, mostrando que en 5h hay también un pico de mejora.

	2H	3H	4H	5H	6H
CORRELATION COEFFICIENT	0.9794	0.9811	0.9854	0.9819	0.982
MEAN ABSOLUTE ERROR	0.9091	0.8804	0.7968	0.8218	0.8509
ROOT MEAN SQUARED ERROR	1.429	1.3791	1.192	1.3364	1.3859
RELATIVE ABSOLUTE ERROR	14.81%	14.28%	13.01%	13.33 %	13.24%
ROOT RELATIVE SQUARED ERROR	20.23%	19.44%	16.97%	18.84 %	18.88%
TOTAL NUMBER OF INSTANCES	111	108	105	102	100

Tabla 10: Resultados de Regresión lineal para 2h a 6h sin error de desviación estándar

Estos resultados muestran que el mejor escenario es considerar el modelo con desviación estándar para las últimas cinco horas, obteniendo un coeficiente de correlación de 0.9875, con un error absoluto-relativo de 12.10% que equivale a 0.746°C.

6.4.2. Redes Neuronales

Tal como con el algoritmo de regresión lineal, con las redes neuronales se obtuvieron mejores resultados para 5 horas. Igualmente, los resultados fueran mejores cuando se consideró apenas los valores anteriores de la temperatura.

	2H	3H	4H	5H	6H
CORRELATION COEFFICIENT	0.9826	0.9836	0.9874	0.987	0.9669
MEAN ABSOLUTE ERROR	0.8916	0.9587	1.31	0.8661	1.8747
ROOT MEAN SQUARED ERROR	1.3662	1.3679	1.5763	1.1863	2.3456
RELATIVE ABSOLUTE ERROR	14.52%	15.55%	21.38%	14.05 %	29.17%
ROOT RELATIVE SQUARED ERROR	19.34%	19.28%	22.44%	16.73 %	31.96%
TOTAL NUMBER OF INSTANCES	111	108	105	102	100

Tabla 11: Resultados de Redes Neuronales para 2h a 6h sin error de desviación estándar

En la Tabla 11, se presentan los resultados del algoritmo de redes neuronales en que lo número de nodos en la capa oculta es 20 (parámetro *hiddenLayers*). Contrario a lo que acontece con regresiones lineales, se obtienen mejores resultados cuando el error de desviación estándar es excluido del modelo de predicción.

Experimentalmente, se intentó diferentes configuraciones para el parámetro *hiddenLayers*, en búsqueda de los mejores resultados. El valor de 20 presentó el mejor porcentaje de aceptación. Nuevamente, los mejores resultados se obtuvieron al considerar valores previos de las últimas cinco horas, con un coeficiente de correlación de 0.987 y un porcentaje de error relativo absoluto de 14.05% que equivale a un error en la predicción promedio de 1.19°C.

6.4.3. Support Vector Machines

Para el algoritmo de *Support Vector Machines* se obtuvieron mejores resultados con 5 horas atrás y cuando se consideró apenas los valores anteriores de la temperatura. Similar a redes neuronales, se obtiene también mejores resultados sin considerar el error de desviación estándar. Estos resultados se muestran en la Tabla 12.

	2H	3H	4H	5H	6H
CORRELATION COEFFICIENT	0.9796	0.9806	0.9853	0.9871	0.9818
MEAN ABSOLUTE ERROR	0.8531	0.7902	0.7381	0.6982	0.8342
ROOT MEAN SQUARED ERROR	1.4276	1.3942	1.1991	1.1292	1.4011
RELATIVE ABSOLUTE ERROR	13.90 %	12.81 %	12.05%	11.31%	12.98%
ROOT RELATIVE SQUARED ERROR	20.20%	19.65%	17.07%	15.92%	19.09%
TOTAL, NUMBER OF INSTANCES	111	108	105	102	100

Tabla 12: Resultados de Support Vector Machines para 2h a 6h sin error de desviación estándar

Experimentalmente, se modificó progresivamente el parámetro ‘C’ del algoritmo en búsqueda de mejores resultados. El valor de complejidad que fue probado y con el cual se consigue un mejor resultado es 23. El error relativo es 11.31%, que equivale a un error absoluto de 0.6975°C en promedio, convirtiéndose en el mejor resultado para implementar el modelo de predicción.

