



**ESPE**  
UNIVERSIDAD DE LAS FUERZAS ARMADAS  
INNOVACIÓN PARA LA EXCELENCIA

**Aplicación móvil de monitoreo, análisis y visualización de datos, para la toma de decisiones en los procesos de producción de Cultivo Florícola**

Cabezas Mendieta, Ayrton Wladimir y Jiménez Collaguazo, Alexander Dario

Departamento de Ciencias de la Computación

Carrera de Ingeniería de Sistemas e informática

Trabajo de titulación, previo a la obtención del título de Ingeniero en Sistemas e Informática

Ing. Díaz Zúñiga, Magi Paúl MBA. MSc.

15 de agosto de 2020

## Document Information

<b>Analyzed document</b>	Tesis-CabezasA-JiménezD.pdf (D78618481)
<b>Submitted</b>	9/4/2020 4:20:00 PM
<b>Submitted by</b>	DIAZ ZUÑIGA PAUL
<b>Submitter email</b>	mpdiaz@espe.edu.ec
<b>Similarity</b>	6%
<b>Analysis address</b>	mpdiaz.espe@analysis.orkund.com



## Sources included in the report

<b>W</b>	URL: <a href="https://fp.uoc.fje.edu/blog/desarrollo-de-aplicaciones-hibridas-con-entorno-ionic/">https://fp.uoc.fje.edu/blog/desarrollo-de-aplicaciones-hibridas-con-entorno-ionic/</a> Fetched: 9/4/2020 4:20:00 PM	 3
<b>SA</b>	<b>AndradeYandunTesis-18-07-2019.docx</b> Document AndradeYandunTesis-18-07-2019.docx (D54477804)	 3
<b>W</b>	URL: <a href="https://www.ticportal.es/glosario-tic/mysql">https://www.ticportal.es/glosario-tic/mysql</a> Fetched: 9/4/2020 4:20:00 PM	 1
<b>W</b>	URL: <a href="https://mape309site.wordpress.com/2017/11/15/ventajas-y-desventajas-de-mysql-oracle-...">https://mape309site.wordpress.com/2017/11/15/ventajas-y-desventajas-de-mysql-oracle-...</a> Fetched: 9/4/2020 4:20:00 PM	 4
<b>W</b>	URL: <a href="https://damiandeluca.com.ar/visual-studio-code-caracteristicas-principales">https://damiandeluca.com.ar/visual-studio-code-caracteristicas-principales</a> Fetched: 9/4/2020 4:20:00 PM	 1
<b>W</b>	URL: <a href="https://tech.tribalyte.eu/blog-que-es-una-api-rest">https://tech.tribalyte.eu/blog-que-es-una-api-rest</a> Fetched: 9/4/2020 4:20:00 PM	 1
<b>W</b>	URL: <a href="https://www.chakray.com/es/cuales-son-las-ventajas-de-una-api-rest/">https://www.chakray.com/es/cuales-son-las-ventajas-de-una-api-rest/</a> Fetched: 9/4/2020 4:20:00 PM	 2
<b>W</b>	URL: <a href="https://www.qualitydevs.com/2018/11/26/aplicaciones-moviles-multiplataforma/">https://www.qualitydevs.com/2018/11/26/aplicaciones-moviles-multiplataforma/</a> Fetched: 9/4/2020 4:20:00 PM	 1
<b>W</b>	URL: <a href="http://sedici.unlp.edu.ar/handle/10915/60497">http://sedici.unlp.edu.ar/handle/10915/60497</a> Fetched: 9/4/2020 4:20:00 PM	 1
<b>W</b>	URL: <a href="https://mobivery.com/desarrollo-de-aplicaciones-multiplataforma-nativo-o-hibrido/">https://mobivery.com/desarrollo-de-aplicaciones-multiplataforma-nativo-o-hibrido/</a> Fetched: 9/4/2020 4:20:00 PM	 1
<b>W</b>	URL: <a href="https://www.revistasocitec.org/index.php/TCE/article/view/27">https://www.revistasocitec.org/index.php/TCE/article/view/27</a> Fetched: 9/4/2020 4:20:00 PM	 1
<b>SA</b>	<b>GMCA-DISEÑO DE LA PLATAFORMA DE LIBROS DIGITALES DE LA UCE.docx</b> Document GMCA-DISEÑO DE LA PLATAFORMA DE LIBROS DIGITALES DE LA UCE.docx (D63603610)	 2



**DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN**  
**CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA**

**Certificación**

Certifico que el trabajo de titulación: **“Aplicación móvil de monitoreo, análisis y visualización de datos, para la toma de decisiones en los procesos de producción de cultivo florícola”** fue realizado por el señor **Cabezas Mendieta, Ayrton Wladimir** y el señor **Jiménez Collaguazo, Alexander Dario**; el cual ha sido revisado y analizado en su totalidad por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

**Sangolquí, 22 de febrero de 2021**

 Firmado electrónicamente por:  
**MAGI PAUL**  
.....**DIAZ....**

**Ing. Magí Paúl Díaz Zúñiga MBA. MSc.**



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN  
CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

#### RESPONSABILIDAD DE AUDITORÍA

Nosotros, **Cabezas Mendieta Ayrton Wladimir** y **Jiménez Collaguazo Alexander Dario** con cédulas de ciudadanía N° 1723451629 y N° 1719627232 declaramos que el contenido, ideas y criterios del trabajo de titulación: **"Aplicación móvil de monitoreo, análisis y visualización de datos, para la toma de decisiones en los procesos de producción de cultivo florícola"** es de nuestra autoría y responsabilidad, cumpliendo con los requerimientos teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciado las citas bibliográficas.

Sangolquí, 22 de febrero de 2021

Cabezas Mendieta  
Ayrton Wladimir  
C.I. 1723451629

Jiménez Collaguazo  
Alexander Dario  
C.I. 1719627232



### AUTORIZACIÓN DE PUBLICACIÓN

Nosotros, **Cabezas Mendieta Ayrton Wladimir** y **Jiménez Collaguazo Alexander Dario** con cédulas de ciudadanía N° 1723451629 y N° 1719627232 autorizamos a la Universidad de las Fuerzas Armadas ESPE, publicar el trabajo de titulación: **“Aplicación móvil de monitoreo, análisis y visualización de datos, para la toma de decisiones en los procesos de producción de cultivo florícola”** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de nuestra responsabilidad.

Sangolquí, 22 de febrero de 2021

Cabezas Mendieta  
Ayrton Wladimir  
C.I. 1723451629

Jiménez Collaguazo  
Alexander Dario  
C.I. 1719627232

### **Dedicatoria**

*El presente trabajo lo dedico a mi padre Pablo, mi madre Lilián y mi hermana Laura, pilar fundamental en mi vida, por el apoyo, cariño, regaños y cuidados diarios que me permitieron terminar y cumplir una de mis metas propuestas.*

*A cada miembro del resto de mi familia, que se preocupa por mí, por estar pendiente de mi progreso, por brindarme su ayuda y compartirme sus consejos.*

*A mis compañeros de estudio, amigos que siempre han estado en los momentos difíciles y también en los momentos de disfrute.*

*Ayrton Cabezas*

## Dedicatoria

*A mi abuela por ser parte fundamental en mi formación como persona enseñándome que todo por lo que se lucha tiene su recompensa y guiándome a lo largo de mi vida con sus valiosos consejos y amor incondicional.*

*A mi madre por acompañarme en todas las decisiones que he tomado siendo ese apoyo necesario el cual en más de una ocasión me ayudo a continuar luchando por alcanzar mis metas y seguir adelante.*

*A mi padre por el aliento y motivación además de siempre ayudarme en momentos difíciles velando por mi bienestar.*

*A mi hermano para transmitirle el ímpetu de luchar por sus objetivos y que continúe con su formación como lo ha hecho hasta ahora.*

*A mi familia, mujer e hija motivo de que siempre continúe adelante dando lo mejor de mí.*

*A mis amigos con los cuales compartimos aulas de clases, horas de estudio, deportes, varias actividades forjando amistades sinceras y duraderas.*

*Dario Jiménez*

## **Agradecimiento**

*Agradezco a Dios por brindarme salud, vida y permitirme llegar y disfrutar, junto a mi familia y amigos, del cumplimiento de uno de los objetivos de mi vida, el presente trabajo de titulación.*

*A mi padre Pablo, madre Lilián y hermana Laura por el apoyo, paciencia, regaños, exigencias y sobre todo el amor que me permitieron seguir adelante.*

*A resto de mi familia por su preocupación, consejos y ayuda brindada cuando más lo necesitaba.*

*A mis docentes, por brindarme el conocimiento necesario, por sus consejos de vida y experiencia que son necesarios tanto en la vida social como profesional.*

*A mis amigos y amigas de estudios, por todo el apoyo brindado en lo social y en lo académico.*

*Ayrton Cabezas*

## Agradecimiento

*Agradezco a Dios por permitirme haber llegado a este punto, y por saber tomar las decisiones adecuadas a lo largo de mi vida, así como de mi formación académica.*

*A mi abuela Clara que dedico gran tiempo en fomentar buenas costumbres y valores en mí, los mismos que delinearon mi carácter y fueron mi guía para mi crecimiento personal, académico y profesional, por su cariño infinito el cual siempre me mantuvo centrado y convencido de que lograre mis metas.*

*A mis padres, Dario y Cecilia pilar fundamental en mi vida los que me cuidaron y me educaron como una persona de bien trasmitiéndome conocimientos y valores en todo este trayecto.*

*A mi familia que en todo momento compartió tiempo conmigo haciéndome saber que todo se puede lograr siempre y cuando exista dedicación de por medio y motivándome para que continúe adelante en los momentos difíciles.*

*A los docentes que compartieron su tiempo conmigo no solo dentro del aula sino también como amigos dándose el tiempo de conocerme más como persona, convirtiéndose en motivación y ejemplo para alcanzar mis objetivos.*

*A todo mi grupo de amigos el cual pude conocer en este trascurso, con los cuales compartimos varias historias juntos, siempre buscamos darnos la mano entre nosotros, sabiendo que el logro de un amigo es un logro común y que la amistad que formamos nos acompañara a lo largo de nuestra vida.*

*De manera especial a mi tutor de tesis el Ing. Paul Diaz que nos brindó el apoyo para el desarrollo y finalización de este trabajo.*

*Por último, a todas aquellas personas que de una u otra manera ayudaron a que pueda alcanzar esta meta.*

*Dario Jiménez*

## Índice

Urkund .....	2
Certificación .....	3
Responsabilidad De Autoría.....	4
Autorización.....	5
Dedicatoria.....	6
Agradecimiento.....	8
Índice.....	11
Índice de tablas .....	15
Índice de imágenes .....	16
Resumen .....	19
Abstract.....	20
<b>Capítulo I.....</b>	<b>21</b>
<b>Introducción .....</b>	<b>21</b>
<b>1.1. Antecedentes.....</b>	<b>21</b>
<b>1.2. Planteamiento del Problema .....</b>	<b>22</b>
<b>1.3. Justificación .....</b>	<b>23</b>
<b>1.4. Objetivos.....</b>	<b>24</b>
<b>1.4.1. Objetivo General .....</b>	<b>24</b>
<b>1.4.2. Objetivos Específicos.....</b>	<b>24</b>
<b>1.5. Alcance.....</b>	<b>25</b>
<b>1.6. Hipótesis .....</b>	<b>25</b>
<b>Capítulo II.....</b>	<b>26</b>
<b>Marco Teórico .....</b>	<b>26</b>
<b>2.1. Tecnologías de la Información y Comunicación .....</b>	<b>26</b>
<b>2.1.1. Conceptos .....</b>	<b>26</b>
<b>2.2. Ingeniería de Software.....</b>	<b>27</b>
<b>2.2.1. Conceptos .....</b>	<b>27</b>
<b>2.3. Arquitectura Cliente Servidor .....</b>	<b>30</b>
<b>2.3.1. Conceptos .....</b>	<b>30</b>
<b>2.4. Herramientas tecnológicas utilizadas.....</b>	<b>31</b>

2.4.1.	Modelamiento UML .....	31
2.4.1.1.	Power Designer .....	31
2.4.2.	Entornos de desarrollo integrado (IDE) .....	32
2.4.2.1.	Spring boot .....	32
2.4.2.2.	Angular 9.....	34
2.4.2.3.	Ionic 5.....	34
2.4.2.4.	Cordova 9.....	36
2.4.3.	Servidor Web.....	36
2.4.3.1.	TomCat.....	36
2.4.4.	Base de Datos .....	37
2.4.4.1.	MySQL.....	37
2.4.5.	Editor de Código .....	38
2.4.5.1.	Visual Code .....	38
2.5.	Api Rest.....	38
2.5.1.	Conceptos .....	38
2.5.2.	Características .....	38
2.6.	Aplicaciones Multiplataforma .....	39
2.6.1.	Conceptos .....	39
2.6.2.	Características .....	39
2.6.3.	Ventajas .....	39
2.7.	Metodología .....	40
2.7.1.	Programación Extrema (XP).....	40
2.7.1.1.	Conceptos .....	40
2.7.1.2.	Ventajas .....	41
2.7.2.	Mobile-D.....	42
2.7.2.1.	Conceptos .....	42
2.7.2.2.	Ventajas .....	42
Capítulo III	.....	43
Desarrollo	.....	43
3.1.1.	Problema .....	43
3.1.2.	Análisis de viabilidad.....	44
3.2.	Especificaciones de Requerimientos .....	44
3.2.1.	Alcance del Sistema.....	44

3.2.2.	Perspectiva del Sistema .....	46
3.3.	Diseño del Sistema .....	47
3.3.1.	Arquitectura .....	47
3.3.2.	Diagramas y Casos de uso .....	49
3.3.3.	Diseño de la base de datos .....	76
3.3.4.	Diseño del Api Rest.....	79
3.4.	Desarrollo del Sistema .....	80
3.4.1.	Desarrollo del Backend .....	80
3.4.1.1.	Resources.....	81
3.4.1.2.	Entity.....	82
3.4.1.3.	Dal .....	84
3.4.1.4.	Service.....	87
3.4.1.5.	Util.....	91
3.4.1.6.	Controller.....	93
3.4.2.	Servicios Rest.....	97
3.4.2.1.	Exposición de Servicios .....	97
3.4.2.2.	CORS.....	98
3.4.2.3.	Tabla de Servicios .....	100
3.4.3.	Desarrollo del FrontEnd .....	104
3.4.3.1.	Web.....	104
3.4.3.2.	App.....	119
3.5.	Validación y pruebas .....	130
Capítulo IV	.....	131
Pruebas y Resultados	.....	131
4.1.	Pruebas .....	131
4.1.1.	Backend .....	131
4.1.1.1.	Servicios.....	131
4.1.2.	Frontend .....	140
4.1.2.1.	Web.....	140
4.1.2.2.	App.....	147
4.2.	Resultados .....	152
4.2.1.	Análisis de resultados sobre la satisfacción del usuario .....	152
Capítulo V	.....	157

<b>Conclusiones y Recomendaciones</b> .....	157
<b>Anexos</b> .....	159
<b>Bibliografía</b> .....	162

## Índice de tablas

<b>Tabla 1</b> Búsqueda en base de datos establecidas .....	44
<b>Tabla 2</b> Caso de uso iniciar sesión .....	50
<b>Tabla 3</b> Caso de uso cerrar sesión .....	53
<b>Tabla 4</b> Caso de uso seleccionar bloque .....	55
<b>Tabla 5</b> Caso de uso Visualizar Estadístico.....	56
<b>Tabla 6</b> Caso de uso Visualizar Resumen.....	58
<b>Tabla 7</b> Caso de uso navegar menú .....	60
<b>Tabla 8</b> Caso de uso salir del sistema .....	61
<b>Tabla 9</b> Caso de uso Consultar Condiciones del Bloque .....	64
<b>Tabla 10</b> Caso de uso Consultar Humedad .....	65
<b>Tabla 11</b> Caso de uso Consultar Temperatura.....	67
<b>Tabla 12</b> Caso de uso Consultar Grados Día Desarrollo .....	70
<b>Tabla 13</b> Caso de uso Consultar Estadístico .....	72
<b>Tabla 14</b> Caso de uso Consultar Grados Día Desarrollo por Rango .....	74
<b>Tabla 15</b> Tabla de Rutas.....	100

## Índice de imágenes

<b>Figura 1</b>	Ciclo de vida del software .....	28
<b>Figura 2</b>	<i>Arquitectura cliente-servidor</i> .....	30
<b>Figura 3</b>	<i>Fases de la metodología programación extrema</i> .....	41
<b>Figura 4</b>	<i>Fases de la metodología Mobile-D</i> .....	42
<b>Figura 5</b>	<i>Arquitectura de la aplicación.</i> .....	48
<b>Figura 6</b>	Diagrama nivel 0 de casos de uso para el ingreso a la aplicación.....	49
<b>Figura 7</b>	<i>Diagrama nivel 1 de casos de uso de la aplicación Green House</i> .....	49
<b>Figura 8</b>	<i>Diagrama nivel 2 de casos de uso de manejar sesión</i> .....	50
<b>Figura 9</b>	<i>Diagrama nivel 2 de casos de uso de manejar componentes</i> .....	54
<b>Figura 10</b>	<i>Diagrama casos de uso nivel 2 consultar datos.</i> .....	63
<b>Figura 11</b>	Modelo conceptual. ....	78
<b>Figura 12</b>	Modelo lógico. ....	78
<b>Figura 13</b>	Modelo físico.....	79
<b>Figura 14</b>	Estructura BackEnd. ....	81
<b>Figura 15</b>	Conexión a la base de datos desde BackEnd. ....	82
<b>Figura 16</b>	Dependencias JPA desde BackEnd.....	82
<b>Figura 17</b>	Mapeo de clases con la base de datos en el BackEnd. ....	83
<b>Figura 18</b>	Constructores en el BackEnd. ....	84
<b>Figura 19</b>	Estructura en la capa Dal en el BackEnd.....	84
<b>Figura 20</b>	Estructura de la sección usuario de la capa “dal” .....	85
<b>Figura 21</b>	Interfaz de la sección usuario de la capa “dal” .....	85
<b>Figura 22</b>	Clase de la sección usuario de la capa “dal” .....	86
<b>Figura 23</b>	Estructura de la capa service en el BackEnd.....	87
<b>Figura 24</b>	Interfaz de la capa servicio .....	88
<b>Figura 25</b>	Clase de la capa servicio .....	89
<b>Figura 26</b>	Método de acceso a Usuario .....	90
<b>Figura 27</b>	Método de acceso General.....	90
<b>Figura 28</b>	Método de acceso al Dashboard .....	90
<b>Figura 29</b>	Capa util en el BackEnd.....	91
<b>Figura 30</b>	Clase “ApiResponse.java” dentro de la capa util .....	92
<b>Figura 31</b>	Dependencias en el Controlador de la capa util .....	93
<b>Figura 32</b>	Etiquetas dentro del controlador de la capa util .....	93
<b>Figura 33</b>	Implementación del servicio dentro del controlador de la capa útil.....	94
<b>Figura 34</b>	Etiquetado del servicio en el controlador de la capa util .....	95
<b>Figura 35</b>	Desarrollo del servicio login dentro de la capa util.....	96
<b>Figura 36</b>	Diagrama CORS. ....	99
<b>Figura 37</b>	<i>Notificación correcta</i> .....	105
<b>Figura 38</b>	<i>Notificación incorrecta</i> .....	105
<b>Figura 39</b>	<i>Interfaz Login</i> .....	106
<b>Figura 40</b>	<i>Parámetros obligatorios en el Login</i> .....	107
<b>Figura 41</b>	<i>Session Store</i> .....	108
<b>Figura 42</b>	<i>Barra de navegación</i> .....	108

<b>Figura 43</b>	<i>Menú de navegabilidad</i>	109
<b>Figura 44</b>	<i>Condiciones actuales del bloque dentro de la pestaña Dashboard</i>	110
<b>Figura 45</b>	<i>Información del bloque actual dentro de la pestaña Dashboard</i>	111
<b>Figura 46</b>	<i>Selección del bloque</i>	111
<b>Figura 47</b>	<i>Cambio de valores en Session Storage</i>	112
<b>Figura 48</b>	<i>Gráfico estadístico con registros del último mes</i>	113
<b>Figura 49</b>	<i>Resumen mensual</i>	114
<b>Figura 50</b>	<i>Pantalla completa del dashboard</i>	115
<b>Figura 51</b>	<i>Sección Humedad</i>	116
<b>Figura 52</b>	<i>Sección Grados Días Desarrollo</i>	117
<b>Figura 53</b>	<i>Sección Grados Días Desarrollo por Rango</i>	118
<b>Figura 54</b>	<i>Gráfico Estadístico de Grados Días Desarrollo</i>	119
<b>Figura 55</b>	<i>Notificaciones en la aplicación móvil</i>	119
<b>Figura 56</b>	<i>Interfaz de login en aplicación móvil</i>	120
<b>Figura 57</b>	<i>Session Store</i>	121
<b>Figura 58</b>	<i>Barra de navegación</i>	122
<b>Figura 59</b>	<i>Menú de navegabilidad</i>	122
<b>Figura 60</b>	<i>Condiciones actuales de temperatura del bloque actual</i>	123
<b>Figura 61</b>	<i>Condiciones actuales de humedad del bloque actual</i>	123
<b>Figura 62</b>	<i>Condiciones actuales de Grados Día Desarrollo del bloque actual</i>	124
<b>Figura 63</b>	<i>Información del bloque actual dentro de la pestaña Resumen</i>	125
<b>Figura 64</b>	<i>Selección del bloque</i>	125
<b>Figura 65</b>	<i>Cambio de valores en Session Storage</i>	126
<b>Figura 66</b>	<i>Búsqueda Mensual</i>	126
<b>Figura 67</b>	<i>Pantalla completa del Resumen</i>	127
<b>Figura 68</b>	<i>Sección Humedad</i>	128
<b>Figura 69</b>	<i>Sección Temperatura</i>	128
<b>Figura 70</b>	<i>Sección Grados Días Desarrollo</i>	129
<b>Figura 71</b>	<i>Sección Grados Días Desarrollo por Rango</i>	130
<b>Figura 72</b>	<i>Pruebas Backend de la ruta login</i>	132
<b>Figura 73</b>	<i>Pruebas Backend de la ruta listbloques</i>	132
<b>Figura 74</b>	<i>Pruebas Backend de la ruta bloque</i>	133
<b>Figura 75</b>	<i>Pruebas Backend de la ruta temperatura</i>	133
<b>Figura 76</b>	<i>Pruebas Backend de la ruta humedad</i>	134
<b>Figura 77</b>	<i>Pruebas Backend de la ruta listtemperatura</i>	134
<b>Figura 78</b>	<i>Pruebas Backend de la ruta listhumedad</i>	135
<b>Figura 79</b>	<i>Pruebas Backend de la ruta gddc</i>	135
<b>Figura 80</b>	<i>Pruebas Backend de la ruta gddd</i>	136
<b>Figura 81</b>	<i>Pruebas Backend de la ruta gddrank</i>	136
<b>Figura 82</b>	<i>Pruebas Backend de la ruta resumenTemp</i>	137
<b>Figura 83</b>	<i>Pruebas Backend de la ruta resumenHum</i>	137
<b>Figura 84</b>	<i>Pruebas Backend de la ruta diasDS</i>	138
<b>Figura 85</b>	<i>Pruebas Backend de la ruta dataDS</i>	138
<b>Figura 86</b>	<i>Pruebas Backend de la ruta condicionesDS</i>	139

<b>Figura 87</b> <i>Caso fallido</i> .....	139
<b>Figura 88</b> <i>Pruebas FrontEnd Login</i> .....	140
<b>Figura 89</b> <i>Session Storage cuando inicia sesión</i> .....	141
<b>Figura 90</b> <i>Session Storage cuando se cierra sesión</i> .....	142
<b>Figura 91</b> <i>Pruebas en la sección Dashboard</i> .....	142
<b>Figura 92</b> <i>Session Storage en la sección Dashboard</i> .....	143
<b>Figura 93</b> <i>Pruebas en la sección Humedad</i> .....	144
<b>Figura 94</b> <i>Pruebas en la sección Temperatura</i> .....	144
<b>Figura 95</b> <i>Pruebas en la sección Grados Día Desarrollo</i> .....	145
<b>Figura 96</b> <i>Pruebas en la sección Grados Día Desarrollo por Rango</i> .....	145
<b>Figura 97</b> <i>Pruebas en la sección Estadístico</i> .....	146
<b>Figura 98</b> <i>Pruebas FrontEnd Login en la aplicación móvil</i> .....	147
<b>Figura 99</b> <i>Session Storage cuando inicia sesión en la aplicación móvil</i> .....	148
<b>Figura 100</b> <i>Session Storage cuando se cierra sesión en la aplicación móvil</i> .....	148
<b>Figura 101</b> <i>Pruebas en la sección Resumen de la aplicación móvil</i> .....	149
<b>Figura 102</b> <i>Session Storage en la sección Resumen</i> .....	149
<b>Figura 103</b> <i>Pruebas en la sección Humedad de la aplicación móvil</i> .....	150
<b>Figura 104</b> <i>Pruebas en la sección Temperatura de la aplicación móvil</i> .....	150
<b>Figura 105</b> <i>Pruebas en la sección Grados Día Desarrollo en la aplicación móvil</i> .....	151
<b>Figura 106</b> <i>Pruebas en la sección Grados Día Desarrollo en la aplicación móvil</i> .....	151
<b>Figura 107</b> <i>Resultado de análisis de usabilidad de aplicaciones de monitoreo y visualización de datos</i> .....	152
<b>Figura 108</b> <i>Resultado de análisis de monitoreo y visualización de datos.</i> .....	153
<b>Figura 109</b> <i>Resultado de análisis de consultas de datos.</i> .....	153
<b>Figura 110</b> <i>Resultado de análisis de navegabilidad.</i> .....	154
<b>Figura 111</b> <i>Resultado de análisis de estética.</i> .....	154
<b>Figura 112</b> <i>Resultado de análisis de estética.</i> .....	155
<b>Figura 113</b> <i>Resultado de análisis de tiempo de respuesta.</i> .....	155
<b>Figura 114</b> <i>Resultado de análisis de satisfacción.</i> .....	156

## Resumen

En la actualidad las aplicaciones móviles cuentan con un gran catálogo de opciones, ya que las ventajas que aportan frente a las aplicaciones web son muy notorias teniendo en cuenta la fácil accesibilidad y navegabilidad que presentan ya que se despliegan en cualquier teléfono inteligente en cualquier lugar.