6.4.4. Otras variables ambientales

En esta sección se recopilan los mejores resultados para las otras variables que concuerdan con el mismo escenario de $m=30$, $h=5$, sin desviación estándar y considerando la misma configuración aplicado al algoritmo de SVM.

En la Tabla 13, se muestra el resultado de la variable Luminosidad y con un parámetro de complejidad del algoritmo de $C=23$. Obteniendo como resultado una predicción con error relativo de 9.619% que equivale a 44.1717 LUX.

5H	
CORRELATION COEFFICIENT	0.967
MEAN ABSOLUTE ERROR	44.1717
ROOT MEAN SQUARED ERROR	120.6528
RELATIVE ABSOLUTE ERROR	9.619 %
ROOT RELATIVE SQUARED ERROR	25.8702 %
TOTAL NUMBER OF INSTANCES	102

Tabla 13: Resultados de Support Vector Machines para 5h luminosidad

En la Tabla 14, se muestra el resultado de la variable Humedad de ambiente y con un parámetro de complejidad del algoritmo de $C=25$. Obteniendo como resultado una predicción con error relativo de 11.886% que equivale a 1.92% (La humedad ambiental es medida en porcentaje).

5H	
CORRELATION COEFFICIENT	0.9855
MEAN ABSOLUTE ERROR	1.9203
ROOT MEAN SQUARED ERROR	3.4956
RELATIVE ABSOLUTE ERROR	11.8869 %
ROOT RELATIVE SQUARED ERROR	17.0471 %
TOTAL NUMBER OF INSTANCES	102

Tabla 14: Resultados de Support Vector Machines para 5h humedad de ambiente

En la Tabla 15, se muestra el resultado de la variable Humedad de suelo y con un parámetro de complejidad del algoritmo de $C=23$. Obteniendo como resultado una predicción con error relativo de 2.236% que equivale a 1.093% (la humedad de suelo es medida en porcentaje y el sensor mide de 0 - 1023).

5H	
CORRELATION COEFFICIENT	0.9914
MEAN ABSOLUTE ERROR	11.1843
ROOT MEAN SQUARED ERROR	15.5442
RELATIVE ABSOLUTE ERROR	2.2369 %
ROOT RELATIVE SQUARED ERROR	3.0409 %
TOTAL NUMBER OF INSTANCES	102

Tabla 15: Resultados de Support Vector Machines para 5h humedad de suelo

6.5. Despliegue

Los modelos resultantes descritos de la Sección 6.4, son exportados desde la herramienta en archivos *. *model* y son consumidos e interpretados por las librerías del *framework* de WEKA en Java. Cada uno de los modelos es expuesto como servicio web REST y consumidos en métodos POST.

El modelo de predicción de cada variable ambiental es un escenario que toma en consideración las medias horas contenidas en las últimas 5 horas y la hora que se desea predecir en formato militar. La Figura 40, muestra la estructura de los datos de una solicitud al *endpoint* de temperatura.

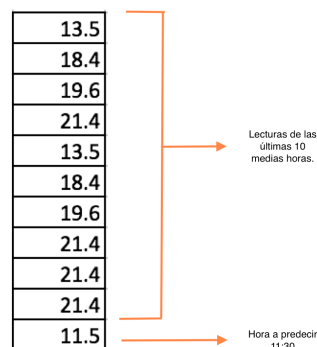


Figura 40: Datos de las últimas 5h para el servicio web

La Figura 41, muestra la petición POST al *endpoint* de temperatura publicado en la dirección URL: <https://predictionapi.herokuapp.com/>, y el valor resultante de la predicción.

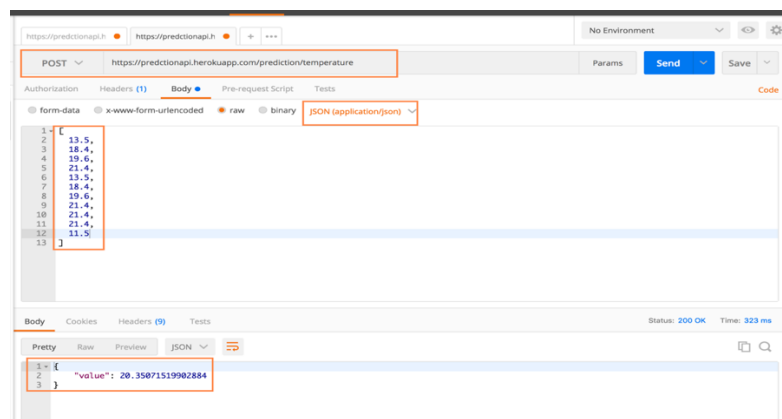


Figura 41: Consumo de servicio web expuesto por el API

6.6. Resumen del capítulo

En este capítulo se explica la implementación de las fases que componen el *Data Mining*, la definición del problema, la exploración y preparación de los datos recolectados desde el invernadero y cuáles son las consideraciones se necesita tener al fabricar los diferentes conjuntos de datos para ser analizados. Posteriormente, se describe la implementación del modelo y sus experiencias, se presentan los resultados obtenidos y su despliegue o puesta en producción. Además, se alienta la importancia que tienen los resultados de las diferentes implementaciones y su impacto, siendo el algoritmo de SVM el que mejor resultado proporciona con un error relativo de 11.33%, que equivale a un error absoluto en promedio de 0.69°C para la variable de temperatura. Para las otras variables ambientales, las configuraciones del algoritmo son semejantes: Para la luminosidad el mejor resultado es una predicción con error relativo de 9.619% que equivale a 44.1717 LUX. Los resultados para humedad de suelo tienen error relativo de 2.236% que equivale a 1.093%. Finalmente, para humedad de ambiente el error relativo es de 11.886% que equivale a 1.92%.

CAPÍTULO VII

7. Conclusiones y trabajo futuro

Este estudio consistió de tres fases muy variadas que enfocan diferentes áreas de la ciencia informática, siendo un proyecto ecléctico, aportando con un aprendizaje muy variado de doctrinas heterogéneas, pero complementarias como son: Hardware, Software y *Data Mining*, permitiendo abordar un problema desde diferentes aristas.

La implementación de la WSN, consistió en el ensamblaje de varios componentes de hardware como nodos controladores o enrutadores y nodos de datos basado en sensores de las variables ambientales; su implementación pretende brindar un repositorio de datos con el histórico de las variables del invernadero. Los nodos construidos funcionan correctamente y sus señales no son sobrepuestas en la transmisión a través de los protocolos de conexión inalámbrica.

La aplicación cliente constituye en dotar de una herramienta usable para los agricultores y ofrece como su principal ventaja no forzar al agricultor a estar presente en el invernadero para ver lo que sucede a tiempo real. El estado actual de la aplicación se encuentra en Beta y está integrada al API de predicción, con el modelo resultante de la predicción. La tecnología desarrollada en este proyecto es fundamentalmente una herramienta de administración, monitoreo y soporte para el sector agrícola y sus resultados de implementación pueden ser extendidos a otros dominios del área agrícola.

El algoritmo de predicción seleccionado, SVM, constituye el mejor resultado para todas las variables ambientales con variaciones mínimas en el parámetro de complejidad del algoritmo. Se espera que el sistema predictivo desarrollado inflencie directamente en la toma de decisiones respecto al adecuado manejo de parámetros agro-ambientales de temperatura, luminosidad, humedad de suelo y ambiental, que afectan directamente al normal crecimiento de los cultivos.

El sistema propuesto garantizará para la agricultura una herramienta de monitoreo para los parámetros antes nombrados. La solución resultante de la integración permitirá a los agricultores tomar acciones preventivas y correctivas sobre su cultivo, cuando sean necesarios, dotándolos de

una plataforma tecnológica basada en software libre y hardware de bajo costo y el uso de técnicas de minería de datos.