La aplicación móvil y la aplicación web, para el campo florícola, que fue desarrollada en este trabajo permite el seguimiento, monitoreo y consulta de registros históricos de un determinado lugar, lo que optimiza procesos y mejora la toma de decisiones por parte del usuario.

Tanto la aplicación móvil y la aplicación web son independientes, estos consumen servicios de tipo REST mediante un Api web, con el propósito de procesar y manejar información clara y concisa.

Se realizaron las respectivas validaciones en ambas aplicaciones además de ser sometidos a pruebas de usabilidad donde todos los involucrados en el manejo de las herramientas mostraron un grado elevado de satisfacción al emplear su versión web como móvil.

Las ventajas de este sistema van desde un bajo costo de infraestructura, mantenibilidad de código y escalabilidad. El uso de la aplicación móvil y su integración con el sistema desarrollado fomentan el rápido seguimiento y accionar del usuario.

### **PALABRAS CLAVE**

- **WEB API**
- **APLICACIÓN DE MONITOREO**
- **APLICACIÓN MÓVIL**
- **GRADOS DÍA DESARROLLO**

## **Abstract**

Currently mobile applications have a large catalog of options, since the advantages they provide in contrast to web applications are very noticeable taking into account the easy accessibility they present since they are deployed on any smartphone anywhere.

The mobile and web application, for the floricultural field, developed in this work allows the monitoring and consultation of historical records of a certain block, which optimizes processes and improves decision-making by the user.

Both the mobile application and the web application are independent, they consume REST type services through a web Api, in order to handle clear and concise information.

The mobile and web application has very detailed functions which help the user in the learning curve for an efficient and effective handling of it.

The respective validations were carried out in both applications in addition to being subjected to usability tests where all those involved in the management of the tools showed a high degree of satisfaction when using their web version as a mobile.

The advantages of this system range from low infrastructure cost, code maintainability, and scalability. The use of the mobile application and its integration with the developed system promote rapid monitoring and action by the user.

## **KEYWORDS**

- **WEB API**
- **MONITORING APPLICATION**
- **MOBILE APP**
- **DEVELOPMENT DAY DEGREE**

## Capítulo I

### Introducción

En este capítulo se describen antecedentes, objetivos y alcance del proyecto, que fue posible gracias al trabajo realizado de manera conjunta con Docentes del Departamento de Ciencias de la Computación de la Universidad de las Fuerzas Armadas ESPE, así como profesionales de la Carrera de Ingeniería en Ciencias Agropecuarias IASA y un exalumno graduado de la Carrera de Ingeniería de Sistemas e Informática.

#### 1.1. Antecedentes

Cada día, en todo el mundo, la tecnología va evolucionando y mejorando para optimizar la vida cotidianamente, especialmente la de los seres humanos. Estas mejoras o nuevos avances tecnológicos influyen en todo tipo de empresas: grandes, medianas o pequeñas, las cuales se ven en necesidad de adaptarse y explotar este nuevo entorno para sobresalir una de la otra.

El Internet es una tecnología de gran importancia para el uso cotidiano, ya que ofrece soluciones en un corto tiempo al usuario. En el año de 1969 se crea el internet, pero por el año de 1990 se llega a expandir alrededor del mundo, siendo en el año 2015 al realizar una valoración se determinó que dos tercios de la población ya tenían disponible este servicio (Swan, 2012).

El internet es una de las herramientas que más avances tecnológicos ha tenido. Actualmente se encuentra presente en toda área como industrial, salud, educación, comunicación, etc.

La utilización de Big Data y de técnicas de minería de datos hacen posible que las empresas puedan obtener información relevante de sus potenciales clientes, además de mejorar sus niveles de captación.

El Internet de las cosas (IoT) son aquellos ambientes en los que se relacionan la capacidad de cómputo y la conectividad de red con los objetos de uso diario que no se consideran computadoras permitiendo la gestión de datos para el usuario con poca o nula intervención humana.

Durante mucho tiempo se ha querido combinar sensores, equipos electrónicos y redes para monitorear, controlar y gestionar dispositivos, esto hizo que en el mercado tecnológico sean tendencias haciendo aún más cercano la realidad del Internet de las cosas.

La implementación de estos dispositivos IoT transforma varios hábitos y costumbres con los que se vive a diario. Esto lleva a una visión de una casa inteligente en la cual se puede encontrar electrodomésticos, dispositivos para la gestión de la energía y componentes para la automatización del hogar que concluyen con el aumento de la seguridad y eficiencia energética. Otra aplicación de estos dispositivos IoT son la gestión y monitoreo cuando se realiza alguna actividad física o para el área de medicina.

## **1.2. Planteamiento del Problema**

Esta investigación esta aplicada en un entorno controlado en donde existen fluctuaciones en temperatura, humedad, luminosidad, radiación uv (ultravioleta) y calidad de aire, de la misma forma, en el lugar se realiza la producción de flores, la cual está relacionada

directamente con los factores abióticos presentes en el lugar ya que, estos inciden en su fisiología, crecimiento y en la producción.

Actualmente, el proceso de monitoreo y toma de datos ambientales se la realiza de forma manual o con el uso de dispositivos electrónicos que necesariamente requieren de la presencia de una persona para su manipulación y que, además, está limitada a tan solo tres capturas de datos en el día durante todo el ciclo de producción, y a la vez que, esto se ha venido realizando desde algunos años atrás.

El cambio o variación de los factores abióticos inciden directamente en la producción de flores, ya que, estas están sujetas a márgenes y rangos de tolerancia para su óptimo crecimiento y que un monitoreo inadecuado puede producir la disminución de la producción o la pérdida total de la misma.

El monitoreo mediante aplicaciones móviles en procesos de cultivos no es un tema muy abordado, siendo este nuestro punto de partida; se analizó la posibilidad de iniciar con una aplicación móvil la cual permita visualizar datos de las condiciones ambientales, estadísticas e información relevante del proceso de crecimiento de una planta o animal dentro de un invernadero, es este caso, de las flores para poder incidir en la toma de decisiones.

### **1.3. Justificación**

En la actualidad todas las entidades las cuales provean productos o servicios están migrando su información a repositorios digitales sean estos Bases de Datos locales, servidores de terceros o la nube. Toda esta data con el paso del tiempo y utilizando dispositivos IoT se

acumula, siendo este el punto de partida para realizar un correcto tratamiento, posteriormente obtendremos resultados los que podemos aplicar directamente a nuestra planificación futura.

Las Técnicas de Análisis de Datos permiten seguir el comportamiento tanto de una muestra como de un grupo amplio de datos, mediante técnicas estadísticas las cuales nos sugieren rasgos marcados en los cuales indagar, para obtener información relevante. El uso de una aplicación móvil para la visualización de datos permite realizar un monitoreo más completo y rápido para el usuario ya que utiliza como medio un dispositivo móvil inteligente utilizado en el diario vivir.

#### **1.4. Objetivos**

##### **1.4.1. Objetivo General**

Desarrollar una aplicación móvil destinada al monitoreo, análisis y visualización de datos, empleando la metodología Mobile-D, para la toma de decisiones en los procesos de producción de cultivo florícola.

##### **1.4.2. Objetivos Específicos**

- i. Implementar una arquitectura de software adecuada siguiendo lineamientos y parámetros de acuerdo con la necesidad del proyecto el cual maximice su funcionalidad.
- ii. Desarrollar el Backend de la aplicación empleando la plataforma JAVA y su lenguaje de programación, para exponer servicios REST en la Web.
- iii. Desarrollar la Aplicación Móvil empleando el framework IONIC para el monitoreo y visualización de datos.

- iv. Integrar las aplicaciones a través de sus puntos de configuración respectivos, para que trabajen de manera conjunta.

### **1.5. Alcance**

Este proyecto de titulación comprende el planteamiento de una aplicación móvil para monitoreo, análisis y visualización de datos, para la toma de decisiones en los procesos de producción de cultivo florícola. Como estudio de caso se utilizarán datos facilitados por un grupo de trabajo que se vincula con IASA, en la cual se llevará a cabo todo el proceso y seguimiento de la investigación.

### **1.6. Hipótesis**

El correcto manejo de la aplicación facilitará la toma de decisiones en un período determinado del proceso de cultivo.

## Capítulo II

### Marco Teórico

En el presente capítulo se describen las bases teóricas para el desarrollo del presente trabajo, herramientas tecnológicas usadas y metodología aplicada.

#### 2.1. Tecnologías de la Información y Comunicación

##### 2.1.1. Conceptos

De acuerdo con la UNAM, las Tecnologías de Información y Comunicación (TICs) contemplan al conjunto de herramientas relacionadas con la transmisión, procesamiento y almacenamiento digitalizado de la información, como al conjunto de procesos y productos derivados de las nuevas herramientas (hardware y software), en su utilización en la enseñanza. (Luna, 2018).

Uno de los aspectos importantes para la incorporación de las TIC es no plantearnos su utilización simplemente para hacer mejor las cosas que hacemos actualmente, sino fundamentalmente plantearnos hacer cosas diferentes, y que no podríamos hacer sin ellas, o que con ellas las haríamos de forma diferente o más exitosa (Almenara, 2015).

Las TIC han impulsado el avance tecnológico para el intercambio de información mediante internet, a través de los dispositivos tecnológicos actuales como tablets y smartphones los que usan variado software el cual gestiona, muestra información requerida por una persona o un grupo.

El desarrollo tecnológico y la revolución de Internet han puesto en evidencia que la red ha pasado a ser parte de nuestras vidas. Atrás quedaron los años en que la informática era un terreno exclusivo de los militares e ingenieros. El crecimiento y la expansión de la red mundial permitieron llegar a millones de usuarios (Ayala & Gonzales Sánchez, 2015).

## **2.2. Ingeniería de Software**

### **2.2.1. Conceptos**

La ingeniería del software es una disciplina de ingeniería que comprende todos los aspectos de la producción de software desde las etapas iniciales de la especificación del sistema, hasta el mantenimiento de este después de que se utiliza (Sommerville, 2005).

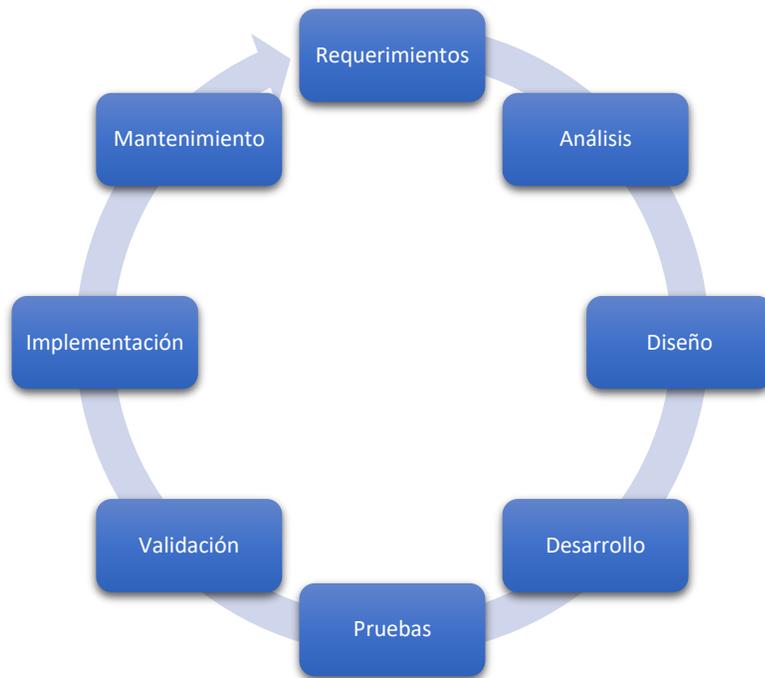
Estudia toda relación que se tiene con la informática y la ciencia de computación, ya que, utilizando metodologías, técnicas y herramientas brindan características esenciales para la óptima ejecución y la conservación del software.

La ingeniería de software nos permite elaborar sistemas complejos con alta calidad en un tiempo adecuado que son aplicables a todo tipo de áreas: medicina, educación, banca, arquitectura, etc.

El ciclo de vida del software es muy importante en la elaboración de un sistema o aplicativo informático proveyendo de procesos, actividades y tareas necesarias que se involucran dentro del desarrollo desde la toma de los requisitos hasta la culminación de su uso.(ISO IEC 12207, 2008)

*Al pasar el tiempo, las etapas del ciclo de vida de un software fueron cambiando por varios motivos: las necesidades de los usuarios, nuevas tecnologías, nuevas herramientas informáticas, etc. Hoy en día se puede nombrar las siguientes etapas del ciclo de vida del software descritas en la figura.*

**Figura 1**  
*Ciclo de vida del software*



*Nota.* La imagen presenta el proceso sistemático del ciclo de vida del software.

**Definición de Requerimientos:** Recopilación de información sobre la necesidad del usuario.

**Análisis de requerimientos y su viabilidad:** Examina y obtiene los requisitos del usuario de una forma más precisa y clara, teniendo en cuenta cualquier restricción, usabilidad de estándar, seguridad y legalidad.

**Diseño:** Se realiza un diseño general de la arquitectura que tendrá el software. Luego se detalla cada componente y su funcionalidad dentro de la arquitectura general.

**Desarrollo/Programación:** Se crean las funciones que se definieron en la etapa de diseño utilizando lenguajes de programación.

**Pruebas:** Varios tipos de pruebas son ejecutadas en esta etapa: las pruebas unitarias, pruebas de integración, pruebas funcionales, pruebas de estrés, etc.

Cabe mencionar que en la etapa de desarrollo/programación también se realizan pruebas para comprobar el correcto comportamiento de la función

**Validación:** También llamadas pruebas beta, garantizan que el software cumpla con los requerimientos originales. Además, al software se le realiza pruebas de aceptación. (Moreno, 2018)

**Implementación:** Se despliega el software en un entorno e infraestructura segura, se realizan pruebas de funcionalidad y si es necesario, una capacitación al usuario final. (Conislla, 2020)

**Mantenimiento:** Son procedimientos correctivos y actualizaciones que se realizan al software, también conocidos como mantenimiento correctivo y mantenimiento continuo respectivamente.(Moreno, 2018)

## 2.3. Arquitectura Cliente Servidor

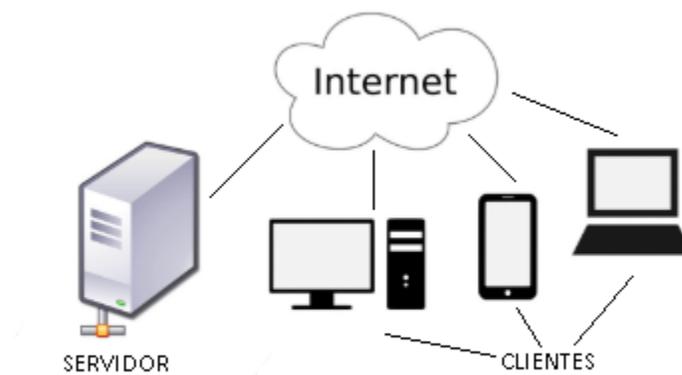
### 2.3.1. Conceptos

La arquitectura cliente servidor está compuesto de dos partes: la del servidor y la del cliente o grupo de clientes.(Vaquero, 2019)

La parte servidor es aquel o aquellos equipos informáticos que dispone de un software en donde se almacena y gestiona toda la información requerida por el cliente.

La parte del cliente o grupo de clientes realizan peticiones al servidor según la necesidad que se tiene y recibe una respuesta.

**Figura 2**  
*Arquitectura cliente-servidor*



*Nota.* La imagen presenta muestra un ejemplo de la arquitectura cliente-servidor.

## **2.4. Herramientas tecnológicas utilizadas**

A continuación, se presentará una reseña de todas las herramientas empleadas para el desarrollo del proyecto.

### **2.4.1. Modelamiento UML**

#### **2.4.1.1. Power Designer**

Power Designer es una herramienta de apoyo para modelamiento de una organización desde la capa de negocio hasta la de tecnológica, nos permite generar modelos conceptuales, lógicos y físicos para los diferentes gestores de base de datos que existen en el mercado, es una herramienta muy completa que nos permite gestionar todas nuestras entidades a emplear y tener un primer acercamiento a cómo será el producto final, en este caso de nuestra base de datos.

PowerDesigner es una poderosa herramienta basada en una tecnología orientada a alinear el negocio y la Tecnología de Información (TI); es una solución de modelado y diseño empresarial que colabora en la implementación efectiva de la arquitectura empresarial y brinda técnicas poderosas de análisis y diseño durante todo el ciclo de vida de desarrollo del proyecto con gestión de meta-datos, funciones de análisis de impacto y verdadero repositorio empresarial (skrapy95, 21:44:28 UTC).

## **2.4.2. Entornos de desarrollo integrado (IDE)**

### **2.4.2.1. Spring boot**

Spring Boot es una herramienta que nace con la finalidad de simplificar aún más el desarrollo de aplicaciones basadas en el ya popular framework Spring Core. Spring Boot busca que el desarrollador solo se centre en el desarrollo de la solución, olvidándose por completo de la compleja configuración que actualmente tiene Spring Core para poder funcionar (Oblancarte, 2018).

Mediante el uso de Spring boot se busca construir aplicaciones de una manera mucho más rápida, evitando de cierta manera la configuración tanto de servidores como dependencias necesarias para el funcionamiento de estas. Además, las aplicaciones de Spring boot son fáciles de configurar ya que su gestor de dependencias nos muestra cuando estas pueden ser mejoradas o reemplazadas en el caso de que estén depreciadas, es posible personalizar fácilmente una aplicación de Spring Boot, tanto en la configuración inicial como, en su desarrollo, por lo mismo se empleó Spring boot en el desarrollo del Backend.

El framework Spring es muy versátil pues se divide en módulos diseñados para solventar diferentes necesidades. Por ejemplo, el módulo Spring-Core nos permite reducir líneas de código; es decir, si tenemos un bloque de código de unas 50 líneas nos la reduce a 3 líneas. Y el módulo relacionado a la configuración de visibilidad y gestión de objetos, unifica distintas APIs de gestión y coordina dichas transacciones, además destacan las siguientes ventajas:

- Desarrollo web sobre REST API; con el cual el framework spring hace gestión de la petición, realiza el análisis correspondiente de datos y luego decide en qué clase de Java procesará los datos.
- Permite el desarrollo de aplicaciones flexibles y escalables; es decir, a medida que crezca el programa, su mantenimiento siempre será fácil pues el código será fácil de entender.
- Promueve una alta cohesión (enfoque preciso del propósito de una clase) y un bajo acoplamiento (interconexión o dependencia entre clases).
- Promueve el uso de POJO's para su codificación. POJO's se refiere al uso de clases simples y que no dependen de un framework en especial. Objetos creados simplemente con miembros y estos con sus getters y setter correspondientes y no heredan de nadie.
- Está basado en la programación orientada a interfaces. (significa que podemos cambiar la implementación de una clase de manera programativa o declarativa en tiempo de ejecución).
- Utiliza el patrón DI (Dependency Injection), este patrón gestiona las dependencias de objetos entre las clases de la aplicación, en lugar de ser ellas quienes las gestionen.
- Utiliza AOP (programación orientada a aspecto) paradigma que permite desacoplar ciertas responsabilidades mejorando la escalabilidad de la aplicación ("Qué es el Framework Spring y las ventajas de utilizarlo", 2019).

#### **2.4.2.2. Angular 9**

Angular es un framework de desarrollo para JavaScript creado por Google. La finalidad de Angular es facilitarnos el desarrollo de aplicaciones web SPA y además darnos herramientas para trabajar con los elementos de una web de una manera más sencilla y optima (“¿Que es Angular y para qué sirve?”, 2017).

Angular 9 posee una mejora notable en su versión actual en cuanto a escalabilidad y mantenimiento del código, una de las características más notorias de este framework es la posibilidad de poder separar la lógica del negocio del resto de la aplicación, por este motivo esta herramienta es empleada para el desarrollo de nuestro Frontend.

El diseño de Angular adopta el estándar de los componentes web. Se trata de un conjunto de APIs que te permiten crear nuevas etiquetas HTML personalizadas, reutilizables y autocontenidas, que posteriormente se pueden utilizar en otras aplicaciones web. Estos componentes personalizados funcionarán en navegadores modernos y con cualquier biblioteca o framework de JavaScript que trabaje con HTML. Los componentes que creas en Angular son fáciles de convertir en componentes web nativos. A largo plazo esto es una gran ventaja pues te permitirá reutilizar componentes creados en Angular a otro tipo de aplicaciones, reutilizando tu trabajo (*Ventajas de utilizar Angular, un framework JavaScript, 2018*).

#### **2.4.2.3. Ionic 5**

Ionic es una estructura tecnológica (Framework) de código abierto que se utiliza en el desarrollo de aplicaciones móviles híbridas, es decir, se combinan el HTML5, CSS y JavaScript dando como resultado aplicaciones con una interfaz amigable e intuitiva para el usuario que

luego se comercializan o descargan en plataformas como Android o IOs (“[✓Qué es Ionic | Quality Devs | Somos Desarrolladores](#)”, 2019).

Ionic Framework es un kit de herramientas de interfaz de usuario móvil de código abierto para crear experiencias de aplicaciones web y nativas multiplataforma de alta calidad. Muévase más rápido con una sola base de código, ejecutándose en todas partes (Ionic, 2015).

Ionic es de código abierto y gratuito y sirve para diseñar aplicaciones híbridas a partir de HTML5 de manera fácil y bastante simple. Ionic, como software de desarrollo de aplicaciones híbridas, es un kit de desarrollo de software (SDK) completo y de código abierto. Basado en AngularJS, Ionic utiliza tecnologías web como CSS, HTML5 y Sass para elaborar aplicaciones móviles híbridas. Fue creado en 2013 y es conocido por tratarse de una plataforma orientada a que los programadores desarrollen aplicaciones web y móviles.

Su amplia gama de herramientas y servicios, junto con el uso de AngularJS, lo convierten en una plataforma ideal para crear aplicaciones altamente interactivas. Además de todo esto, Ionic cuenta con una interfaz de línea de comandos (CLI, por sus siglas en inglés) que permite construir y probar aplicaciones iónicas en cualquier plataforma.

Por un lado, las principales ventajas de Ionic son:

- Es un entorno compatible con todas las plataformas (iOS y Android).
- Utiliza lenguajes muy conocidos por los desarrolladores (HTML, CSS y JS), de modo que su implantación en los equipos es relativamente fácil.

- Da soporte al entorno AngularJS y estudia incluir otros, como Ember.js o Knockout.
- Se puede reutilizar el código de una aplicación web en una aplicación móvil.
- En el desarrollo de aplicaciones híbridas, solo se necesita un único proceso de desarrollo e implantación para Android, iOS y web (*Desarrollo de aplicaciones híbridas con entorno Ionic, 2019*).

#### **2.4.2.4. Cordova 9**

Es un framework de código abierto que permite la codificación para crear aplicaciones móviles utilizando HTML5, CSS y JavaScript. Además, Cordova tiene una variedad de complementos que permiten a la aplicación trabajar con las funcionalidades de dispositivos, como sensores, cámaras, entre otros.

La creación de aplicaciones móviles para las distintas plataformas (iOS y Android) es posible ya que Cordova es compatible con todas las bibliotecas de JavaScript, es decir, se pueden instalar en los dispositivos móviles de los usuarios, de igual manera como una aplicación nativa, sin importar que la aplicación móvil se haya creado con tecnología web. (Watson, 2019)

#### **2.4.3. Servidor Web**

##### **2.4.3.1. TomCat**

Apache Tomcat, el servidor web usado tradicionalmente para proyectos Java por su implementación de servlets o páginas JSP, es otra de las aplicaciones que podemos desplegar fácilmente en los Servidores Cloud de Arsys desde el Catálogo de Aplicaciones. Apache Tomcat (o, sencillamente, Tomcat) es un contenedor de servlets que se puede usar para compilar y ejecutar aplicaciones web realizadas en Java. Implementa y da soporte tanto a servlets como a

páginas JSP (Java Server Pages) o Java Sockets. Tomcat puede funcionar de manera autónoma como motor de aplicaciones web desarrolladas con Java, aunque habitualmente se usa en combinación con otros productos como el servidor web Apache, para dar un mayor soporte a tecnologías y aumentar sus características (Arsys, 2017).

#### **2.4.4. Base de Datos**

##### **2.4.4.1. MySQL**

Es una herramienta para gestión de base de datos (SGBD) de código abierto. El SGBD MySQL pertenece actualmente a Oracle. Funciona con un modelo cliente-servidor. Eso quiere decir que los ordenadores que instalan y ejecutan el software de gestión de base de datos se denominan clientes. Todas las ocasiones en que los clientes deseen establecer conexión con el servidor este les solicitara las respectivas credenciales de acceso.

##### **Ventajas:**

- MySQL software es Open Source
- Velocidad al realizar las operaciones, lo que le hace uno de los gestores con mejor rendimiento.
- Bajo costo en requerimientos para la elaboración de bases de datos, ya que debido a su bajo consumo puede ser ejecutado en una máquina con escasos recursos sin ningún problema.
- Facilidad de configuración e instalación.
- Soporta gran variedad de Sistemas Operativos
- Baja probabilidad de corromper datos, incluso si los errores no se producen en el propio gestor, sino en el sistema en el que está.

- Su conectividad, velocidad, y seguridad hacen de MySQL Server altamente apropiado para acceder bases de datos en Internet
- El software MySQL usa la licencia GPL (Madayeliperezhernandez309, 2017).

#### 2.4.5. Editor de Código

##### 2.4.5.1. Visual Code

Es la herramienta para edición de código de Microsoft que en la actualidad es un editor muy completo y proporciona una serie de paquetes los cuales permiten el control y manejo de gran variedad de fuentes de código de una manera muy versátil, además esta herramienta está en constante mantenimiento por lo que errores asociados con sus extensiones son corregidos rápidamente permitiendo realizar la codificación de una manera muy ágil.

#### 2.5. Api Rest

##### 2.5.1. Conceptos

##### 2.5.2. Características

- **Escalabilidad.** Este protocolo destaca por su escalabilidad. Gracias a la separación entre el cliente y el servidor, el producto se puede escalar con un equipo de desarrollo sin que ello represente muchas dificultades.
- **Flexibilidad y Portabilidad.** Con el requisito imprescindible de que los datos de cada una de las peticiones sean enviados de forma correcta, es posible realizar una migración de un servidor a otro o practicar cambios en la base de datos en todo momento. De esta forma el front y el back se pueden alojar en servidores diferentes, lo que supone una enorme ventaja de manejo.

- **Independencia.** Debido a la separación entre el cliente y el servidor, el protocolo facilita que los desarrollos de las diferentes partes de un proyecto se puedan dar de manera independiente. Además de ello, la **API REST** se adapta en todo momento al tipo de sintaxis o plataformas de trabajo. Esto brinda la oportunidad de probar varios entornos dentro del desarrollo (“¿CUÁLES SON LAS VENTAJAS DE UNA API REST?”, 2017).

## 2.6. Aplicaciones Multiplataforma

### 2.6.1. Conceptos

### 2.6.2. Características

El desarrollo multiplataforma, a diferencia del desarrollo nativo, se centra en el reuso de código. La construcción de aplicaciones web móviles constituye un ejemplo que representa este enfoque. Sin embargo, las limitaciones derivadas de su ejecución dentro de un navegador, ha motivado a los ingenieros de software a dirigir su atención hacia otro tipo de aplicaciones multiplataforma con el que se obtienen resultados más cercanos a las soluciones nativas (Delía, 2017).

### 2.6.3. Ventajas

Desarrollar aplicaciones para distintos sistemas operativos utilizando tecnologías multiplataforma puede ofrecerte múltiples ventajas:

- Mayor alcance de la aplicación. Al ofrecer tu aplicación a una variedad más numerosa de dispositivos y de canales de distribución, tu público también será más amplio.
- Ahorro de tiempo y dinero. Al desarrollar una aplicación para varias plataformas de forma simultánea evitamos duplicar tiempo y esfuerzos, lo que se traduce en proyectos más económicos.

- Diseño unificado en todos los dispositivos. Al diseñar la aplicación a la vez para todas las plataformas, el diseño será transversal a todos los sistemas operativos, y proporcionará una imagen uniforme de la aplicación.
- Mantenimiento más sencillo. Es posible solucionar bugs y actualizar la aplicación de forma centralizada y en paralelo, pues un mismo proyecto nos genera versiones para todas las plataformas disponibles.

## **2.7. Metodología**

### **2.7.1. Programación Extrema (XP)**

#### **2.7.1.1. Conceptos**

Se basa en las mejores prácticas para el desarrollo de software, misma que utiliza el modelo iterativo incremental para lograr una rápida entrega del software y mejorar las capacidades de la gestión del riesgo (Calvo, 2018). Generalmente la programación extrema se basa en cuatro procesos:

**Figura 3**  
Fases de la metodología programación extrema



*Nota.* La imagen presenta el proceso de la metodología de la programación extrema y las tareas de cada uno de ellos.

### 2.7.1.2. Ventajas

Utilizando la metodología de programación extrema se tiene varias ventajas, entre ellas:

- Software sometido a continuas pruebas para su estabilidad
- Mitigación de errores al programar en parejas
- Comprensión sencilla de código
- Trabajo de programación innecesaria casi nula
- Rapidez en implementación de las actualizaciones

La programación extrema se ajusta a las necesidades del cliente, ya que los requisitos pueden ir cambiando a lo largo de este proceso, disminuyendo el tiempo de desarrollo de una aplicación publicando y realizando pruebas en ciclos cortos para eliminar y corregir los errores que se vayan presentando. (IONOS, 2019)

## 2.7.2. Mobile-D

### 2.7.2.1. Conceptos

Esta metodología permite obtener ciclos de desarrollos demasiado rápidos implementado en equipos no menos de diez desarrolladores, siempre y cuando, trabajen en un mismo espacio físico. Si se utiliza esta metodología se debería obtener productos funcionales en poco tiempo.

**Figura 4**  
*Fases de la metodología Mobile-D*

Exploración	Inicialización	Fase de Producto	Fase de estabilización	Fase de pruebas y Reparación
<ul style="list-style-type: none"> <li>•Se dedica a la planificación y conceptos básicos del proyecto.</li> </ul>	<ul style="list-style-type: none"> <li>•Se preparan e identifican todos los recursos necesarios</li> <li>•Se establece un entorno técnico</li> </ul>	<ul style="list-style-type: none"> <li>•Se repiteren subfases con tareas planificadas según los días de trabajo</li> </ul>	<ul style="list-style-type: none"> <li>•Se llevan a cabo las acciones de integración para asegurar el correcto funcionamiento del sistema</li> </ul>	<ul style="list-style-type: none"> <li>•Tiene como objetivo la disponibilidad y correcta funcionalidad del sistema</li> </ul>

*Nota.* La imagen presenta las fases de la metodología Mobile-D y las tareas de cada sub-fase.

Mobile-D utiliza métodos conocidos y consolidados como solución para desarrollo de una aplicación o sistema. Extreme Programming (XP) es utilizado para las prácticas de desarrollo, Crystal Methodologies para escalar los métodos y Rational Unified Process (RUP) como base en el diseño del ciclo de vida (Ramírez Vique, 2016).

### 2.7.2.2. Ventajas

Utilizando la metodología de programación extrema se tiene varias ventajas, entre ellas:

- Desarrollo de software en un corto tiempo
- Rápida implementación de cambios en los requerimientos
- Solución para creación de aplicaciones móviles
- Aplicación en cualquier herramienta de desarrollo para aplicaciones móviles

## Capítulo III

### Desarrollo

En el presente capítulo se describen las metodologías, así como cada una de sus fases para el desarrollo del proyecto. Teniendo en cuenta que el alcance del proyecto está enfocado en la entrega de una aplicación móvil multiplataforma. Las fases desarrolladas siguieron las metodologías antes mencionadas resaltando los aspectos de: Análisis y Planificación, Especificación de Requisitos, Diseño, Desarrollo, y Pruebas.

De esta misma manera también se actualizo todo lo referente al proyecto en su versión inicial como lo es su Backend y Frontend, ya que esto es un módulo el cual formara parte del sistema el cual pretende continuar su crecimiento de manera modular.

#### 3.1.1. Problema

En la actualidad el monitoreo a campos florícolas es periódico y se lo realiza de una manera tradicional, siendo esta la medición de sus condiciones por parte de los empleados que recopilan información limitada al número de registros en cada revisión, desde los registros que muestran la evolución del cultivo, así como los resultados que se esperan obtener de este, entre varios otros procesos que detallan el crecimiento o su progreso diario.

Estos precedentes limitan el número de registros confiables en la obtención de información relevante para la producción y el cuidado del campo florícola como de sus cultivos.

### 3.1.2. Análisis de viabilidad

**Tabla 1**

*Búsqueda en base de datos establecidas*

<b>Tipo</b>	<b>Descripción</b>
<b>Factibilidad Técnica</b>	El grupo de anteriores colaboradores. se encarga de proveer todas las herramientas necesarias para continuar trabajando con sus investigaciones, en el caso del presente proyecto se utilizarán en su mayoría herramientas de código abierto, enfocándose en el manejo y presentación de la data, así como documentación de tipo científica colgada en la red. Se hará uso de los equipos personales de los estudiantes a cargo del proyecto para el desarrollo de la investigación y la posterior validación de esta.
<b>Factibilidad Operativa</b>	El proyecto cuenta con el apoyo de ingenieros y colaboradores, los cuales brindarán apertura y acceso a datos e información necesaria para el desarrollo de la investigación, así como herramientas las cuales ayuden al desarrollo de esta; Además, contamos con la ayuda de nuestro tutor que brinda sus comentarios y observaciones referente al tema.
<b>Factibilidad Económica</b>	Todo el proyecto será autofinanciado por los estudiantes y colaboradores.

*Nota.* La tabla muestra la viabilidad del problema presentado en el capítulo II.

## 3.2. Especificaciones de Requerimientos

En esta etapa nos corresponde fijar los procesos, así como objetivos los que nos permiten desarrollar el proyecto.

### 3.2.1. Alcance del Sistema

Se ha diseñado de acuerdo con lo requerido un módulo de monitoreo, el mismo permite la visualización de información actual como histórica y estadísticas de un determinado bloque florícola.

Se ha creado una base de datos en MySQL y freesqldb.com con los datos necesarios para emular la base de datos que contiene el sensor.

Se ha implementado un "job" en la base de datos y se ejecutará al finalizar el día, también llevarán a cabo ciertos procesos con el fin de tener registros históricos en nuestra base de datos.

Se ha desarrollado un Web API REST el mismo que responde a todas las peticiones solicitadas por nuestros usuarios, siendo independiente a su versión web como móvil.

Se diseñaron interfaces de usuario para cada requerimiento, como para todos los procesos que lo componen teniendo en cuenta los filtros solicitados.

El sistema debe permitir el acceso a usuarios tanto vía web como por su aplicación móvil.

El sistema permite visualizar la información relevante de un bloque florícola (Nombre, Extensión, Ubicación, Temperatura, Humedad)

El sistema debe visualizar en tiempo real las características de un determinado bloque florícola (Humedad, Temperatura).

El sistema permite la visualización estadística de datos con respecto al crecimiento o desarrollo del cultivo en lapso de un mes seleccionable.

El sistema permite visualizar datos actuales como históricos con respecto a las condiciones climáticas del bloque florícola (Temperatura, Humedad, Grados Día Desarrollo), en un periodo seleccionable por el usuario.

El sistema permite visualizar un resumen de condiciones en un periodo seleccionable por el usuario el cual este compuesto por una fecha inicial y una fecha final.

El sistema tanto en su versión móvil como web responde a cada una de estas características únicamente variando en su presentación por el comportamiento de sus componentes dependiendo el tipo de dispositivo (PC, Tablet, smartphone).

### **3.2.2. Perspectiva del Sistema**

El sistema se ha denominado GreenHouse, el desarrollo de este estuvo centrado en mejorar su versión preliminar y dar un salto en el empleo de tecnologías actuales, para con esto poder escalar el producto posteriormente. GreenHouse consta de 2 partes importantes el Backend y el Frontend este último cambia de acuerdo con el dispositivo que lo despliegue.

Con respecto al Backend la aplicación se conectara a un base de datos en la nube la cual es la encargada de almacenar toda la data y emular el comportamiento del sensor, esta misma base esta parametrizada y mapeada dentro de nuestra API desarrollada con Spring boot, la misma que nos permite la gestión de información de la base de datos dependiendo la petición o función que se solicite y devolviendo información en formato JSON el cual viaja por internet mediante protocolos HTTP a nuestro dispositivos cliente.

El Frontend está segmentado tanto a dispositivos móviles como cualquier tipo de pc, ya que por la parte Web este se encuentra desarrollado con angular 9 el cual nos permite el manejo de respuestas con nuestro backend y el intercambio de información siendo nuestro frontend el encargado de armar las peticiones o consultas los cuales se enviaran a nuestra Api para que entregue la información solicitada, así mismo nuestra app móvil se encuentra desarrollada con IONIC 5, el mismo que emplea componentes de angular como varias funcionalidades de este que permite ejecutar dichas acciones desde nuestro dispositivo móvil, únicamente variando la presentación de la información.

### **3.3. Diseño del Sistema**

En esta sección se realizó una vista preliminar de como quedara estructurado el proyecto en cuanto a su arquitectura, herramientas, diagramas de caso de uso, diseños de interfaces, bases de datos, y manejo de Respuestas por parte de nuestros Servicios.

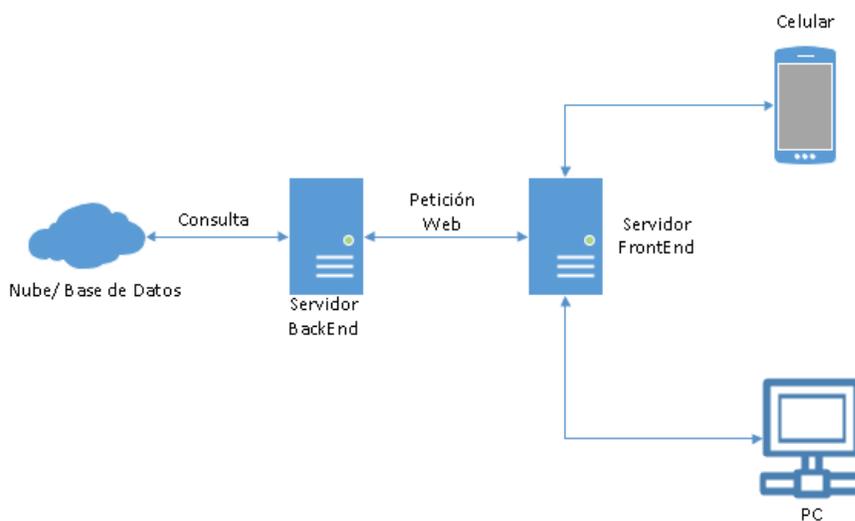
#### **3.3.1. Arquitectura**

La selección de arquitectura se basó tanto en las capacidades y beneficios que nos ofrece Spring boot al momento de desarrollar y también en las prestaciones de los servicios rest además de estar enfocados en que la aplicación se escalara a posterior de manera modular. Dicho esto, el producto cuenta con un servidor de base de datos con MySQL 5.6 bajo un sistema operativo Windows 10 de 64 bits para ejecución local y una base de pruebas en freesqldb.com para emular el sensor.

Para nuestro backend se cuenta con un servidor web tomcat propio de spring boot el cual es el encargado de desplegar nuestra aplicación y exponer nuestros servicios, soportado por un sistema Windows 10 de 64 bits. Los puertos de comunicación habilitados entre nuestra base de datos y nuestra aplicación son el 5000 para el Api REST y el 3306 para nuestra base de datos.

Por parte del frontend este se despliega en el servidor embebido de angular posteriormente esto se trasladará a un IIS levantado sobre Windows 10 de 64 bits. Los puertos para usar dependerán de la configuración que se asigne posteriormente ya que el paso a producción del frontend se traduce en una configuración por parte del servidor.

**Figura 5**  
*Arquitectura de la aplicación.*



*Nota.* La imagen presenta la arquitectura de la aplicación web y móvil diseñada para satisfacer el problema.

### 3.3.2. Diagramas y Casos de uso

**Figura 6**

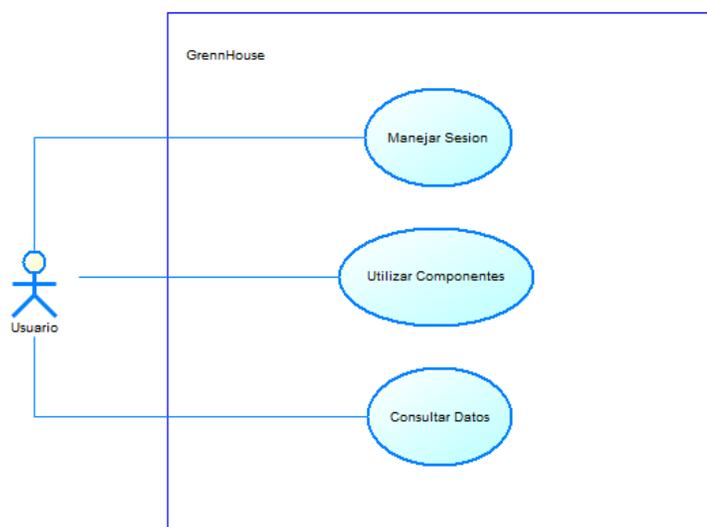
*Diagrama nivel 0 de casos de uso para el ingreso a la aplicación*



*Nota.* La imagen presenta el caso de uso para ingresar a la aplicación web y móvil.

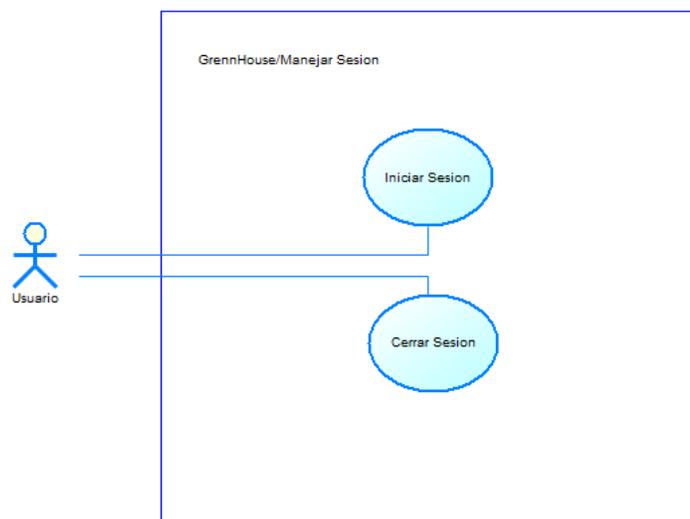
**Figura 7**

*Diagrama nivel 1 de casos de uso de la aplicación Green House*



*Nota.* La imagen presenta el caso de uso entre el usuario y de toda la aplicación web y móvil.