El aprendizaje y las experiencias obtenidas durante la implementación de este proyecto permitió que el trabajo pueda ser presentado a través de un artículo, mismo que fue aceptado en la conferencia Centeris a ser realizada en Barcelona (Centeris, 2017).

Existen algunas mejoras en relación a las oportunidades para el futuro del proyecto y el trabajo que podría tomar como punto de partida el estado actual de esta investigación. Entre ellas en cuanto a la red WSN:

La implementación de un banco de energía solar para dotar al nodo de autosuficiencia en cuestión eléctrica. Para la resolución de problemas como los descritos en la Sección 4.3.2, podrían adaptarse unos motores *servos* que brinden al nodo la rotación necesaria para que el banco de energía reciba perpendicularmente los rayos solares y tenga la capacidad de recargarse. Otro punto de mejora podría ser la instalación de cámaras de alta precisión que brinde la capacidad de procesamiento de imágenes para capturar las fases de los estados fenológicos del crecimiento de las plantas en el invernadero.

En cuanto a la aplicación cliente los cambios en la WSN podrían ser nuevas funcionalidades para el monitoreo del estado del invernadero y el monitorio de las plantas. Así como también, se podrían actualizar a mayores versiones muchos de los *frameworks* y librerías con la finalidad de simplificar ciertos funcionamientos que exigían demasiada configuración o desarrollo.

Mientras que en el algoritmo predictivo una sustancial mejora para el futuro podría consistir en identificar nuevos escenarios con una WSN sólida y de funcionamiento ininterrumpido podría ayudar a considerar otras variables no climatológicas que incidan en el crecimiento de los cultivos y haciendo los modelos más confiables. Además de si se cuenta con imágenes del crecimiento de las plantas podría ser útil para predecir nueva información como cuando las estarán listas para ser cortadas.

8. Bibliografía

- A. K. Tripathy, J. A. (2013). Data mining and wireless sensor network for groundnut pest/disease precision protection. *2013 National Conference* (págs. 1–8). --: Parallel Computing Technologies.
- Abrahamsson, P. (2010). *Mobile-D: An Agile Approach for Mobile Application Development*. University of Oulu, Department of Information Processing Science. Oulu: VTT Technical Research Centre of Finland.
- airbnb. (17 de 08 de 2017). *Airbnb JavaScript Style Guide()* {. Recuperado el 18 de 08 de 2017, de Github: <https://github.com/airbnb/javascript>
- Andre Charland, B. L. (2011). Mobile Application Development: Web vs. Native. *acmqueue*, 49.
- Apache. (05 de 2017). *Github*. Recuperado el 04 de 07 de 2017, de Github Apache Cordoba: <https://github.com/apache/cordova-js>
- Asencios, V. V. (12 de 2004). DATA MINING Y EL DESCUBRIMIENTO DEL CONOCIMIENTO. *Revista de la Facultad de Ingeniería Industrial*.
- Aurelio Barrera, J. C. (2007). *Producción de Rosa Coster*. Secretaría de Agricultura, Ganadería, Desarrollo Rural, Pesca y Alimentación, Zacatepec, Morelos, México. Morelos: Secretaría de Agricultura, Ganadería, Desarrollo Rural, Pesca y Alimentación.
- Banco Mundial. (28 de 09 de 2016). *Banco Mundial BIRF - AIF*. Recuperado el 28 de 12 de 2016, de Ecuador: panorama general: <http://www.bancomundial.org/es/country/ecuador/overview>
- Bhagawati, K., Sen, A., Shukla, K. K., & Bhagawati, R. (07 de 2016). Application and Scope of Data Mining in Agriculture. *International Journal of Advanced Engineering Research and Science (IJAERS)*, 3(7).