**Figura 8**  
*Diagrama nivel 2 de casos de uso de manejar sesión*



*Nota.* La imagen presenta el caso de uso entre el usuario y el manejo de sesión

**Tabla 2**

*Caso de uso iniciar sesión*

<b>CU-01</b>	<b>Iniciar Sesión</b>
<b>Actor</b>	Administrador, Usuario
<b>Descripción</b>	El sistema deberá responder a la petición de inicio de sesión de un determinado usuario.
	<b>Paso    Acción</b>
<b>Secuencia</b>	1    Se ingresan las credenciales (usuario, contraseña)
	2    Se presiona el botón "Iniciar Sesión"

<b>CU-01</b>	<b>Iniciar Sesión</b>	
3		Se envía un a petición hacia el web Api con las credenciales
4		Se verifica la conexión y permisos de petición
5		Se accede al servicio requerido
6		El sistema valida los datos recibidos
7		El sistema responde un objeto tipo JSON con la información del usuario solicitado
8		Se confirma la existencia del usuario
9		Se crea una variable de sesión en el navegador
10		Se direcciona a la pantalla de inicio de la aplicación
		Se muestra una alerta: "¡OK, Consulta Correcta!"
		Se crea un variable de sesión con datos del usuario y bloque
		Se carga toda la información del usuario, empresa y el primer bloque registrado
<b>Postcondición</b>		Se carga las estadísticas del bloque predeterminado con respecto al último mes
		Se carga el resumen de condiciones del bloque predeterminado con respecto al último mes
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>

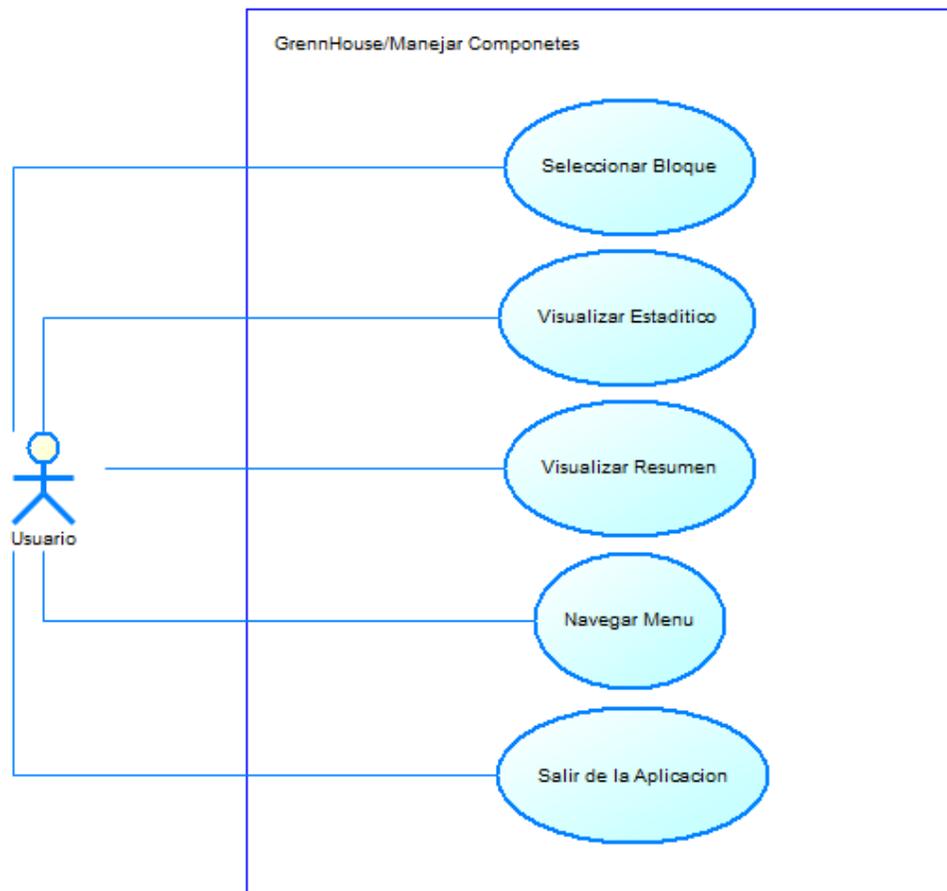
CU-01	Iniciar Sesión
4	<p>De no poder conectarse con los servicios:</p> <hr/> <p>E.1.- El sistema mostrará una alerta: "ERROR, No se encontraron los datos"</p> <hr/> <p>E.2.- El caso de uso termina</p>
6	<p>De no comprobar que los datos son correctos estén vacíos o tengan errores de tipeo.</p> <hr/> <p>E.1.- El sistema mostrará una alerta: "ERROR, No se encontraron los datos"</p> <hr/> <p>E.2.- El sistema resaltara los campos requeridos con rojo.</p> <hr/> <p>E.3.- El caso de uso termina</p>
<b>Comentarios</b>	<p>No hay un número máximo de intentos</p> <hr/> <p>Las cuentas no se bloquean por intentos erróneos</p>
<p><i>Nota.</i> La tabla presenta el desglosamiento del caso de uso entre el usuario y el inicio de sesión</p>	

**Tabla 3***Caso de uso cerrar sesión*

<b>CU-02</b>	<b>Cerrar Sesión</b>								
<b>Actor</b>	Administrador, Usuario								
<b>Descripción</b>	El sistema deberá responder al cierre de sesión de un determinado usuario.								
	<table border="1"> <thead> <tr> <th><b>Paso</b></th> <th><b>Acción</b></th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Se dirige a la pestaña junto a la imagen del usuario (parte superior derecha)</td> </tr> <tr> <td>2</td> <td>Se presiona el botón "Cerrar Sesión"</td> </tr> <tr> <td>3</td> <td>Se destruyen todas las variables de sesión en el navegador</td> </tr> </tbody> </table>	<b>Paso</b>	<b>Acción</b>	1	Se dirige a la pestaña junto a la imagen del usuario (parte superior derecha)	2	Se presiona el botón "Cerrar Sesión"	3	Se destruyen todas las variables de sesión en el navegador
<b>Paso</b>	<b>Acción</b>								
1	Se dirige a la pestaña junto a la imagen del usuario (parte superior derecha)								
2	Se presiona el botón "Cerrar Sesión"								
3	Se destruyen todas las variables de sesión en el navegador								
<b>Secuencia</b>									
<b>Postcondición</b>	Se destruyen las variables de sesión								
	Se redirecciona a la pantalla de inicio de sesión								
	<table border="1"> <thead> <tr> <th><b>Paso</b></th> <th><b>Acción</b></th> </tr> </thead> <tbody> <tr> <td>NA</td> <td></td> </tr> </tbody> </table>	<b>Paso</b>	<b>Acción</b>	NA					
<b>Paso</b>	<b>Acción</b>								
NA									
<b>Excepciones</b>									
<b>Comentarios</b>	NA								

*Nota.* La tabla presenta el desglosamiento del caso de uso entre el usuario y el cierre de sesión

**Figura 9**  
*Diagrama nivel 2 de casos de uso de manejar componentes*



*Nota.* La figura presenta el caso de uso entre el usuario y el manejo de componentes

Tabla 4

*Caso de uso seleccionar bloque*

<b>CU-03</b>	<b>Seleccionar Bloque</b>																		
<b>Actor</b>	Usuario																		
<b>Descripción</b>	El sistema deberá permitir la selección de un bloque determinado, mediante una lista de bloques disponibles																		
	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Se pulsa sobre la selección de bloque</td> </tr> <tr> <td>2</td> <td>Se selecciona el bloque deseado</td> </tr> <tr> <td>3</td> <td>Se envía un a petición hacia el web Api empleando filtros de la variable de sesión del navegador y el bloque seleccionado.</td> </tr> <tr> <td>4</td> <td>Se verifica la conexión y permisos de petición</td> </tr> <tr> <td>5</td> <td>Se accede al servicio requerido</td> </tr> <tr> <td>6</td> <td>El sistema valida los datos recibidos</td> </tr> <tr> <td>7</td> <td>El sistema responde un objeto tipo JSON con la información del bloque seleccionado</td> </tr> <tr> <td>8</td> <td>Se cargan los datos en los respectivos campos.</td> </tr> </tbody> </table>	Paso	Acción	1	Se pulsa sobre la selección de bloque	2	Se selecciona el bloque deseado	3	Se envía un a petición hacia el web Api empleando filtros de la variable de sesión del navegador y el bloque seleccionado.	4	Se verifica la conexión y permisos de petición	5	Se accede al servicio requerido	6	El sistema valida los datos recibidos	7	El sistema responde un objeto tipo JSON con la información del bloque seleccionado	8	Se cargan los datos en los respectivos campos.
Paso	Acción																		
1	Se pulsa sobre la selección de bloque																		
2	Se selecciona el bloque deseado																		
3	Se envía un a petición hacia el web Api empleando filtros de la variable de sesión del navegador y el bloque seleccionado.																		
4	Se verifica la conexión y permisos de petición																		
5	Se accede al servicio requerido																		
6	El sistema valida los datos recibidos																		
7	El sistema responde un objeto tipo JSON con la información del bloque seleccionado																		
8	Se cargan los datos en los respectivos campos.																		
<b>Secuencia</b>																			
	Se muestra una alerta: "¡OK, Consulta Correcta!"																		
<b>Postcondición</b>	Se actualiza la variable se sesión con el código del bloque																		

<b>CU-03</b>	<b>Seleccionar Bloque</b>														
Se setea el bloque en el sistema.															
	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td></td> <td>De no poder conectarse con los servicios:</td> </tr> <tr> <td>4</td> <td>E.1.- El sistema mostrará una alerta de que no hay respuesta, se obtendrá un error 404</td> </tr> <tr> <td></td> <td>E.2.- El caso de uso termina</td> </tr> <tr> <td><b>Excepciones</b></td> <td>De no comprobar que los datos son correctos estén vacíos o tengan errores de tipeo.</td> </tr> <tr> <td>6</td> <td>E.1.- El sistema mostrara una alerta de que no se encontraron resultados, se devolverá un JSON con un mensaje de "ERROR"</td> </tr> <tr> <td></td> <td>E.2.- El caso de uso termina</td> </tr> </tbody> </table>	Paso	Acción		De no poder conectarse con los servicios:	4	E.1.- El sistema mostrará una alerta de que no hay respuesta, se obtendrá un error 404		E.2.- El caso de uso termina	<b>Excepciones</b>	De no comprobar que los datos son correctos estén vacíos o tengan errores de tipeo.	6	E.1.- El sistema mostrara una alerta de que no se encontraron resultados, se devolverá un JSON con un mensaje de "ERROR"		E.2.- El caso de uso termina
Paso	Acción														
	De no poder conectarse con los servicios:														
4	E.1.- El sistema mostrará una alerta de que no hay respuesta, se obtendrá un error 404														
	E.2.- El caso de uso termina														
<b>Excepciones</b>	De no comprobar que los datos son correctos estén vacíos o tengan errores de tipeo.														
6	E.1.- El sistema mostrara una alerta de que no se encontraron resultados, se devolverá un JSON con un mensaje de "ERROR"														
	E.2.- El caso de uso termina														
<b>Comentarios</b>	No se permite ingresar manualmente un bloque														

**Nota.** La tabla presenta el desglosamiento del caso de uso entre el usuario y el seleccionar bloque.

**Tabla 5**

*Caso de uso Visualizar Estadístico*

<b>CU-04</b>	<b>Visualizar Estadístico</b>
<b>Actor</b>	Usuario

<b>CU-04</b>	<b>Visualizar Estadístico</b>																
<b>Descripción</b>	El sistema deberá mostrar un gráfico estadístico con la información de grados día desarrollo del último mes finalizado.																
<b>Secuencia</b>	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Se envía un a petición hacia el web Api de manera automática empleando filtros de la variable de sesión del navegador.</td> </tr> <tr> <td>2</td> <td>Se verifica la conexión y permisos de petición</td> </tr> <tr> <td>3</td> <td>Se accede al servicio requerido</td> </tr> <tr> <td>4</td> <td>El sistema valida los datos recibidos</td> </tr> <tr> <td>5</td> <td>El sistema responde con objetos tipo JSON tanto para días como para datos</td> </tr> <tr> <td>6</td> <td>Se cargan los datos al grafico estadístico</td> </tr> <tr> <td>7</td> <td>Se grafican los datos</td> </tr> </tbody> </table>	Paso	Acción	1	Se envía un a petición hacia el web Api de manera automática empleando filtros de la variable de sesión del navegador.	2	Se verifica la conexión y permisos de petición	3	Se accede al servicio requerido	4	El sistema valida los datos recibidos	5	El sistema responde con objetos tipo JSON tanto para días como para datos	6	Se cargan los datos al grafico estadístico	7	Se grafican los datos
Paso	Acción																
1	Se envía un a petición hacia el web Api de manera automática empleando filtros de la variable de sesión del navegador.																
2	Se verifica la conexión y permisos de petición																
3	Se accede al servicio requerido																
4	El sistema valida los datos recibidos																
5	El sistema responde con objetos tipo JSON tanto para días como para datos																
6	Se cargan los datos al grafico estadístico																
7	Se grafican los datos																
<b>Postcondición</b>	<p>Se muestra una alerta: "¡OK, Consulta Correcta!"</p> <p>Se puede navegar por la gráfica.</p>																
<b>Excepciones</b>	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td></td> <td>De no poder conectarse con los servicios:</td> </tr> <tr> <td>2</td> <td>E.1.- El sistema mostrará una alerta: "ERROR, No se encontraron los datos"</td> </tr> </tbody> </table>	Paso	Acción		De no poder conectarse con los servicios:	2	E.1.- El sistema mostrará una alerta: "ERROR, No se encontraron los datos"										
Paso	Acción																
	De no poder conectarse con los servicios:																
2	E.1.- El sistema mostrará una alerta: "ERROR, No se encontraron los datos"																

<b>CU-04</b>	<b>Visualizar Estadístico</b>
	E.2.- La grafica adoptara valores por defecto.
	E.3.- El caso de uso termina
	De no comprobar que los datos son correctos estén vacíos o tengan errores de tipeo.
4	E.1.- El sistema mostrará una alerta: "ERROR, No se encontraron los datos"
	E.2.- La grafica adoptara valores por defecto.
	E.3.- El caso de uso termina
<b>Comentarios</b>	El filtro debe ser llenado empleando el componente de fecha, para evitar problemas de tipeo.
<i>Nota.</i> La tabla presenta el desglosamiento del caso de uso entre el usuario y el visualizar estadístico.	
<b>Tabla 6</b>	
<i>Caso de uso Visualizar Resumen</i>	
<b>CU-05</b>	<b>Visualizar Resumen</b>
<b>Actor</b>	Usuario
<b>Descripción</b>	El sistema deberá llenar los campos de resumen con la información de las condiciones del último mes finalizado.

CU-05	Visualizar Resumen	
	<b>Paso</b>	<b>Acción</b>
	1	Se envía un a petición hacia el web Api de manera automática empleando filtros de la variable de sesión del navegador.
	2	Se verifica la conexión y permisos de petición
<b>Secuencia</b>	3	Se accede al servicio requerido
	4	El sistema valida los datos recibidos
	5	El sistema responde con un objeto tipo JSON con la información del resumen de condiciones.
	6	Se cargan los datos en los respectivos campos.
	Se muestra una alerta: "¡OK, Consulta Correcta!"	
<b>Postcondición</b>	Se muestran los datos.	
	<b>Paso</b>	<b>Acción</b>
		De no poder conectarse con los servicios:
<b>Excepciones</b>	2	E.1.- El sistema mostrará una alerta: "ERROR, No se encontraron los datos"
		E.2.- Los campos toman las condiciones por defecto.
		E.3.- El caso de uso termina

<b>CU-05</b>	<b>Visualizar Resumen</b>
	De no comprobar que los datos son correctos estén vacíos o tengan errores de tipeo.
4	E.1.- El sistema mostrará una alerta: "ERROR, No se encontraron los datos"
	E.2.- Los campos toman las condiciones por defecto.
	E.3.- El caso de uso termina
<b>Comentarios</b>	Los filtros deben ser llenados empleando el componente de fechas, para evitar problemas de tipeo.
<i>Nota.</i> La tabla presenta el desglosamiento del caso de uso entre el usuario y el visualizar resumen.	

**Tabla 7**

*Caso de uso navegar menú*

<b>CU-06</b>	<b>Navegar Menú</b>				
<b>Actor</b>	Administrador, Usuario				
<b>Descripción</b>	El sistema deberá permitir la selección de las diferentes opciones en la barra de menú además de dirigirnos a la opción seleccionada				
<b>Secuencia</b>	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Se pulsa sobre la sección deseada</td> </tr> </tbody> </table>	Paso	Acción	1	Se pulsa sobre la sección deseada
Paso	Acción				
1	Se pulsa sobre la sección deseada				

<b>CU-06</b>	<b>Navegar Menú</b>	
	2	Se selecciona la opción a usar
	3	Se redirecciona a la pantalla de la respectiva opción
<b>Postcondición</b>	Se cargan los filtros necesarios	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
<b>Comentarios</b>	NA	

*Nota. La tabla presenta el desglosamiento del caso de uso entre el usuario y el navegar menú*

**Tabla 8**

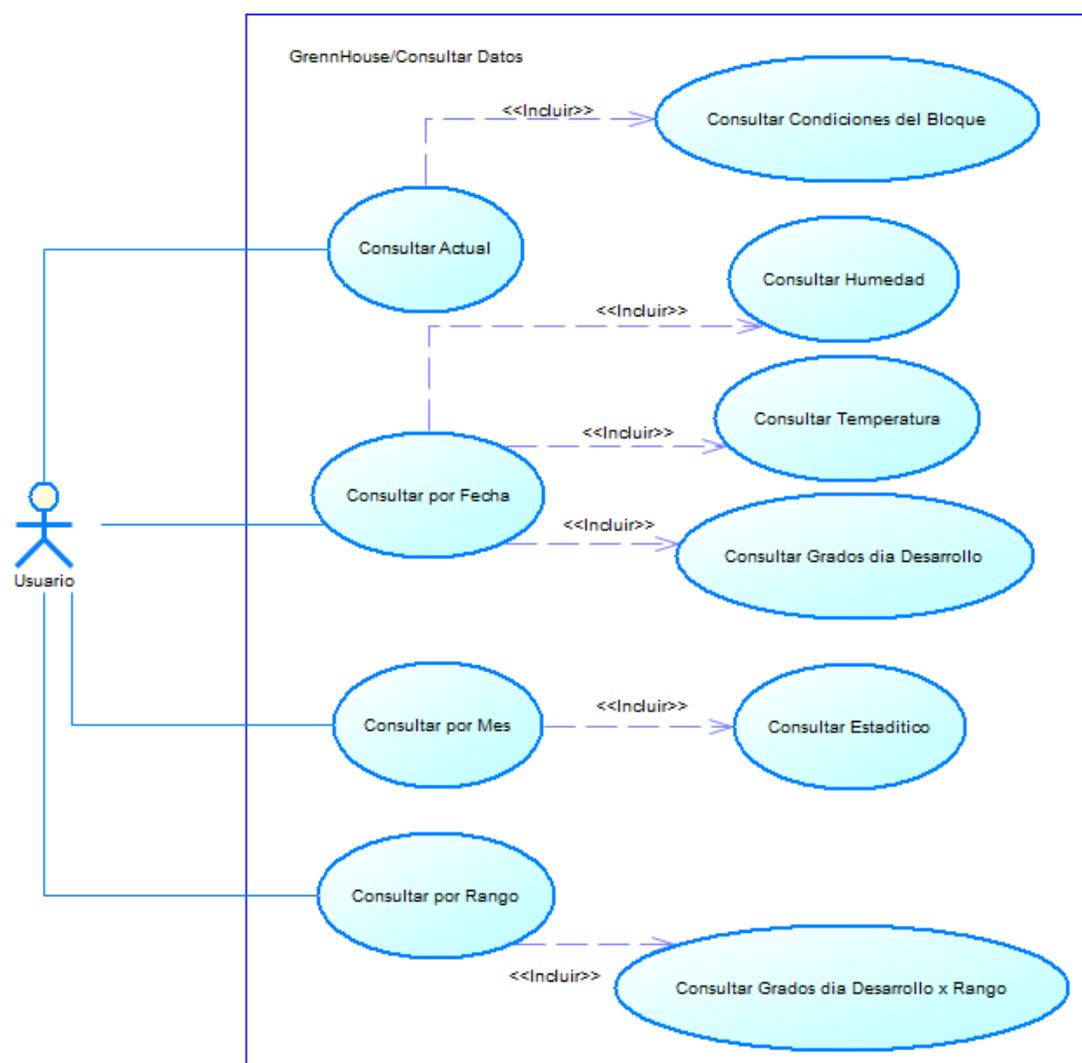
*Caso de uso salir del sistema*

<b>CU-07</b>	<b>Salir del Sistema</b>	
<b>Actor</b>	Administrador, Usuario	
<b>Descripción</b>	El sistema deberá mostrar la barra de acciones en la parte superior derecha.	
	<b>Paso</b>	<b>Acción</b>
<b>Secuencia</b>	1	Se dirige a la pestaña junto a la imagen del usuario (parte superior derecha)
	2	Se presiona el botón "Cerrar Sesión"
	3	Se destruyen todas las variables de sesión en el navegador
<b>Postcondición</b>	Se destruyen las variables de sesión	

<b>CU-07</b>	<b>Salir del Sistema</b>
	Se redirecciona a la pantalla de inicio de sesión
	<b>Paso    Acción</b>
<b>Excepciones</b>	NA
<b>Comentarios</b>	NA

*Nota.* La tabla presenta el desglosamiento del caso de uso entre el usuario y el salir del sistema.

**Figura 10**  
*Diagrama casos de uso nivel 2 consultar datos.*



*Nota.* La figura presenta el caso de uso entre el usuario y el consultar datos.

Tabla 9

*Caso de uso Consultar Condiciones del Bloque*

<b>CU-08</b>	<b>Consultar Condiciones del Bloque</b>	
<b>Actor</b>	Usuario	
<b>Descripción</b>	El sistema mostrara las condiciones actuales de un bloque predeterminado	
	<b>Paso</b>	<b>Acción</b>
	1	Se envía un a petición hacia el web Api de manera automática cada 2 segundos empleando filtros de la variable de sesión del navegador.
	2	Se verifica la conexión y permisos de petición
<b>Secuencia</b>	3	Se accede al servicio requerido
	4	El sistema valida los datos recibidos
	5	El sistema responde con un objeto tipo JSON con la información del resumen de condiciones.
	6	Se cargan los datos en los respectivos campos.
<b>Postcondición</b>	Se muestra las condiciones actuales de Altitud, Humedad y Temperatura	
	<b>Paso</b>	<b>Acción</b>
<b>Excepciones</b>		De no poder conectarse con los servicios:
	4	E.1.- El sistema mostrará una alerta de que no hay respuesta, se obtendrá un error 404

<b>CU-08</b>	<b>Consultar Condiciones del Bloque</b>
	E.2.- Los campos no se mostrarán
	E.3.- El caso de uso termina
	De no comprobar que los datos son correctos estén vacíos o tengan errores de tipeo.
6	E.1.- El sistema mostrara una alerta de que no se encontraron resultados, se devolverá un JSON con un mensaje de "ERROR"
	E.2.- Los campos no se mostrarán
	E.3.- El caso de uso termina

**Comentarios** Esta funcionalidad únicamente es visible en el dashboard

*Nota.* La tabla presenta el desglosamiento del caso de uso entre el usuario y el Consultar Condiciones del Bloque.

**Tabla 10**

*Caso de uso Consultar Humedad*

<b>CU-09</b>	<b>Consultar Humedad</b>
<b>Actor</b>	Usuario
<b>Descripción</b>	El sistema deberá mostrar la humedad máxima y mínima de un bloque en una determinada fecha

CU-09	Consultar Humedad	
	<b>Paso</b>	<b>Acción</b>
	1	Se selecciona en el menú la opción "Humedad"
	2	Se llena el filtro de búsqueda
	3	Se presiona el botón "Consultar"
	4	Se envía una petición hacia el web Api empleando filtros de la variable de sesión del navegador y el filtro de búsqueda.
<b>Secuencia</b>	5	Se verifica la conexión y permisos de petición
	6	Se accede al servicio requerido
	7	El sistema valida los datos recibidos
	8	El sistema responde un objeto tipo JSON con la información de las Humedades según el filtro enviado.
	9	Se cargan los datos en los respectivos campos
		Se muestra una alerta: "¡OK, Consulta Correcta!"
<b>Postcondición</b>		Se muestran los datos.
	<b>Paso</b>	<b>Acción</b>
		De no poder conectarse con los servicios:
<b>Excepciones</b>	5	E.1.- El sistema mostrará una alerta de que no hay respuesta, se obtendrá un error 404

CU-09	Consultar Humedad
	E.2.- Los campos toman las condiciones por defecto.
	E.3.- El caso de uso termina
	De no comprobar que los datos son correctos estén vacíos o tengan errores de tipeo.
7	E.1.- El sistema mostrara una alerta de que no se encontraron resultados, se devolverá un JSON con un mensaje de "ERROR"
	E.2.- Los campos toman las condiciones por defecto.
	E.3.- El caso de uso termina
<b>Comentarios</b>	El filtro debe ser llenado empleando el componente de fecha, para evitar problemas de tipeo.