- Centeris. (2017). *Centeris Conference*. Recuperado el 08 de 09 de 2017, de Centeris Conference: <http://centeris.scika.org>
- Cisneros-Mata, M. A. (1996). Artificial neural networks to forecast biomass of pacific sardine and its environment. *Ciencias Marinas*, 22(4), 427-442.
- Codebeat. (27 de 08 de 2017). *Codebeat*. Recuperado el 14 de 09 de 2017, de Codebeat: <https://codebeat.co/a/schubertd/projects>
- Computer Hope. (27 de 06 de 2017). *Epoch*. Recuperado el 15 de 08 de 2017, de Computer Hope Epoch Time: <https://www.computerhope.com/jargon/e/epoch.htm>
- El Telégrafo. (16 de 07 de 2016). *El PIB ecuatoriano registró una reducción de -1,9% en el primer trimestre de 2016* . Recuperado el 28 de 12 de 2016, de El Telégrafo: <http://www.eltelegrafo.com.ec/noticias/economia/8/el-pib-ecuatoriano-registro-una-reduccion-de-1-9-en-el-primer-trimestre-de-2016>
- Evaluando Software. (15 de 01 de 2013). *Evaluando Software*. Recuperado el 08 de 09 de 2016, de Evaluando Software Que es el modelo predictivo: <http://www.evaluandosoftware.com/que-es-el-analisis-predictivo-2/>
- Expoflores. (01 de 06 de 2016). *Expoflores*. Recuperado el 28 de 12 de 2016, de Expoflores Balance Mayo 2016: <https://sway.com/5kAk1IIHwAEOXwwE>
- Fernando Izaurieta, C. S. (2000). *Redes Neuronales Artificiales*. Universidad de Concepción, Departamento de Física. Concepción: Universidad de Concepción.
- Fundación Big Data. (s.f.). *Fundación Big Data*. Recuperado el 08 de 09 de 2016, de Análisis predictivo: <http://fundacionbigdata.org/analisis-predictivo/>
- G. E. Meyer, J. N. (2004). Intensified fuzzy clusters for classifying plant, soil, and residue regions of interest from color images. *Computer Electronics Agric*, 42, 161-180.
- Gao, B., Malik, S., Santoso, Y., & Zhu, Z. (12 de 05 de 2016). Using Data Mining Technique to Predict Seasonal Climate Change.

- González, A. P. (2010). *ANÁLISIS PREDICTIVO DE DATOS MEDIANTE TÉCNICAS DE REGRESIÓN ESTADÍSTICA*. Universidad Complutense de Madrid, Facultad de Informática, Madrid.
- Google. (2016). *Android Developers Getting Started*. Recuperado el 20 de 09 de 2016, de Android Developers: <https://developer.android.com/training/basics/supporting-devices/index.html>
- Google Inc. (17 de 05 de 2017). *Google I/O*. Recuperado el 17 de 05 de 2017, de <https://events.google.com/io/>
- Info Agro. (2016). *EL CULTIVO DE LAS ROSAS PARA CORTE (1ª parte)*. Recuperado el 05 de 09 de 2016, de infoagro.com: <http://www.infoagro.com/flores/flores/rosas.htm>
- J. Su, C.-C. T.-T. (2012). Area temperature system monitoring and computing based on adaptive fuzzy logic in wireless sensor networks. *Applied Soft Computing*, 12, 1532–1541.
- Jetbrains. (22 de 11 de 2016). *IntelliJ IDEA*. Recuperado el 4 de 12 de 2016, de Jetbrains s.r.o.: <https://www.jetbrains.com/idea/whatsnew/>
- Jetbrains. (2016). *WebStorm*. Recuperado el 26 de 12 de 2016, de Jetbrains s.r.o.: <https://www.jetbrains.com/webstorm/whatsnew/>
- Jetbrains. (2017). *Try Kotlin*. Recuperado el 30 de 05 de 2017, de Kotlin Language: <https://kotlinlang.org>
- JetBrains s.r.o. (2016). *PyCharm*. Recuperado el 10 de 06 de 2017, de JetBrains: <https://www.jetbrains.com/pycharm/whatsnew/>
- JQuery Foundation. (2017). *JQuery*. Recuperado el 20 de 08 de 2017, de JQuery: <https://jquery.com>
- Juan Jesús Romero, C. D. (2007). *Inteligencia Artificial y Computación Avanzada (Vol. 13)*. (F. A. Brañas, Ed.) Santiago de Compostela: Fundación Alfredo Brañas.
- L. Zhao, L. H. (2013). Intelligent agricultural forecasting system based on wireless sensor. *Journal of networks*, 8, 1817–1824.