*Nota.* La tabla presenta el desglosamiento del caso de uso entre el usuario y el Consultar Humedad.

**Tabla 11**

*Caso de uso Consultar Temperatura*

CU-10	Consultar Temperatura
<b>Actor</b>	Usuario

<b>CU-10</b>	<b>Consultar Temperatura</b>																				
<b>Descripción</b>	El sistema deberá mostrar la temperatura máxima y mínima de un bloque en una determinada fecha																				
<b>Secuencia</b>	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Se selecciona en el menú la opción "Temperatura"</td> </tr> <tr> <td>2</td> <td>Se llena el filtro de búsqueda</td> </tr> <tr> <td>3</td> <td>Se presiona el botón "Consultar"</td> </tr> <tr> <td>4</td> <td>Se envía un a petición hacia el web Api empleando filtros de la variable de sesión del navegador y el filtro de búsqueda.</td> </tr> <tr> <td>5</td> <td>Se verifica la conexión y permisos de petición</td> </tr> <tr> <td>6</td> <td>Se accede al servicio requerido</td> </tr> <tr> <td>7</td> <td>El sistema valida los datos recibidos</td> </tr> <tr> <td>8</td> <td>El sistema responde un objeto tipo JSON con la información de las Temperaturas según el filtro enviado.</td> </tr> <tr> <td>9</td> <td>Se cargan los datos en los respectivos campos</td> </tr> </tbody> </table>	Paso	Acción	1	Se selecciona en el menú la opción "Temperatura"	2	Se llena el filtro de búsqueda	3	Se presiona el botón "Consultar"	4	Se envía un a petición hacia el web Api empleando filtros de la variable de sesión del navegador y el filtro de búsqueda.	5	Se verifica la conexión y permisos de petición	6	Se accede al servicio requerido	7	El sistema valida los datos recibidos	8	El sistema responde un objeto tipo JSON con la información de las Temperaturas según el filtro enviado.	9	Se cargan los datos en los respectivos campos
Paso	Acción																				
1	Se selecciona en el menú la opción "Temperatura"																				
2	Se llena el filtro de búsqueda																				
3	Se presiona el botón "Consultar"																				
4	Se envía un a petición hacia el web Api empleando filtros de la variable de sesión del navegador y el filtro de búsqueda.																				
5	Se verifica la conexión y permisos de petición																				
6	Se accede al servicio requerido																				
7	El sistema valida los datos recibidos																				
8	El sistema responde un objeto tipo JSON con la información de las Temperaturas según el filtro enviado.																				
9	Se cargan los datos en los respectivos campos																				
<b>Postcondición</b>	<p>Se muestra una alerta: "¡OK, Consulta Correcta!"</p> <p>Se muestran los datos.</p>																				
<b>Excepciones</b>	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>5</td> <td>De no poder conectarse con los servicios:</td> </tr> </tbody> </table>	Paso	Acción	5	De no poder conectarse con los servicios:																
Paso	Acción																				
5	De no poder conectarse con los servicios:																				

CU-10	Consultar Temperatura
	E.1.- El sistema mostrará una alerta de que no hay respuesta, se obtendrá un error 404
	E.2.- Los campos toman las condiciones por defecto.
	E.3.- El caso de uso termina
	De no comprobar que los datos son correctos estén vacíos o tengan errores de tipeo.
7	E.1.- El sistema mostrara una alerta de que no se encontraron resultados, se devolverá un JSON con un mensaje de "ERROR"
	E.2.- Los campos toman las condiciones por defecto.
	E.3.- El caso de uso termina
<b>Comentarios</b>	El filtro debe ser llenado empleando el componente de fecha, para evitar problemas de tipeo.

*Nota.* La tabla presenta el desglosamiento del caso de uso entre el usuario y el Consultar Temperatura.

Tabla 12

*Caso de uso Consultar Grados Día Desarrollo*

<b>CU-11</b>	<b>Consultar Grados Día Desarrollo</b>																		
<b>Actor</b>	Usuario																		
<b>Descripción</b>	El sistema deberá mostrar información de temperatura y su promedio, también de los grados día desarrollo alcanzados en una determinada fecha además de todos los registros de Humedad y Temperatura																		
<b>Secuencia</b>	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Se selecciona en el menú la opción "Grados Día Desarrollo"</td> </tr> <tr> <td>2</td> <td>Se llena el filtro de búsqueda</td> </tr> <tr> <td>3</td> <td>Se presiona el botón "Consultar"</td> </tr> <tr> <td>4</td> <td>Se envía un a petición hacia el web Api empleando filtros de la variable de sesión del navegador y el filtro de búsqueda.</td> </tr> <tr> <td>5</td> <td>Se verifica la conexión y permisos de petición</td> </tr> <tr> <td>6</td> <td>Se accede al servicio requerido</td> </tr> <tr> <td>7</td> <td>El sistema valida los datos recibidos</td> </tr> <tr> <td>8</td> <td>El sistema responde con objetos tipo JSON con la información de las Temperaturas, Grados Día Desarrollo y un listado con el detalle de los registros según el filtro enviado.</td> </tr> </tbody> </table>	Paso	Acción	1	Se selecciona en el menú la opción "Grados Día Desarrollo"	2	Se llena el filtro de búsqueda	3	Se presiona el botón "Consultar"	4	Se envía un a petición hacia el web Api empleando filtros de la variable de sesión del navegador y el filtro de búsqueda.	5	Se verifica la conexión y permisos de petición	6	Se accede al servicio requerido	7	El sistema valida los datos recibidos	8	El sistema responde con objetos tipo JSON con la información de las Temperaturas, Grados Día Desarrollo y un listado con el detalle de los registros según el filtro enviado.
Paso	Acción																		
1	Se selecciona en el menú la opción "Grados Día Desarrollo"																		
2	Se llena el filtro de búsqueda																		
3	Se presiona el botón "Consultar"																		
4	Se envía un a petición hacia el web Api empleando filtros de la variable de sesión del navegador y el filtro de búsqueda.																		
5	Se verifica la conexión y permisos de petición																		
6	Se accede al servicio requerido																		
7	El sistema valida los datos recibidos																		
8	El sistema responde con objetos tipo JSON con la información de las Temperaturas, Grados Día Desarrollo y un listado con el detalle de los registros según el filtro enviado.																		

<b>CU-11</b>	<b>Consultar Grados Dia Desarrollo</b>
	9 Se cargan los datos en los respectivos campos
	Se muestra una alerta: "¡OK, Consulta Correcta!"
<b>Postcondición</b>	Se muestran los datos.
	<b>Paso Acción</b>
	De no poder conectarse con los servicios:
	E.1.- El sistema mostrará una alerta de que no hay respuesta, se obtendrá un error 404
5	E.2.- Los campos toman las condiciones por defecto.
	E.3.- El caso de uso termina
<b>Excepciones</b>	De no comprobar que los datos son correctos estén vacíos o tengan errores de tipeo.
	E.1.- El sistema mostrara una alerta de que no se encontraron resultados, se devolverá un JSON con un mensaje de "ERROR"
7	E.2.- Los campos toman las condiciones por defecto.
	E.3.- El caso de uso termina
<b>Comentarios</b>	El filtro debe ser llenado empleando el componente de fecha, para evitar problemas de tipeo.

*Nota.* La tabla presenta el desglosamiento del caso de uso entre el usuario y el Consultar Grados Día Desarrollo.

**Tabla 13**

*Caso de uso Consultar Estadístico*

<b>CU-12</b>	<b>Consultar Estadístico</b>																		
<b>Actor</b>	Usuario																		
<b>Descripción</b>	El sistema deberá mostrar un gráfico estadístico con la información de grados día desarrollo del último mes finalizado.																		
	<table border="1"> <thead> <tr> <th><b>Paso</b></th> <th><b>Acción</b></th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Se selecciona en el menú la opción "Estadístico"</td> </tr> <tr> <td>2</td> <td>Se llena el filtro de búsqueda</td> </tr> <tr> <td>3</td> <td>Se presiona el botón "Consultar"</td> </tr> <tr> <td><b>Secuencia</b></td> <td>Se envía un a petición hacia el web Api empleando filtros de la variable de sesión del navegador y los filtros de búsqueda.</td> </tr> <tr> <td>4</td> <td></td> </tr> <tr> <td>5</td> <td>Se verifica la conexión y permisos de petición</td> </tr> <tr> <td>6</td> <td>Se accede al servicio requerido</td> </tr> <tr> <td>7</td> <td>El sistema valida los datos recibidos</td> </tr> </tbody> </table>	<b>Paso</b>	<b>Acción</b>	1	Se selecciona en el menú la opción "Estadístico"	2	Se llena el filtro de búsqueda	3	Se presiona el botón "Consultar"	<b>Secuencia</b>	Se envía un a petición hacia el web Api empleando filtros de la variable de sesión del navegador y los filtros de búsqueda.	4		5	Se verifica la conexión y permisos de petición	6	Se accede al servicio requerido	7	El sistema valida los datos recibidos
<b>Paso</b>	<b>Acción</b>																		
1	Se selecciona en el menú la opción "Estadístico"																		
2	Se llena el filtro de búsqueda																		
3	Se presiona el botón "Consultar"																		
<b>Secuencia</b>	Se envía un a petición hacia el web Api empleando filtros de la variable de sesión del navegador y los filtros de búsqueda.																		
4																			
5	Se verifica la conexión y permisos de petición																		
6	Se accede al servicio requerido																		
7	El sistema valida los datos recibidos																		

<b>CU-12</b>	<b>Consultar Estadístico</b>
8	El sistema responde con objetos tipo JSON tanto para días como para datos
9	Se cargan los datos al grafico estadístico
<b>Postcondición</b>	Se muestra una alerta: "¡OK, Consulta Correcta!"
	Se puede navegar por la gráfica.
	<b>Paso    Acción</b>
	De no poder conectarse con los servicios:
5	E.1.- El sistema mostrará una alerta: "ERROR, No se encontraron los datos"
	E.2.- La grafica adoptara valores por defecto.
	E.3.- El caso de uso termina
<b>Excepciones</b>	De no comprobar que los datos son correctos estén vacíos o tengan errores de tipeo.
7	E.1.- El sistema mostrará una alerta: "ERROR, No se encontraron los datos"
	E.2.- La grafica adoptara valores por defecto.
	E.3.- El caso de uso termina

<b>CU-12</b>	<b>Consultar Estadístico</b>
<b>Comentarios</b>	El filtro debe ser llenado empleando el componente de fecha, para evitar problemas de tipeo.
<i>Nota.</i> La tabla presenta el desglosamiento del caso de uso entre el usuario y el Consultar Estadístico.	

**Tabla 14**

*Caso de uso Consultar Grados Día Desarrollo por Rango*

<b>CU-13</b>	<b>Consultar Grados Día Desarrollo por Rango</b>										
<b>Actor</b>	Usuario										
<b>Descripción</b>	El sistema deberá mostrar información de humedad y temperatura (máxima, mínima y promedio), también de los grados día desarrollo (total y promedio) alcanzados en un determinado periodo de tiempo además de todos los registros de Grado día desarrollo con su respectiva fecha.										
<b>Secuencia</b>	<table border="1"> <thead> <tr> <th>Paso</th> <th>Acción</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Se selecciona en el menú la opción "Grados Día Desarrollo X Rango"</td> </tr> <tr> <td>2</td> <td>Se llenan los filtros de búsqueda</td> </tr> <tr> <td>3</td> <td>Se presiona el botón "Consultar"</td> </tr> <tr> <td>4</td> <td>Se envía un a petición hacia el web Api empleando filtros de la variable de sesión del navegador y los filtros de búsqueda.</td> </tr> </tbody> </table>	Paso	Acción	1	Se selecciona en el menú la opción "Grados Día Desarrollo X Rango"	2	Se llenan los filtros de búsqueda	3	Se presiona el botón "Consultar"	4	Se envía un a petición hacia el web Api empleando filtros de la variable de sesión del navegador y los filtros de búsqueda.
Paso	Acción										
1	Se selecciona en el menú la opción "Grados Día Desarrollo X Rango"										
2	Se llenan los filtros de búsqueda										
3	Se presiona el botón "Consultar"										
4	Se envía un a petición hacia el web Api empleando filtros de la variable de sesión del navegador y los filtros de búsqueda.										

CU-13	Consultar Grados Dia Desarrollo por Rango	
5		Se verifica la conexión y permisos de petición
6		Se accede al servicio requerido
7		El sistema valida los datos recibidos
8		El sistema responde con objetos tipo JSON con la información de la Humedad, Temperatura, Grados Dia Desarrollo del periodo y un listado con el detalle de los registros grado día desarrollo ordenados por fecha según los filtros.
9		Se cargan los datos en los respectivos campos
Se muestra una alerta: "¡OK, Consulta Correcta!"		
<b>Postcondición</b>		
Se muestran los datos.		
<b>Paso    Acción</b>		
De no poder conectarse con los servicios:		
5		E.1.- El sistema mostrará una alerta de que no hay respuesta, se obtendrá un error 404
<b>Excepciones</b>		
		E.2.- Los campos toman las condiciones por defecto.
		E.3.- El caso de uso termina
7		De no comprobar que los datos son correctos estén vacíos o tengan errores de tipeo.

---

<b>CU-13</b>	<b>Consultar Grados Día Desarrollo por Rango</b>
--------------	--

---

	E.1.- El sistema mostrara una alerta de que no se encontraron resultados, se devolverá un JSON con un mensaje de "ERROR"
--	--

---

	E.2.- Los campos toman las condiciones por defecto.
--	---

---

	E.3.- El caso de uso termina
--	------------------------------

---

<b>Comentarios</b>	Los filtros deben ser llenados empleando los componentes de fecha, para evitar problemas de tipeo.
--------------------	--

---

*Nota.* La tabla presenta el desglosamiento del caso de uso entre el usuario y el Consultar Grados Día Desarrollo por Rango.

### 3.3.3. Diseño de la base de datos

Una vez depurados todas las entidades, así como los atributos necesarios empleamos la herramienta Power Designer en su versión 16.6, en la cual generamos todos los modelos que se mencionan a continuación; modelo conceptual de la base de datos Figura 11, modelo lógico de la base de datos Figura 12, modelo físico de la base de datos Figura 13.

El modelo final quedo conformado por 9 tablas las que se encuentran agrupadas de la siguiente manera:

- Sesión: Maneja el nivel de seguridad referente al acceso a la información de un determinado usuario.

Persona: Almacena datos relevantes de una persona natural

Usuario: Almacena datos referentes a credenciales del sistema

- Información de la Entidad o Empresa: Encargada de almacenar datos relevantes con respecto a una determinada entidad o empresa, numero de dispositivos y bloques donde se ubican.

Empresa: Almacena datos relevantes con respecto a una entidad o empresa.

Bloque: Almacena datos propios de cada bloque (ubicación, clima, extensión).

Dispositivo: Almacena datos informativos del dispositivo.

- Información de Producción: Encargada del almacenamiento de datos obtenidos por sensores pertenecientes a un determinado ciclo y producto. También almacena registros históricos asociados a un bloque.

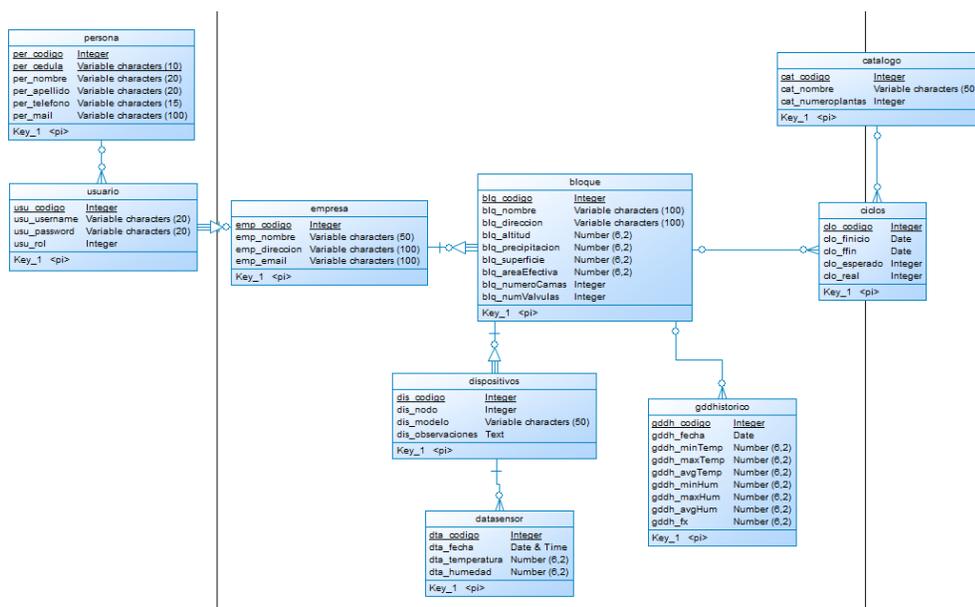
Catálogos: Almacena las diferentes variedades de productos disponibles.

Ciclos: Almacena datos referentes a un periodo de tiempo de producción.

Data sensor: Almacena todos los datos enviados por el sensor.

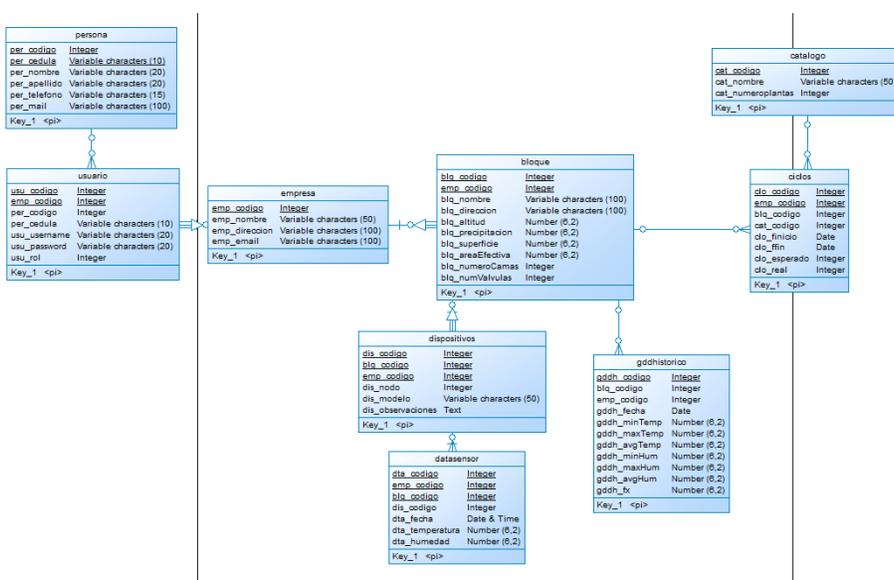
Gddhistorico: Almacena datos estadísticos de cada día (históricos).

**Figura 11**  
Modelo conceptual.



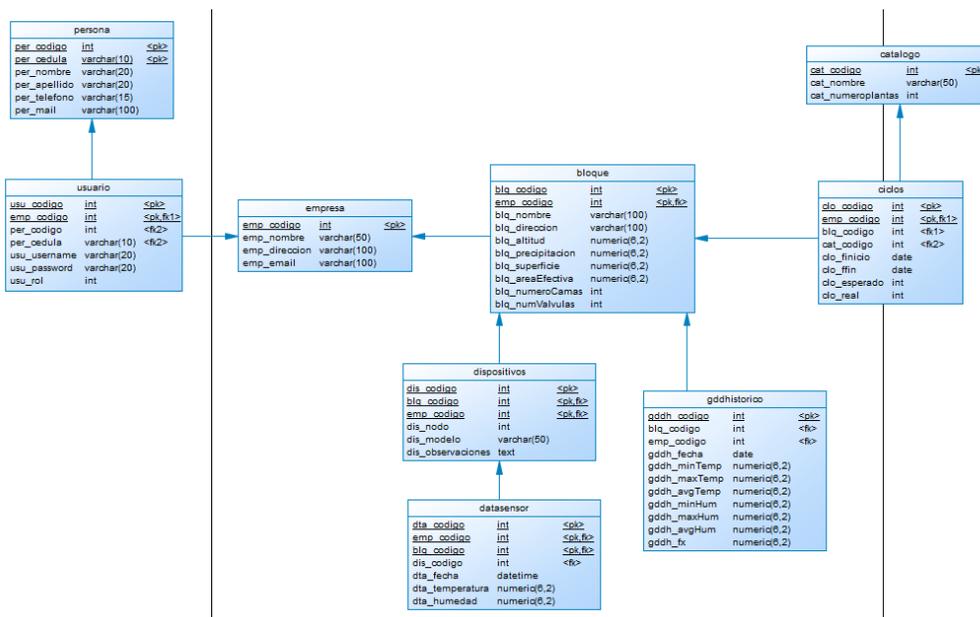
*Nota.* En esta imagen se visualiza modelo conceptual de la base de datos utilizada en la aplicación móvil y web.

**Figura 12**  
Modelo lógico.



*Nota.* En esta imagen se visualiza modelo lógico de la base de datos utilizada en la aplicación móvil y web.

**Figura 13**  
*Modelo físico.*



*Nota.* En esta imagen se visualiza modelo físico de la base de datos utilizada en la aplicación móvil y web.

### 3.3.4. Diseño del Api Rest

Definidas las entidades a usar y con base en estas se procedió a crear el proyecto de desarrollo en spring boot para lo cual se tomó en cuenta las posibles dependencias a usarse ya que al ser un Api rest este deberá manejar herramientas o paquetes los cuales nos permitan usar estas características.

Como primer punto se descargaron dependencias JDBC las cuales ayudaran a la comunicación entre el backend y la base de datos, también se agregaron paquetes de persistencia para poder mapear las entidades y poder trabajarlas como clases, se agregaron paquetes de Servicios Rest para emplear diferentes peticiones, así como permitir el uso de sus

protocolos y finalmente se agregaron paquetes CORS para permitir el intercambio de datos entre servidores de diferente origen.

### **3.4. Desarrollo del Sistema**

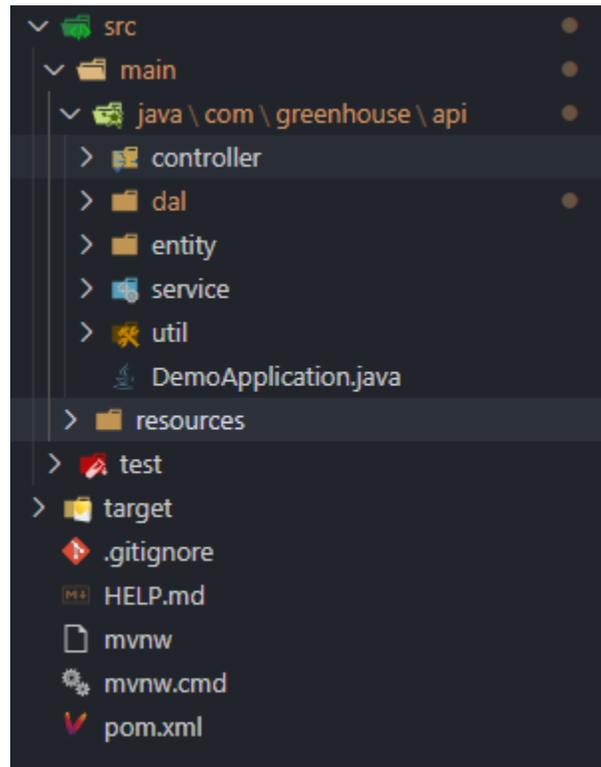
#### **3.4.1. Desarrollo del Backend**

Como punto de partida se configuró la aplicación para permitir la conexión con la base de datos, una vez comprobada la comunicación que tiene la base de datos con nuestro backend, se realizó el diseño de un prototipo de la estructura básica con una capa de entidades y una de acceso a datos “dal”. Posterior, se analizó los servicios que se expondrán y como se enviarán las respuestas de las peticiones hacia el frontend.

Se definieron las capas que se encargan de interactuar entre ellas manipulando la información recibida desde el frontend con el fin de obtener una respuesta a dicha petición.

Dentro de la estructura final de nuestro backend se cuenta con varias capas: de conexión a nuestra base de datos propia de Spring bot denominada “resources”, de entidades con el nombre “entity”, de acceso a datos “dal”, de servicios “service”, de controladores denominada “controller” y de utilidades denominada “útil”, todas ellas se muestran en la Figura 14 y se explican seguidamente.

**Figura 14**  
*Estructura BackEnd.*



*Nota.* En esta imagen se visualiza la estructura back end utilizada en la aplicación móvil y web.

#### 3.4.1.1. Resources

En esta capa se establece nuestra conexión con la base de datos, al ser spring boot un framework basado en java este utiliza complementos propios de dicho lenguaje para establecer la conexión, para nuestro caso puntual fue necesario implementar la librería jdbc correspondiente a MySQL y posterior configurarla como se muestra en la Figura 15.

**Figura 15**  
*Conexión a la base de datos desde BackEnd.*

```
server.port=5000
spring.jpa.show-sql=true
spring.datasource.url=jdbc:mysql://localhost:3306/gh?zeroDateTimeBehavior=convertToNull
spring.datasource.username=root
spring.datasource.password=SQLdosis622
```

*Nota.* En esta imagen se visualiza la configuración para la conexión a la base de datos desde el back end utilizada en la aplicación móvil y web.

### 3.4.1.2. Entity

Una vez agregada la dependencia de JPA en nuestro entorno como muestra la Figura 17, en esta capa se procedió al mapeo de todas las entidades mediante etiquetas que nos provee esta dependencia, lo que permite tratar las clases del proyecto como entidades de base de datos y convertir esto en instrucciones que interpretara el gestor de base de datos.

**Figura 16**  
*Dependencias JPA desde BackEnd.*

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-data-jpa</artifactId>
</dependency>
```

*Nota.* En esta imagen se visualiza las dependencias utilizadas desde el back end para la aplicación móvil y web.

Las clases se mapearon agregando las correspondientes etiquetas de acuerdo con las características propias de cada una detalladas en nuestra base de datos, para de esta manera evitar conflictos entre relación de tipo de datos o incompatibilidad entre ellos como se muestra en la Figura 17

**Figura 17**

*Mapeo de clases con la base de datos en el BackEnd.*

```

1  package com.greenhouse.api.entity;
2
3  import javax.persistence.Column;
4  import javax.persistence.Entity;
5  import javax.persistence.GeneratedValue;
6  import javax.persistence.GenerationType;
7  import javax.persistence.Id;
8  import javax.persistence.Table;
9
10 @Entity
11 @Table(name = "usuario")
12 public class Usuario {
13
14     @Id
15     @GeneratedValue(strategy = GenerationType.IDENTITY)
16     @Column(name = "usu_codigo")
17     private int usu_codigo;
18
19     @Column(name = "emp_codigo", nullable = false)
20     private int emp_codigo;
21
22     @Column(name = "per_codigo", nullable = true)
23     private int per_codigo;
24
25     @Column(name = "per_cedula", nullable = false)
26     private String per_cedula ;
27
28     @Column(name = "usu_username", nullable = false)
29     private String usu_username ;
30
31     @Column(name = "usu_password", nullable = false)
32     private String usu_password ;
33
34     @Column(name = "usu_rol", nullable = true)
35     private int usu_rol;

```

*Nota.* En esta imagen se visualiza las columnas de las tablas de la base de datos utilizadas desde el back end para la aplicación móvil y web.

Seguidamente se agregaron tanto constructores por defecto como constructor vacío y los correspondientes métodos “get” y “set” propios de cada clase como muestra la Figura 18

**Figura 18**  
Constructores en el BackEnd.

```

37 public Usuario() {
38 }
39
40 public Usuario(int usu_codigo, int emp_codigo, int per_codigo, String per_cedula, String usu_username,
41               String usu_password, int usu_rol) {
42     this.usu_codigo = usu_codigo;
43     this.emp_codigo = emp_codigo;
44     this.per_codigo = per_codigo;
45     this.per_cedula = per_cedula;
46     this.usu_username = usu_username;
47     this.usu_password = usu_password;
48     this.usu_rol = usu_rol;
49 }
50
51 public int getUsu_codigo() {
52     return usu_codigo;
53 }
54
55 public void setUsu_codigo(int usu_codigo) {
56     this.usu_codigo = usu_codigo;
57 }
58
59 public int getEmp_codigo() {
60     return emp_codigo;
61 }
62
63 public void setEmp_codigo(int emp_codigo) {
64     this.emp_codigo = emp_codigo;
65 }

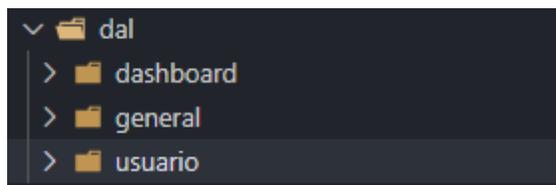
```

*Nota.* En esta imagen se visualiza el constructor y los métodos “Get” y “Set” utilizadas desde el back end para la aplicación móvil y web.

### 3.4.1.3. Dal

En esta capa se marcaron tres grupos principales, siendo estos: Usuario, Dashboard y General como los nuestra la Figura 19. Esto con la finalidad de mejorar la mantenibilidad del código y separar los diferentes accesos a datos.

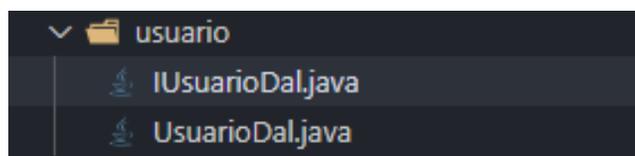
**Figura 19**  
Estructura en la capa Dal en el BackEnd



*Nota.* En esta imagen se visualiza la estructura de la capa “dal” desde el back end para la aplicación móvil y web.

Para la explicación del funcionamiento usaremos la sección de usuario, ya que esta configuración se replica en las demás secciones, únicamente variando los filtros de consulta o peticiones que se envían a la base de datos. La sección se encuentra compuesta por una interfaz y una clase como lo muestra la Figura 20.

**Figura 20**  
*Estructura de la sección usuario de la capa "dal"*



*Nota.* En esta imagen se visualiza la estructura de la capa "dal" desde el back end para la aplicación móvil y web.

En la interfaz declaramos nuestras funciones, así como las características de esta, como que nos devuelve de respuesta, de que tipo y que es lo que esta necesita para ejecutar todos sus procesos como lo muestra la Figura 21.

**Figura 21**  
*Interfaz de la sección usuario de la capa "dal"*

```
1 package com.greenhouse.api.dal.usuario;
2
3 import java.util.List;
4
5 import com.greenhouse.api.entity.*;
6 import com.greenhouse.api.util.CatalogoGenerico;
7 import com.greenhouse.api.util.Credencial;
8
9 public interface IUsuarioDal {
10
11     Credencial loggin(String user, String password) throws Exception;
12     List<CatalogoGenerico> geListaBloques(int cempresa) throws Exception;
13     Bloque getBloque(int cbloque) throws Exception;
14
15 }
```

*Nota.* En esta imagen se visualiza la interfaz del usuario en la capa "dal" desde el back end para la aplicación móvil y web.

Mientras que, en nuestra clase, previamente implementada la interfaz, procedemos al desarrollo de la función, la que cuenta con tres procesos principales mostrados en la Figura 22, los que son:

- El acceso a la base de datos mediante una variable sesión
- El llenado de la sentencia SQL a ejecutar
- Y la descomposición de la respuesta en un objeto personalizado

**Figura 22**  
*Clase de la sección usuario de la capa “dal”*

```

7  import javax.persistence.EntityManager;
8
9  import com.greenhouse.api.entity.*;
10 import com.greenhouse.api.util.CatalogoGenerico;
11 import com.greenhouse.api.util.Credencial;
12
13 import org.hibernate.Session;
14 import org.springframework.beans.factory.annotation.Autowired;
15 import org.springframework.stereotype.Repository;
16
17 @Repository
18 public class UsuarioDal implements IUserarioDal {
19
20     @Autowired
21     private EntityManager entityManager;
22
23     @Override
24     public Credencial loggin(String user, String password) throws Exception {
25         Session currentSession = null;
26         Credencial consulta = null;
27         try {
28             currentSession = entityManager.unwrap(Session.class);
29             String query = "SELECT us.usu_rol, us.usu_username, us.emp_codigo, emp.emp_nombre FROM usuario us \n"
30                 + "INNER JOIN empresa emp ON emp.emp_codigo = us.emp_codigo \n"
31                 + "WHERE us.usu_username = ? AND us.usu_password = ?";
32
33             Iterator iterador = currentSession.createQuery(query).setString(1,user).setString(2,password).list().iterator();
34
35             Object obj[] = null;
36             while (iterador.hasNext()) {
37                 consulta = new Credencial();
38                 obj = (Object[]) iterador.next();
39                 consulta.setRol(Integer.valueOf(obj[0].toString()));
40                 consulta.setUsername(String.valueOf(obj[1].toString()));
41                 consulta.setCempresa(Integer.valueOf(obj[2].toString()));
42                 consulta.setEmpresa(String.valueOf(obj[3].toString()));
43             }
44
45         } catch (Exception e) {
46             e.printStackTrace();
47         } finally {
48             currentSession.close();
49         }
50         return consulta;
51     }

```

**Nota.** En esta imagen se visualiza la clase del usuario en la capa “dal” desde el back end para la aplicación móvil y web.

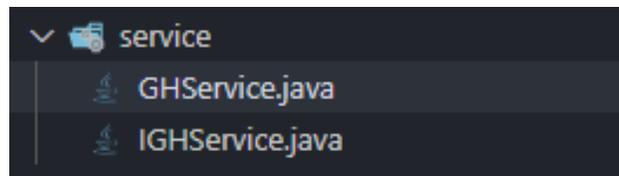
Los objetos en los que se descomponen los resultados de la consulta pertenecen a la capa de Utilidades la cual se explicara más adelante.

#### 3.4.1.4. Service

En esta capa se empleó únicamente una interfaz y una clase como se muestra en la Figura 23, ya que la carga de codificación no es extensa.

**Figura 23**

*Estructura de la capa service en el BackEnd*



*Nota.* En esta imagen se visualiza la estructura de la capa “service” desde el back end para la aplicación móvil y web.

Siguiendo los mismos lineamientos en nuestra interfaz declaramos todos los métodos y sus características como lo muestra la Figura 24. Esto teniendo en cuenta que aquí se encontrarán todas nuestras capas de acceso.

**Figura 24**  
*Interfaz de la capa servicio*

```

1 package com.greenhouse.api.service;
2
3 import java.util.List;
4
5 import com.greenhouse.api.entity.*;
6 import com.greenhouse.api.util.*;
7
8 public interface IGHService {
9
10     Credencial login(String user, String password) throws Exception;
11     List<CatalogoGenerico> getListaBloques(int cempresa) throws Exception;
12     Bloque getBloque(int cbloque) throws Exception;
13
14     /*-----DASHBOARD-----*/
15     List<String> getDiasDash(String fecha, int cempresa, int cbloque) throws Exception;
16     List<Double> getDataDash(String fecha, int cempresa, int cbloque) throws Exception;
17     List<MaxMin> getCondiciones(String fecha, int cempresa, int cbloque) throws Exception;
18
19     //public void save(DatosSensores data);
20     /*-----ACTUALES-----*/
21     DataActual getTemperatura(int cempresa, int cbloque) throws Exception;
22     DataActual getHumedad(int cempresa, int cbloque) throws Exception;
23
24     /*-----HISTORICOS-----*/
25     DataSensor getDatosbyId(int id);
26     List<MaxMin> getTemperaturaMaxMin(String fecha, int cempresa, int cbloque);
27     List<MaxMin> getHumedadMaxMin(String fecha, int cempresa, int cbloque);
28     List<GddByDate> getGDD(String fecha, int cempresa, int cbloque) throws Exception;
29     List<DataSensor> getGDDDetail(String fecha, int cempresa, int cbloque) throws Exception;
30     List<GddRank> getGDDRank(String fechaIn, String fechaFn, int cempresa, int cbloque) throws Exception;
31     MaxMin getTemperaturaRank(String fechaIn, String fechaFn, int cempresa, int cbloque) throws Exception;
32     MaxMin getHumedadRank(String fechaIn, String fechaFn, int cempresa, int cbloque) throws Exception;
33
34 }

```

*Nota.* En esta imagen se visualiza la interfaz de la capa “service” desde el back end para la aplicación móvil y web

Con respecto a la clase lo particular de esta es la inyección de dependencias como lo muestra la Figura 25, la cual se emplea con el propósito de evitar la instanciación de objetos cada vez que debamos acceder a un método y la que nos ayuda a reducir la carga en memoria al realizar una petición a cualquiera de nuestras capas de acceso.

**Figura 25**  
*Clase de la capa servicio*

```
1 package com.greenhouse.api.service;
2
3 import java.util.ArrayList;
4 import java.util.List;
5
6 import com.greenhouse.api.dal.dashboard.*;
7 import com.greenhouse.api.dal.general.*;
8 import com.greenhouse.api.dal.usuario.*;
9 import com.greenhouse.api.entity.*;
10 import com.greenhouse.api.util.*;
11
12 import org.springframework.beans.factory.annotation.Autowired;
13 import org.springframework.stereotype.Service;
14
15 @Service
16 public class GHService implements IGHService {
17
18     @Autowired
19     private IGeneralDal dal;
20
21     @Autowired
22     private IDashboardDal dalDash;
23
24     @Autowired
25     private IUserarioDal daluser;
26
```

*Nota.* En esta imagen se visualiza la clase de la capa “service” desde el back end para la aplicación móvil y web

Realizada la inyección de las dependencias estas pueden acceder a cada uno de sus métodos propios devolviéndonos los objetos personalizados que nosotros asignaremos como respuesta.

A continuación, se muestran el empleo de los métodos con respecto a su correspondiente dependencia inyectada en el código, método de acceso a Usuario Figura 26, método de acceso a General Figura 27 y método de acceso a Dashborad Figura 28.

**Figura 26**  
*Método de acceso a Usuario*

```
@Override
public Credencial loggin(String user, String password) throws Exception {
    Credencial data = new Credencial();
    try {
        data = daluser.loggin(user,password);
    } catch (Exception e) {
        e.printStackTrace();
    }
    return data;
}
```

*Nota.* En esta imagen se visualiza el método de acceso del usuario para la aplicación móvil y web.

**Figura 27**  
*Método de acceso General*

```
@Override
public List<GddByDate> getGDD(String fecha, int cempresa, int cbloque) throws Exception {
    List<GddByDate> lista = new ArrayList<>();
    try {
        lista = dal.getGDD(fecha,cempresa,cbloque);
    } catch (Exception e) {
        e.printStackTrace();
    }
    return lista;
}
```

*Nota.* En esta imagen se visualiza el método de acceso general para la aplicación móvil y web.

**Figura 28**  
*Método de acceso al Dashboard*

```
@Override
public List<Double> getDataDash(String fecha, int cempresa, int cbloque) throws Exception {
    List<Double> lista = new ArrayList<>();
    try {
        lista = dalDash.getDataDash(fecha,cempresa,cbloque);
    } catch (Exception e) {
        e.printStackTrace();
    }
    return lista;
}
```

*Nota.* En esta imagen muestra el método de acceso al Dashboard para la aplicación móvil y web.

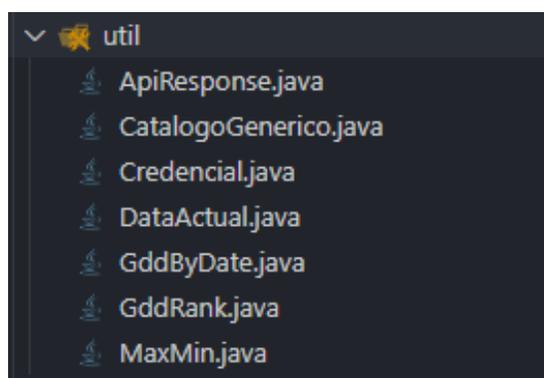
### 3.4.1.5. Util

La capa de utilidades “útil”, fue pensada con el fin de manejar objetos personalizados los que estarán compuestos por uno o varios atributos de nuestros objetos entidad, además cuentan con un atributo verificador, el cual nos permitirá conocer si la petición se ejecutó de manera correcta o no pudo ejecutarse, siendo este un punto de control propio y atributo validador para el frontend.

La capa de utilidades (útil) se conformó de siete clases como lo muestra la Figura 29.

**Figura 29**

*Capa util en el BackEnd*



*Nota.* En esta imagen muestra la estructura de la capa “util” en el backend para la aplicación móvil y web.

De este grupo de clases la que vuelve posible el manejo de un nivel de personalización en el objeto es “ApiResponse.java”, ya que en esta las diferentes clases de utilidades pasan a formar parte de un map o diccionario de objetos el que permite personalizar nuestras respuestas agregando elementos en dicho map o diccionario e identificándolos por un atributo propio el cual podemos definir.

La clase “ApiResponse.java” cuenta únicamente con dos atributos como lo muestra la Figura 30, siendo el primer atributo mensaje de tipo String, el que será usado como atributo verificador y un atributo contenido de tipo Map en el que se cargaran los resultados de una o varias consultas dependiendo el caso.

**Figura 30**

*Clase “ApiResponse.java” dentro de la capa util*

```
5 public class ApiResponse {
6     private String mensaje;
7     private Map<String, Object> contenido;
8
9     public ApiResponse() {
10    }
11
12    public String getMensaje() {
13        return mensaje;
14    }
15
16    public void setMensaje(String mensaje) {
17        this.mensaje = mensaje;
18    }
19
20    public Map<String, Object> getContenido() {
21        return contenido;
22    }
23
24    public void setContenido(Map<String, Object> contenido) {
25        this.contenido = contenido;
26    }
27
28 }
```

*Nota.* En esta imagen muestra la clase “ApiResponse.java” dentro de la capa “util” en el backend para la aplicación móvil y web.

Posteriormente esta clase se implementará en nuestro controlador donde se podrá mostrar su uso y profundizar en su importancia.

### 3.4.1.6. Controller

En esta capa se encuentran todos los métodos o servicios los cuales expondrá la aplicación, como primer paso se cargaron las dependencias necesarias como lo indica la Figura 31, para poder emplear tanto métodos de acceso como etiquetas las que definirán a la aplicación como tipo Rest.

**Figura 31**

*Dependencias en el Controlador de la capa util*

```
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.CrossOrigin;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;
```

*Nota.* En esta imagen muestra las dependencias dentro de la capa “util” en el backend para la aplicación móvil y web.

Seguidamente se debe asignar ciertas anotaciones como muestra la Figura 32. Las cuales denotan a nuestra clase como un controlador de tipo Rest y además asignan la ruta que se deberá seguir para acceder a los servicios.

**Figura 32**

*Etiquetas dentro del controlador de la capa util*

```
21 @RestController
22 @RequestMapping("/api")
```

*Nota.* En esta imagen muestra las etiquetas dentro de la capa “util” en el backend para la aplicación móvil y web.

Una vez ya adecuada nuestra clase con todas las dependencias a usar y con las etiquetas requeridas implementamos nuestros servicios creados por medio de una inyección de la interfaz “IGHService.java” como se muestra en la Figura 33, esto nos permitirá comunicarnos con nuestra capa de servicios llevando los filtros necesarios para obtener información de la base de datos.

**Figura 33**

*Implementación del servicio dentro del controlador de la capa útil*

```
26     @Autowired  
27     private IGHService service;
```

*Nota.* En esta imagen muestra la implementación del servicio dentro de la capa “util” en el backend para la aplicación móvil y web.

Contando con nuestros servicios implementados en el controlador se procede a declarar los métodos que pasarán a ser los servicios tipo rest de nuestra aplicación, teniendo en cuenta que estos van a ser accedidos desde varios dispositivos en la red se debe habilitar la visibilidad de estos mediante las políticas “CORS” las cuales se explicaran más adelante.

Procedemos al etiquetado de nuestro servicio como lo muestra la Figura 33, donde la etiqueta “@CrossOrigin” hace referencia al intercambio de datos con nuestro frontend que se encuentra en ejecución en otro servidor y la etiqueta “@PostMapping” que cumple dos funciones, la primera de señalarnos que este es un método de acceso tipo POST y también setear la ruta de acceso al mismo. Con el etiquetado de nuestro método realizado de la manera correcta se procede a declarar el método y lo que este devolverá, así como sus parámetros de entrada.

**Figura 34**

*Etiquetado del servicio en el controlador de la capa util*

```
@CrossOrigin(origins = "http://localhost:4200")  
@PostMapping("/login")  
public ResponseEntity<ApiResponse> login(@RequestBody Map<String, String> json)
```

*Nota.* En esta imagen muestra el etiquetado del servicio dentro de la capa “util” en el backend para la aplicación móvil y web.

Todos nuestros servicios son de tipo POST ya que en al usar este método los datos no son visibles al usuario en la web, todos los servicios devuelven un `ResponseEntity` a manera de lista el cual hace referencia a nuestra clase personalizada del paquete de utilidades “`ApiResponse.java`”, la cual se detalla en el capítulo 3.4.1.5. Util, además dicho servicios reciben parámetros de tipo JSON los cuales posteriormente son descompuestos para extraer los filtros.

A continuación, se mostrará un servicio desarrollado y las partes importantes del mismo como lo muestra la Figura 35, ya que todos los servicios siguen estas directrices, variando únicamente su respuesta.

**Figura 35**

*Desarrollo del servicio login dentro de la capa util*