- Machine Learning Group at the University of Waikato. (01 de 04 de 2017). *Weka 3: Data Mining Software in Java*. Recuperado el 06 de 08 de 2017, de Weka 3 - Data Mining with Open Source Machine Learning Software: <http://www.cs.waikato.ac.nz/ml/weka/>
- Meteor Developers. (2016). *Introducing Meteor API Docs*. Recuperado el 20 de 09 de 2016, de Meteor Developers: <http://docs.meteor.com/#/full/>
- Molina, L. C. (2000). *Torturando los Datos Hasta que Confiesen*. Universidad de Cataluña, Departamento de Lenguajes y Sistemas Informáticos, Barcelona.
- Node.js Foundation. (15 de 08 de 2017). *NodeJs*. Recuperado el 17 de 08 de 2017, de <https://nodejs.org/en/>
- Npm Inc. (2017). *npm*. Recuperado el 15 de 08 de 2017, de npm: <https://docs.npmjs.com/getting-started/what-is-npm>
- Pérez López, C. (2017). *Minería de datos: técnicas y herramientas*. Madrid, España: Paraninfo.
- R. Jain, S. M. (2009). Machine learning for forewarning crop diseases. *J. Ind. Soc. Agri. Stat.*, 63(1), 97-107.
- REDCEDIA. (2016). *Sistema predictivo para la producción Just in Time (JIT) en el cultivo de rosas ecuatorianas*. Fundación consorcio ecuatoriano para el desarrollo del internet avanzado, REDCEDIA, Sangolquí.
- Rossi, G., Pastor, Ó., Schwabe, D., & Olsina, L. (s.f.). *Web Engineering: Modelling and Implementing Web Applications*. Valencia, España.
- S. W. Lee, L. K. (1998). Methodology and life cycle model for data mining and knowledge discovery in precision agriculture. *IEEE International Conference on Systems, Man and Cybernetics*. 3, págs. 2882–2887. IEEE.
- SAS Institute. (2017). *Data Mining What it is and why it matters*. Recuperado el 28 de 08 de 2017, de SAS The power to know: https://www.sas.com/en_us/insights/analytics/data-mining.html

- SV Europe. (2015). *SV Europe*. Recuperado el 14 de 11 de 2016, de SV Europe Crisp: <http://www.sv-europe.com/crisp-dm-methodology/>
- Traeg, P. (22 de 11 de 2013). *Four Ways To Build A Mobile Application*. Recuperado el 20 de 09 de 2016, de Smashing Magazine: <https://www.smashingmagazine.com/2013/11/four-ways-to-build-a-mobile-app-part1-native-ios/>
- Universidad de California. (21 de 06 de 2016). *Degree-Days: Single Sine Method*. Recuperado el 04 de 12 de 2016, de University of California Agriculture & Natural resources: http://ipm.ucanr.edu/WEATHER/ddss_tbl.html
- Universidad Nacional Abierta y a Distancia. (2008). *Leccion 10. Cultivo de Rosa*. Recuperado el 05 de 09 de 2016, de Datateca UNAD: http://datateca.unad.edu.co/contenidos/302568/Material_didactico_definitivo/leccion_10_cultivo_de_rosa_rosa_spp.html
- Universidad Nacional Abierta y a Distancia. (2008). *Lección 40. Medición de la humedad del suelo*. Recuperado el 05 de 09 de 2016, de Datateca: http://datateca.unad.edu.co/contenidos/30172/MODULO%20HIDROLOGIA/leccin_40_medicin_de_la_humedad_del_suelo.html
- Witten, I. H., Frank, E., & Hall, M. A. (2011). *Data Mining: Practical Machine Learning Tools and Techniques* (3ra Edición ed.). Burlington, USA.
- Wodehouse, C. (2017). *UpWork*. Recuperado el 15 de 08 de 2017, de Is MeteorJS the Right Node.js Framework for Your App?: <https://www.upwork.com/hiring/development/meteorjs-node-js-framework/>