```

@CrossOrigin(origins = "http://localhost:4200")
@PostMapping("/login")
public ResponseEntity<ApiResponse> login(@RequestBody Map<String, String> json) {
    ApiResponse apiresp = new ApiResponse();
    Map<String, Object> map = new HashMap<>();
    try {
        Credencial data = service.login(json.get("username"), json.get("password"));
        if (data != null) {
            apiresp.setMensaje("OK");
            map.put("Usuario", data);
            apiresp.setContenido(map);
            ResponseEntity.status(HttpStatus.OK).body(apiresp);
            return ResponseEntity.status(HttpStatus.OK).body(apiresp);
        } else {
            apiresp.setMensaje("ERROR");
            map.put("Usuario", null);
            apiresp.setContenido(map);
            ResponseEntity.status(HttpStatus.OK).body(apiresp);
            return ResponseEntity.status(HttpStatus.OK).body(apiresp);
        }
    } catch (Exception e) {
        e.printStackTrace();
        ResponseEntity.status(HttpStatus.NOT_FOUND).body(apiresp);
        return ResponseEntity.status(HttpStatus.NOT_FOUND).body(apiresp);
    }
}

```

*Nota.* En esta imagen muestra el código utilizado del servicio login dentro de la capa “util” en el backend para la aplicación móvil y web.

Los servicios cuentan con un bloque Try-catch el mismo que dependiendo los filtros de consulta o la disponibilidad de información nos devolverán un objeto de tipo JSON con nuestra información.

En el bloque “try” comprobamos el acceso a nuestros servicios internos llevándonos los filtros extraídos del cuerpo de nuestro objeto JSON y almacenando la respuesta en un objeto personalizado, para este caso el objeto Credencial. Seguidamente comprobamos si este tuvo o

no un resultado valido de ser la respuesta afirmativa se procede a llenar nuestro objeto personalizado con el mensaje verificador "OK" y setear un objeto de tipo map con el o los respectivos resultados dependiendo el caso, este pasara a ser parte de nuestro objeto personalizado "ApiResponse.java" y se setearan las características de un response web, cargando nuestro objeto personalizado al body y su estado como "HttpStatus.OK". Caso contrario se llenará nuestro objeto personalizado con el mensaje verificador "ERROR" y su map con null, seguido se setearan estas características al body y su estado como "HttpStatus.OK" lo que nos indicará que no hay información disponible para esos parámetros.

El bloque catch se ejecutará únicamente cuando haya problemas con nuestros filtros o con nuestros servicios es decir se enviaron menos o más filtros de los requeridos o no se está disponible el servicio solicitado, de ser alguno de estos el caso se devolverá el body del objeto JSON vacío y con un estado "HttpStatus.NOT\_FOUND". La lista completa de servicios se detallará en la siguiente sección.

### **3.4.2. Servicios Rest**

#### **3.4.2.1. Exposición de Servicios**

Al tratarse de un RestApi el cual se encontrará alojado en un servidor, los servicios de este deben mostrarse en la red, como se mencionó en las secciones anteriores además estos comparten el mismo número de parámetros de entrada, la misma estructura y el mismo tipo de objetos JSON de respuesta.

Seguidamente profundizaremos más en los temas relacionados a como se hace posibles exponer estos servicios en la red y mostrar las características de los servicios desarrollados.

### 3.4.2.2. CORS

Es una medida de seguridad la que evita que usuarios puedan cargar contenido malicioso o accedan a determinados métodos dentro del servidor web, algunas políticas de seguridad mencionan que los datos deben provenir de la misma fuente, es decir deben estar en el mismo servidor, dicho esto CORS es una política la cual permite o rechaza solicitudes de servidores externos dependiendo la configuración que se asigne.

El servidor o aplicación donde se configure CORS podrá permitir el intercambio de recursos con otros servidores una vez estos se encuentren previamente identificados y configurados en la aplicación, caso contrario este devolverá un error 404, negando totalmente el acceso a la aplicación y sus servicios.

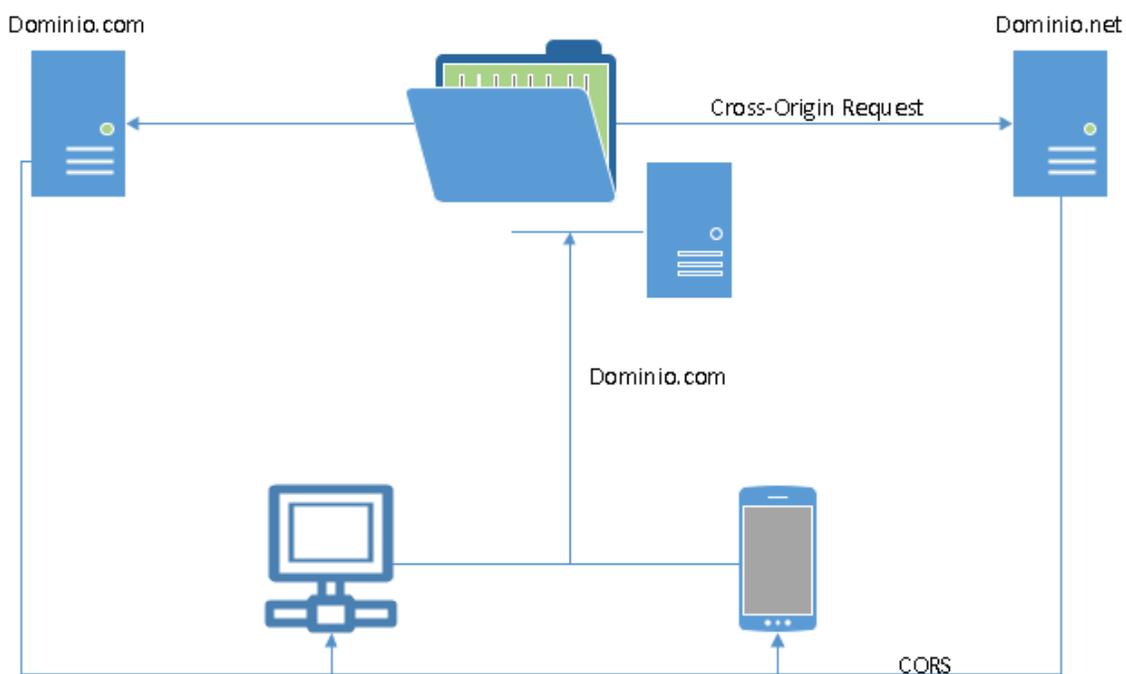
Existen varios tipos de cabeceras o CORS headers todas con diferentes características. A continuación, se muestra una lista de estas:

- Access-Control-Allow-Origin: ¿qué origen está permitido?
- Access-Control-Allow-Credentials: ¿también se aceptan solicitudes cuando el modo de credenciales es incluir (include)?
- Access-Control-Allow-Headers: ¿qué cabeceras pueden utilizarse?
- Access-Control-Allow-Methods: ¿qué métodos de petición HTTP están permitidos?
- Access-Control-Expose-Headers: ¿qué cabeceras pueden mostrarse?
- Access-Control-Max-Age: ¿cuándo pierde su validez la solicitud preflight?
- Access-Control-Request-Headers: ¿qué header HTTP se indica en la solicitud preflight?

- Access-Control-Request-Method: ¿qué método de petición HTTP se indica en la solicitud preflight?
- Origin: ¿de qué origen proviene la solicitud?

El primer header es especialmente importante, ya que especifica desde qué otro host se puede acceder al servidor solicitado. Además de una dirección concreta, en dicha cabecera también se puede incluir un comodín en forma de asterisco. De esta manera, el servidor permitirá cross-origin requests de cualquier origen.

**Figura 36**  
*Diagrama CORS.*



*Nota.* En esta imagen se visualiza el funcionamiento de CORS.

### 3.4.2.3. Tabla de Servicios

**Tabla 15**  
*Tabla de Rutas*

Host	Ruta	Método	Parámetros	Respuesta	Descripción
http://localhost:5000/api	/login				Retorna un objeto JSON el cual contiene la información de un determinado usuario y un mensaje verificador
	/listbloques				Retorna un objeto JSON con los bloques registrados correspondientes a una determinada empresa y un mensaje verificador
	/bloque	POST	JSON	JSON	Retorna un objeto JSON la información detallada de un determinado bloque y un mensaje verificador
	/temperatura				Retorna un objeto tipo JSON con la temperatura actual de un bloque y un mensaje verificador
	/humedad				Retorna un objeto tipo JSON con la humedad actual de un bloque y un mensaje verificador

Host	Ruta	Método	Parámetros	Respuesta	Descripción
	/listemperatura				Retorna un objeto tipo JSON con un listado de temperaturas de un bloque predeterminado en un día y un mensaje verificador
	/listhumedad				Retorna un objeto tipo JSON con un listado de humedades de un bloque predeterminado en día y un mensaje verificador
	/gddc				Retorna un objeto tipo JSON con los datos sobre fecha, máximo, mínimo y promedio de temperatura en ese día y el factor de crecimiento (gdd/grado día desarrollo) y un mensaje verificador
	/gddd				Retorna un objeto tipo JSON con un listado de temperaturas y humedades de un bloque predeterminado en día y un mensaje verificador
	/gddrank				Retorna un objeto tipo JSON con el total y promedio de gdd de un bloque predeterminado en un periodo de tiempo además de un listado del promedio de cada gdd por día registrado y un mensaje verificador

Host	Ruta	Método	Parámetros	Respuesta	Descripción
	/resumenTemp				Retorna un objeto tipo JSON con el mínimo, máximo y promedio de temperaturas de un bloque predeterminado en un periodo de tiempo y un mensaje verificador
	/resumenHum				Retorna un objeto tipo JSON con el mínimo, máximo y promedio de humedades de un bloque predeterminado en un periodo de tiempo y un mensaje verificador
	/diasDS				Retorna un objeto tipo JSON con los días registrados por los sensores de un mes predeterminado en un bloque predeterminado con un formato día-Mes (1-Mar) y un mensaje verificador
	/dataDS				Retorna un objeto tipo JSON con los datos de cada gdd promedio registrados por los sensores de un mes predeterminado en un bloque predeterminado y un mensaje verificador

---

Host	Ruta	Método	Parámetros	Respuesta	Descripción
	/condicionesDS				Retorna un objeto tipo JSON con los datos de máximo, mínimo y promedio tanto de temperaturas como humedades de un mes predeterminado en un bloque predeterminado y un mensaje verificador.

---

*Nota.* Esta tabla indica las rutas de acceso que tiene la aplicación móvil y web Green House.

### **3.4.3. Desarrollo del FrontEnd**

El frontend de la aplicación como se ha mencionado a lo largo del documento se encuentra segmentado ya que este cuenta con su versión web y la aplicación móvil los cuales son independientes entre sí, pero comparten el mismo backend.

Teniendo esto en cuenta las funcionalidades que posee cada uno son las mismas, con la diferencia que determinados componentes se comportan de mejor manera en la web por motivo de procesamiento y presentación. Sin embargo, todas las funcionalidades se pueden llevar a cabo desde su versión web como su versión móvil. Como parte importante en el diseño de la versión móvil se limitaron los procesos los cuales devuelven información en tiempo real, ya que esto al ser una petición constante de información por parte del dispositivo móvil a nuestro sistema, el consumo de datos es constante y muy amplio, por lo cual se tomó en cuenta solo una parte de todas estas funciones.

Para el desarrollo del frontend se tomó en cuenta los procesos principales que definió el usuario los cuales se encuentran en la sección 3.2.2. Diagramas y Casos de Uso y sus filtros o parámetros de entrada necesarios para estos, varios se encuentran conformados por más de un componente y otros están especializados en una determinada funcionalidad esto se aplicó tanto a las versiones web como móvil y se detallaran a continuación.

#### **3.4.3.1. Web**

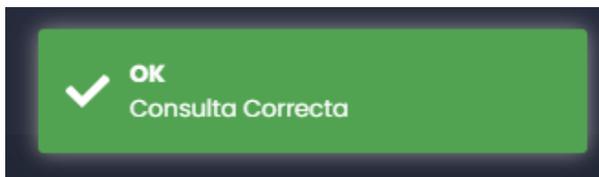
##### **3.4.3.1.1. Notificaciones**

El frontend cuenta con notificaciones las cuales nos indican si una petición web o un proceso interno se ejecutaron de manera correcta, estas aparecerán en la parte superior derecha

de nuestra pantalla y se sirven de nuestro objeto personalizado, en concreto de nuestro mensaje validador para ser ejecutadas. Lo que resume nuestras notificaciones a dos opciones “Ok” y “ERROR” las que se muestran en la Figura 37 y Figura 38 respectivamente.

**Figura 37**

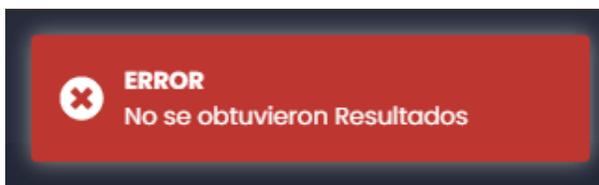
*Notificación correcta*



*Nota.* Imagen que muestra la notificación de una consulta correcta.

**Figura 38**

*Notificación incorrecta*

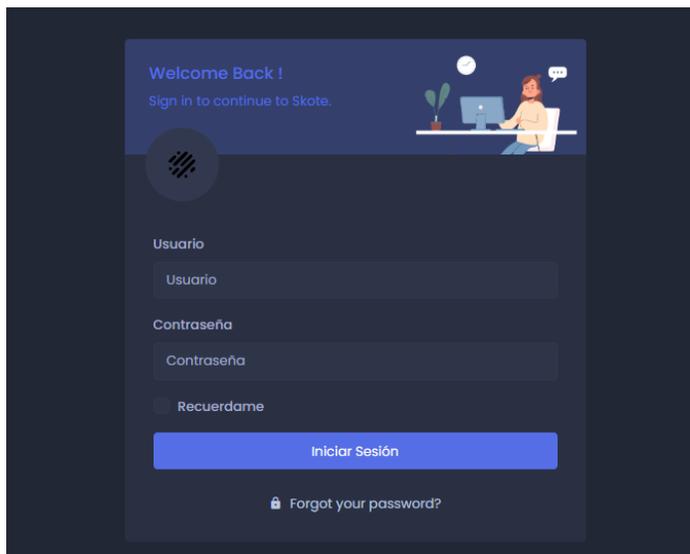


*Nota.* Imagen que muestra la notificación de una consulta incorrecta.

#### **3.4.3.1.2. Inicio de Sesión**

La pantalla de Inicio de Sesión cuenta con un formulario como muestra la Figura 39, el cual recupera los datos que ingresa el usuario y un botón de “Iniciar Sesión”, el mismo que después de ser presionado emiten un evento el cual realiza la petición web al backend llevando los datos ingresados. Además, esta pantalla cuenta con un diseño en la parte superior referente a una estación de trabajo.

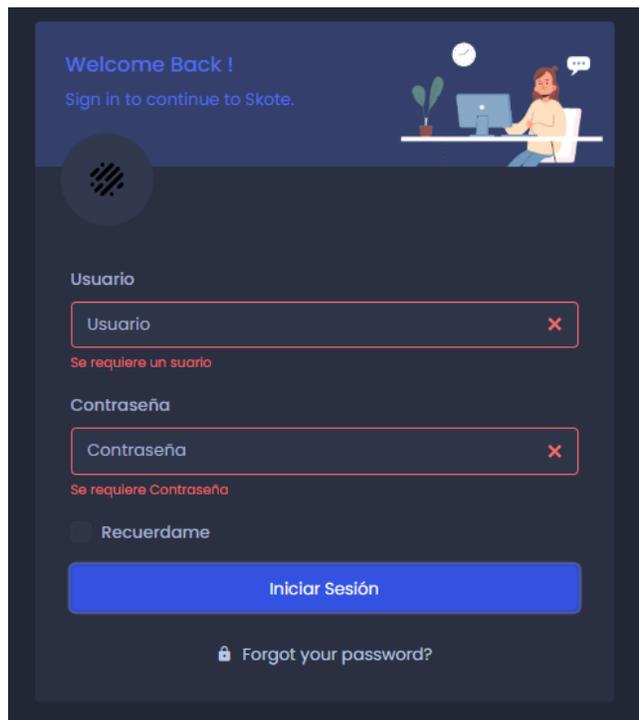
**Figura 39**  
*Interfaz Login*



*Nota.* Imagen que muestra la interfaz del login para acceder a la aplicación web.

En caso de que el formulario no cuente con todos, los parámetros solicitados o se haya cometido un error de tipeo los o el campo se marca como muestra la siguiente Figura 40.

**Figura 40**  
*Parámetros obligatorios en el Login*

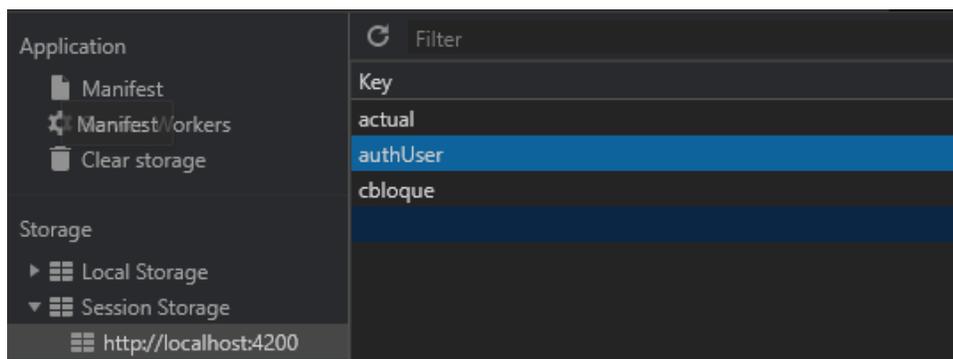


The image shows a login interface for 'Skote'. At the top, it says 'Welcome Back!' and 'Sign in to continue to Skote.' Below this is a dark blue header with a person working at a computer. The main form area is dark blue and contains two input fields: 'Usuario' and 'Contraseña'. Both fields have a red border and a red 'X' icon, indicating an error. Below the 'Usuario' field, the text 'Se requiere un suario' is displayed in red. Below the 'Contraseña' field, the text 'Se requiere Contraseña' is displayed in red. There is a 'Recuerdame' checkbox and a blue 'Iniciar Sesión' button. At the bottom, there is a link 'Forgot your password?' with a lock icon.

*Nota.* Imagen que muestra la interfaz cuando existe un error del acceso en el login para acceder a la aplicación web.

Finalizado el proceso de Inicio de Sesión de manera correcta, se almacenarán las credenciales del usuario en el Session Storage de nuestro navegador como se muestra en la Figura 41, las que emplearemos como filtros de búsqueda a lo largo de toda la sesión y ejecución de funcionalidades presentadas en el sistema.

**Figura 41**  
*Session Store*



*Nota.* Imagen que muestra la variable de sesión cuando se inicia la aplicación web.

### 3.4.3.1.3. Menú y Navegación

Posterior al inicio de sesión el sistema redirecciona a la página principal donde se visualiza nuestro componente Dashboard el cual se tratará en la siguiente sección.

En la parte superior podemos encontrar la barra de navegación la cual está fijada y es visible en todas las pantallas disponibles, como lo muestra la Figura 42, la misma cuenta con un botón de mostrar/ocultar barra de menú que se visualiza en toda la parte izquierda de nuestra pantalla y cuenta con dos secciones las cuales contendrán todas las funcionalidades del sistema. También cuenta con un botón para desplegar el contenido en modo pantalla completa y una pestaña la cual nos permitirá cerrar sesión que posee la imagen del usuario.

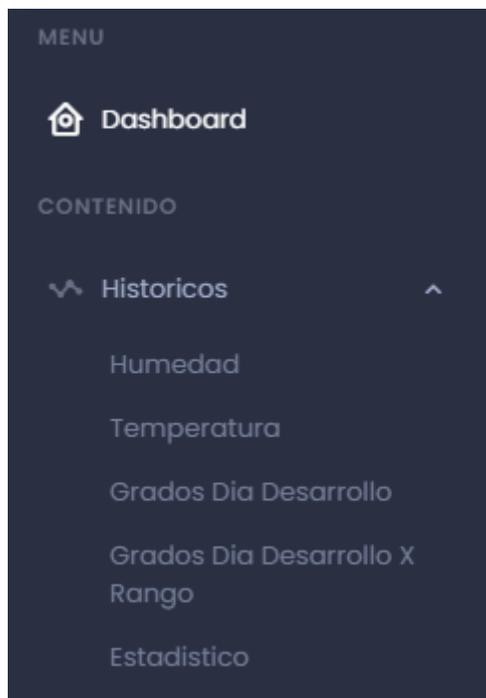
**Figura 42**  
*Barra de navegación*



*Nota.* Imagen que muestra la barra de navegación cuando se inicia la aplicación web.

El menú se encuentra separado en dos secciones como muestra la Figura 43, la primera mostrando la información de bienvenida y datos requeridos por el usuario denominada “Dashboard” y la segunda “Históricos” la que permite la consulta de información de acuerdo con los parámetros establecidos.

**Figura 43**  
*Menú de navegabilidad*



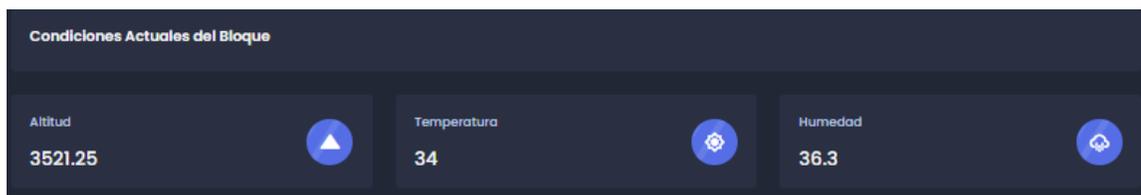
*Nota.* Imagen que muestra el menú de navegabilidad cuando se inicia la aplicación web.

#### **3.4.3.1.4. Dashboard**

El Dashboard está compuesto por varias secciones o componentes los cuales trabajan de manera independiente y cada uno responde a una petición web única. Dentro de las secciones con las que cuenta esta pantalla están las condiciones actuales del bloque las mismas que como su nombre lo dice nos muestran las condiciones de un determinado bloque como se observa en la Figura 44.

**Figura 44**

*Condiciones actuales del bloque dentro de la pestaña Dashboard*

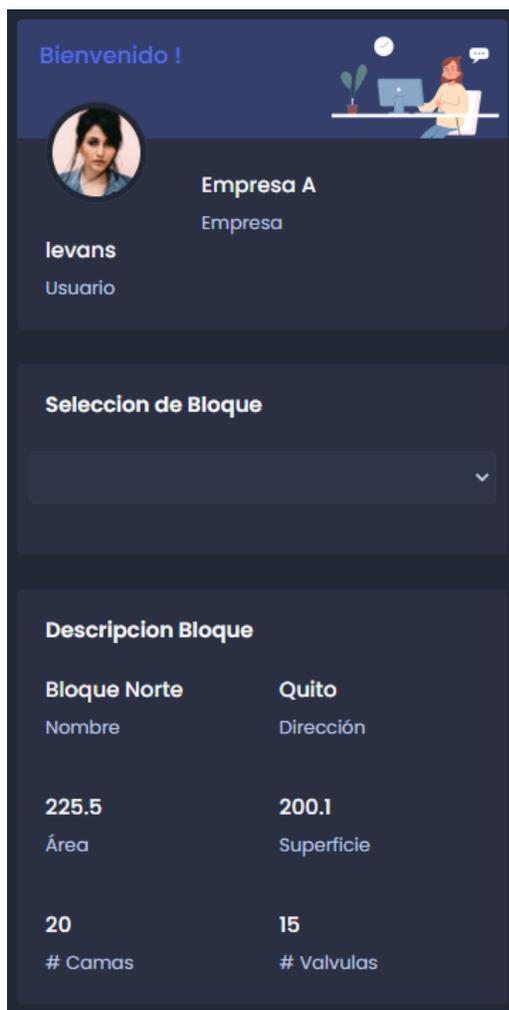


*Nota.* Imagen que muestra las condiciones actuales en la aplicación web.

Otra sección importante es la de información de usuario e información del bloque actual como se muestra en la Figura 45 y la Figura 46 respectivamente, esto nos permite saber en qué bloque nos encontramos, además se configurará este bloque como un filtro en nuestra variable de sesión en el Session Storage del Navegador como se observa en la Figura 47, para poder efectuar peticiones cada que este bloque cambie.

**Figura 45**

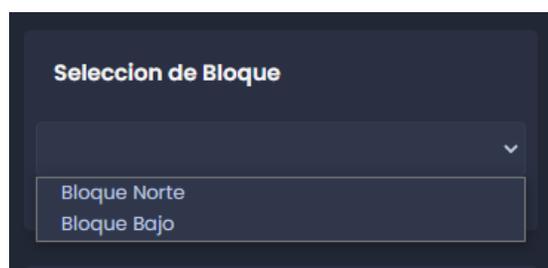
Información del bloque actual dentro de la pestaña Dashboard



*Nota.* Imagen que muestra la información del bloque actual en la aplicación web.

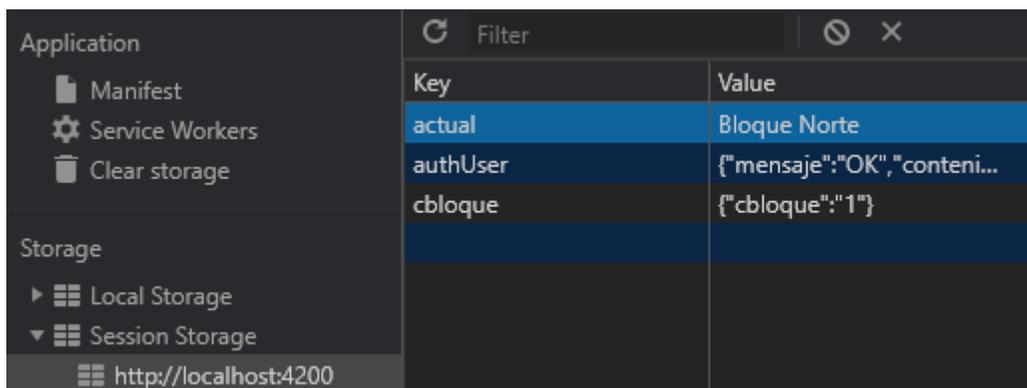
**Figura 46**

Selección del bloque



*Nota.* Imagen que muestra todos los bloques existentes en la aplicación web.

**Figura 47**  
Cambio de valores en Session Storage



*Nota.* Imagen que muestra el cambio de la variable de sesión en la aplicación web.

El dashboard también cuenta con una gráfica estadística como muestra la Figura 48, la cual muestra los registros obtenidos del último mes finalizado, estos se componen en los ejes “X” y “Y”, para días y cantidades respectivamente, una vez el componente dashboard es iniciado es decir su pantalla es seleccionada esta grafica automáticamente realiza la consulta tomando la fecha del sistema como parámetro necesario para poder obtener los datos.

De ser necesario este cuenta con un filtro el cual permitirá buscar datos no solo del mes otorgado por el sistema sino de un determinado mes que el usuario elija siempre y cuando este mes ya haya culminado.

**Figura 48**

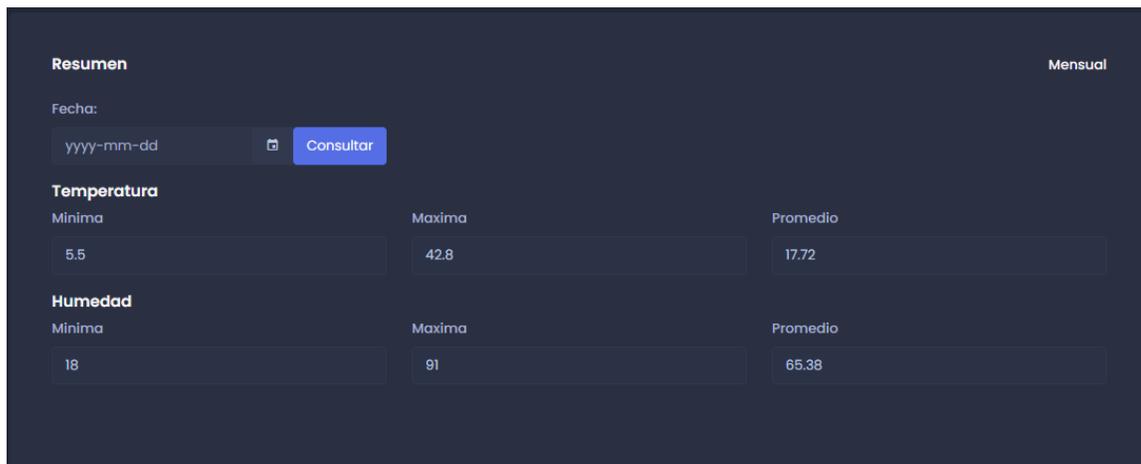
Gráfico estadístico con registros del último mes



*Nota.* Imagen que muestra una gráfica del último mes de los grados días desarrollo en la aplicación web.

Por último, tenemos el componente de Resumen que al igual que la gráfica estadística realiza una consulta automática de los datos y temperaturas del mes, y al igual que el componente anterior cuenta con un filtro el cual posee las mismas características antes mencionadas.

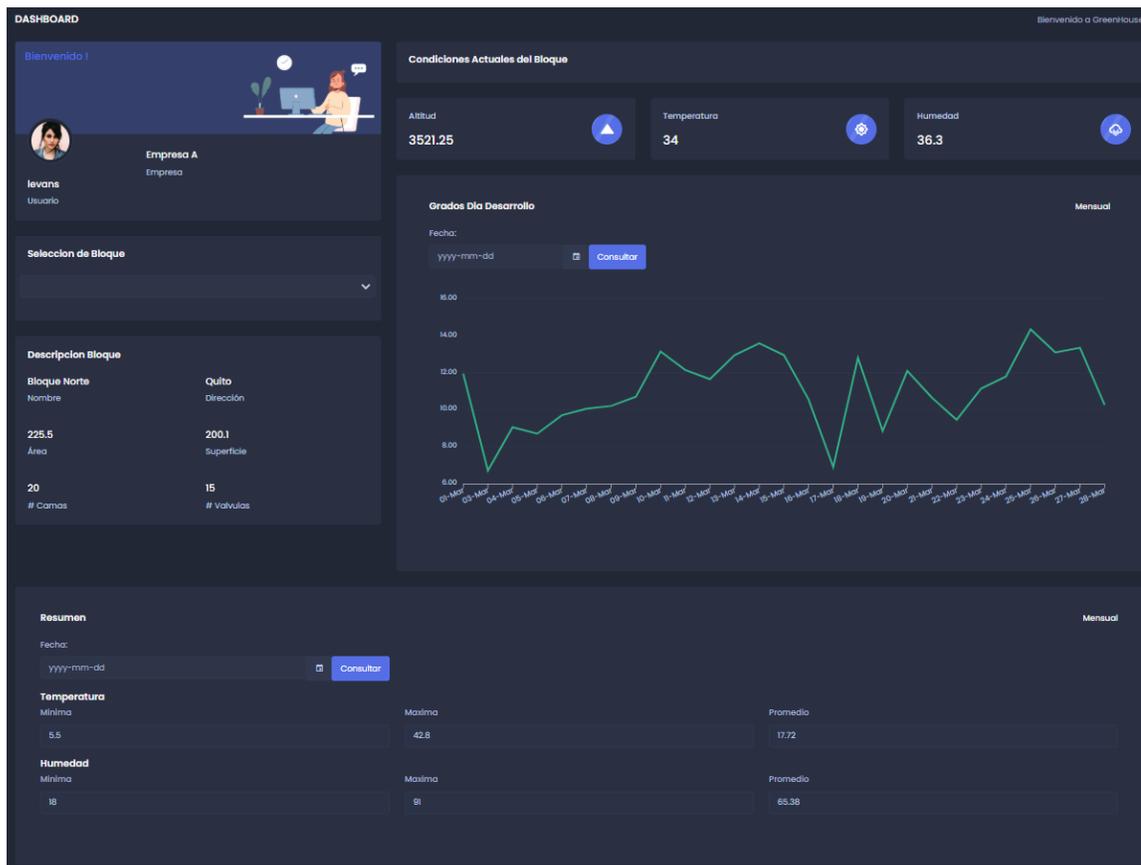
**Figura 49**  
*Resumen mensual*



*Nota.* Imagen que muestra el resumen mensual en la aplicación web.

En la Figura 50 podemos observar la pantalla de nuestro Dashboard de manera completa, así como la distribución de cada uno de los componentes antes mencionados.

**Figura 50**  
Pantalla completa del dashboard



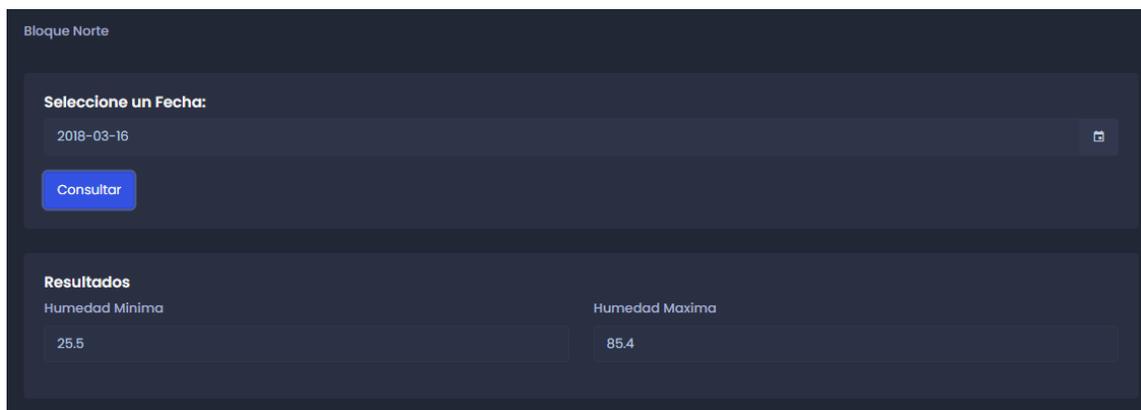
*Nota.* Imagen que muestra la pantalla completa del dashboard en la aplicación web.

### 3.4.3.1.5. Históricos

Las diferentes opciones de la sección "Históricos" serán explicados de manera general ya que en el capítulo V se mostrará el funcionamiento de cada una de las pantallas de manera individual. Dicho esto, las opciones de esta sección cuentan con filtros de consulta similares siendo estos una única fecha o un periodo, mediante los cuales mostraremos información relacionada a Humedad, Temperatura, Grados días Desarrollo, Grados día Desarrollo por Rango y Estadístico.

Las opciones de Humedad y Temperatura comparten la misma presentación ya que estas poseen únicamente un filtro de búsqueda tipo fecha el mismo que devolverá registros máximos, mínimos en la fecha seleccionada como lo muestra la Figura 51.

**Figura 51**  
*Sección Humedad*



The screenshot shows a web application interface for humidity data. At the top left, it says 'Bloque Norte'. Below that, there is a section titled 'Seleccione un Fecha:' with a date input field containing '2018-03-16' and a calendar icon. A blue button labeled 'Consultar' is positioned below the date field. Underneath, a 'Resultados' section displays two columns: 'Humedad Minima' with the value '25.5' and 'Humedad Maxima' with the value '85.4'.

Resultados	
Humedad Minima	Humedad Maxima
25.5	85.4

*Nota.* Imagen que muestra la pantalla humedad en la aplicación web.

La opción Grados día desarrollo cuenta con un filtro de búsqueda tipo fecha el mismo que devolverá información de temperatura, grados día desarrollo, acompañada del detalle de registros de temperatura y humedad en dicha fecha, como lo muestra la Figura 52.

**Figura 52**  
Sección Grados Días Desarrollo

Bloque Norte

**Seleccione un Fecha:**

2018-03-16

Consultar

**Resultados**

Fecha	Temperatura Minima	Temperatura Maxima
2018-03-16	8.3	36.4

Promedio Temperatura	Grados Día Desarrollo
17.1	10.55

**Detalle**

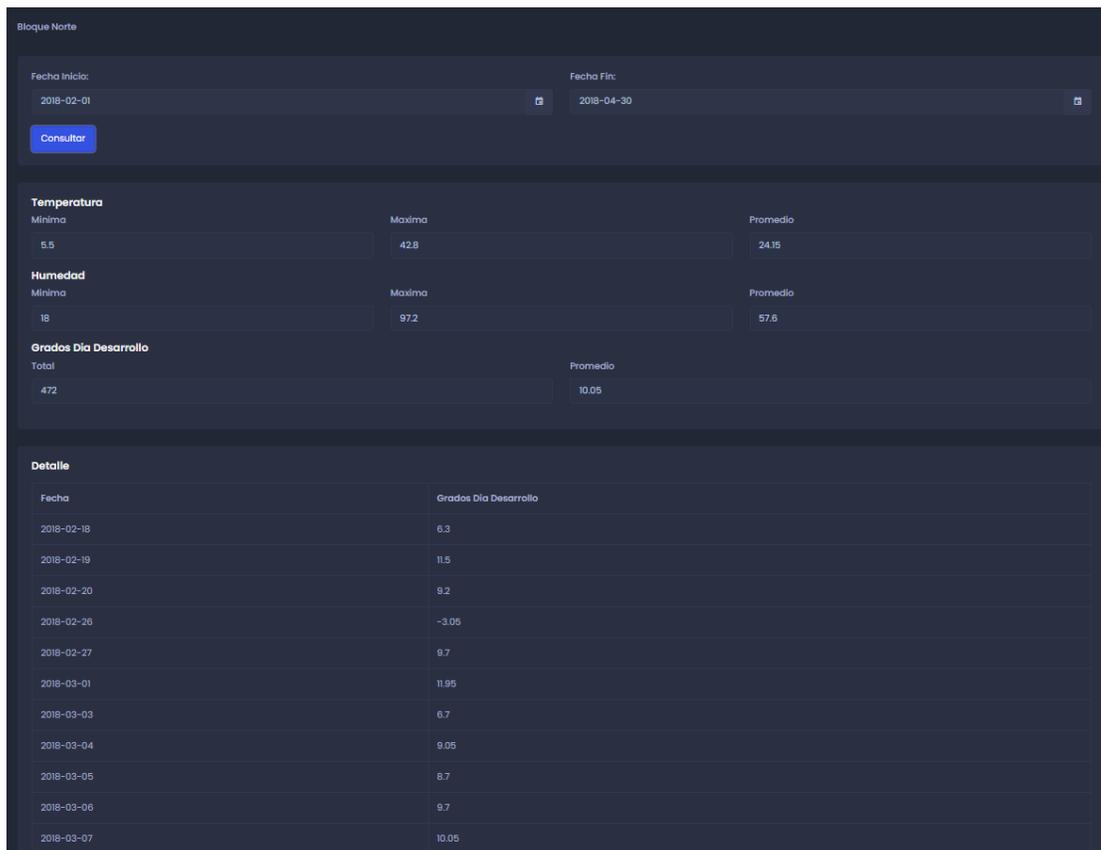
Mostrar 100 registros

Temperatura	Humedad
10.9	80.2
10.8	80.7
10.9	80.5
10.9	80.4
10.7	80.9
10.7	80.6

*Nota.* Imagen que muestra la pantalla Grados Días Desarrollo en la aplicación web.

La opción Grados día Desarrollo por Rango cuenta con dos filtros de búsqueda, ya que para esta sección se emplea un rango de fechas como filtros, los mismos que devolverán información relacionada a Humedad, Temperatura (máxima, mínima y promedio) además del total y promedio de grados día desarrollo en ese periodo. En la misma pantalla se compone un detalle de las fechas pertenecientes al periodo seleccionado con el grado día desarrollo promedio respectivo como lo muestra la Figura 53.

**Figura 53**  
*Sección Grados Días Desarrollo por Rango*



*Nota.* Imagen que muestra la pantalla Grados Días Desarrollo por Rango en la aplicación web.

La opción de estadístico es una sección la cual tiene la misma funcionalidad del gráfico estadístico, pero se le otorgo una sección exclusiva ya que al estar incrustada en el dashboard y dependiendo de las dimensiones de la pantalla donde esta se despliegue, la información se muestra muy compacta, por tal razón se agregó esta opción para el componente en nuestro menú, la que permite desplegarse la gráfica en pantalla completa y así también mostrar su información de mejor manera como se puede observar en la Figura 54.

**Figura 54**  
*Gráfico Estadístico de Grados Días Desarrollo*



*Nota.* Imagen que muestra un gráfico estadístico de Grados Días Desarrollo en la aplicación web.

### 3.4.3.2. App

#### 3.4.3.2.1. Notificaciones

El frontend cuenta con notificaciones las cuales nos indican si una petición móvil o un proceso interno se ejecutaron de manera correcta, estas aparecerán en la parte inferior de nuestra pantalla y se sirven de nuestro objeto personalizado, en concreto de nuestro mensaje validador para ser ejecutadas. La figura 55 muestra un ejemplo de las notificaciones.

**Figura 55**  
*Notificaciones en la aplicación móvil*



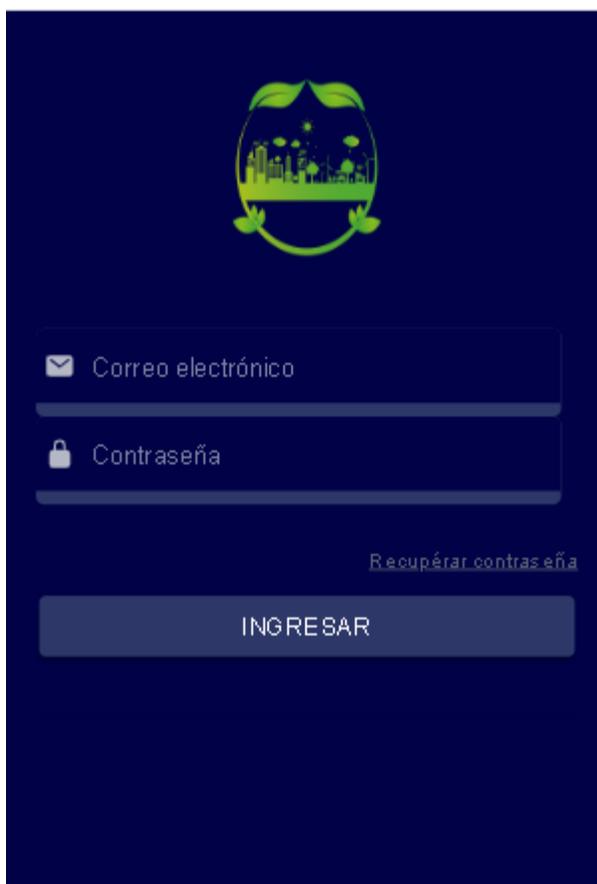
*Nota.* Imagen que muestra el tipo de notificaciones en la aplicación móvil.

### 3.4.3.2.2. Inicio de Sesión

La pantalla de Inicio de Sesión cuenta con un formulario como muestra la Figura 56, el cual recupera los datos que ingresa el usuario y un botón de “Iniciar Sesión”, el mismo que después de ser presionado emiten un evento el cual realiza la petición web al backend llevando los datos ingresados. Además, esta pantalla cuenta con un diseño en la parte superior referente a una estación de trabajo.

**Figura 56**

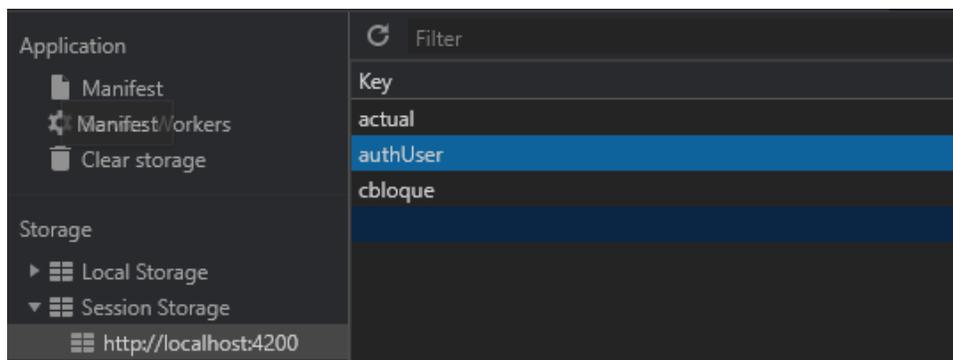
*Interfaz de login en aplicación móvil*



*Nota.* Imagen que muestra la interfaz del login para acceder a la aplicación móvil.

Finalizado el proceso de Inicio de Sesión de manera correcta, se almacenarán las credenciales del usuario en el Session Storage de nuestro navegador como se muestra en la Figura 57, las que emplearemos como filtros de búsqueda a lo largo de toda la sesión y ejecución de funcionalidades presentadas en el sistema.

**Figura 57**  
*Session Store*



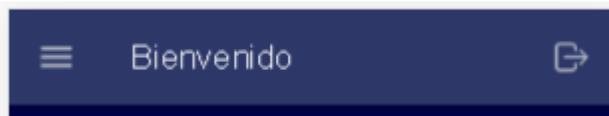
*Nota.* Imagen que muestra la variable de sesión cuando se inicia la aplicación móvil.

### 3.4.3.2.3. Menú y Navegación

Posterior al inicio de sesión el sistema redirecciona a la página principal donde se visualiza nuestra página “Resumen” la cual se tratará en la siguiente sección.

En la parte superior podemos encontrar la barra de navegación la cual está fijada y es visible en todas las pantallas disponibles, como lo muestra la Figura 58, la misma cuenta con un botón de mostrar/ocultar barra de menú que se visualiza en toda la parte izquierda de nuestra pantalla y cuenta con las secciones en las cuales se contienen todas las funcionalidades del sistema. También cuenta con un ícono ubicado en la parte superior derecha el cual nos permite cerrar sesión.

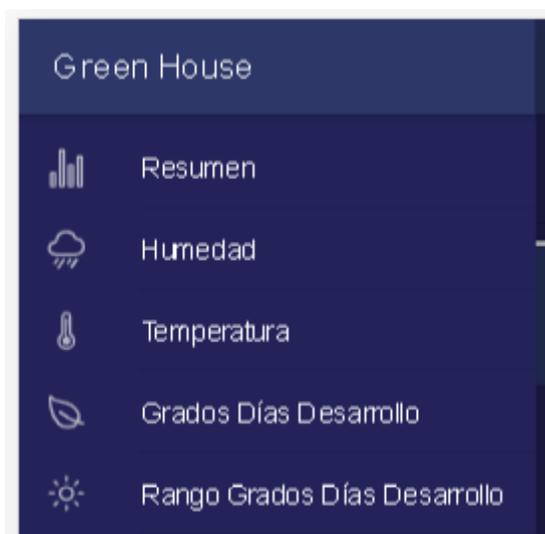
**Figura 58**  
*Barra de navegación*



*Nota.* Imagen que muestra la barra de navegación cuando se inicia la aplicación móvil.

El menú se encuentra separado en cinco secciones como muestra la Figura 59, la primera mostrando la información de bienvenida y datos requeridos por el usuario denominada "Resumen", la segunda "Humedad" y la tercera "Temperatura" muestra la información correspondiente en una determinada fecha, la cuarta "Grados Días Desarrollo" muestra la información en una determinada fecha, además, se muestra un detalle de toda esta información, y por última la quinta "Rango Grados Días Desarrollo" muestra la información en periodo de tiempo, además, se muestra un detalle de toda la información.

**Figura 59**  
*Menú de navegabilidad*



*Nota.* Imagen que muestra el menú de navegabilidad cuando se inicia la aplicación móvil.

#### 3.4.3.2.4. Resumen

El Resumen está compuesto por varias secciones o componentes los cuales trabajan de manera independiente y cada uno responde a una petición móvil única. Dentro de las secciones con las que cuenta esta pantalla están las condiciones actuales del bloque las mismas que como su nombre lo dice, nos muestran las condiciones de un determinado bloque mensualmente. Esta información se encuentra contenida en un slider como se muestra en la Figuras 60, 61 y 62.

**Figura 60**

*Condiciones actuales de temperatura del bloque actual.*



*Nota.* Imagen que muestra las condiciones actuales de temperatura en la aplicación móvil.

**Figura 61**

*Condiciones actuales de humedad del bloque actual.*



*Nota.* Imagen que muestra las condiciones actuales de humedad en la aplicación móvil.

**Figura 62**

*Condiciones actuales de Grados Día Desarrollo del bloque actual.*



*Nota.* Imagen que muestra las condiciones actuales de Grados Día Desarrollo en la aplicación móvil.

Otra sección importante es la de información de usuario e información del bloque actual como se muestra en la Figura 63 y la Figura 64 respectivamente, esto nos permite saber en qué bloque nos encontramos, además se configurará este bloque como un filtro en nuestra variable de sesión en el Session Storage del Navegador como se observa en la Figura 65, para poder efectuar peticiones cada que este bloque cambie.

**Figura 63**

Información del bloque actual dentro de la pestaña Resumen



*Nota.* Imagen que muestra la información del bloque actual en la aplicación móvil.

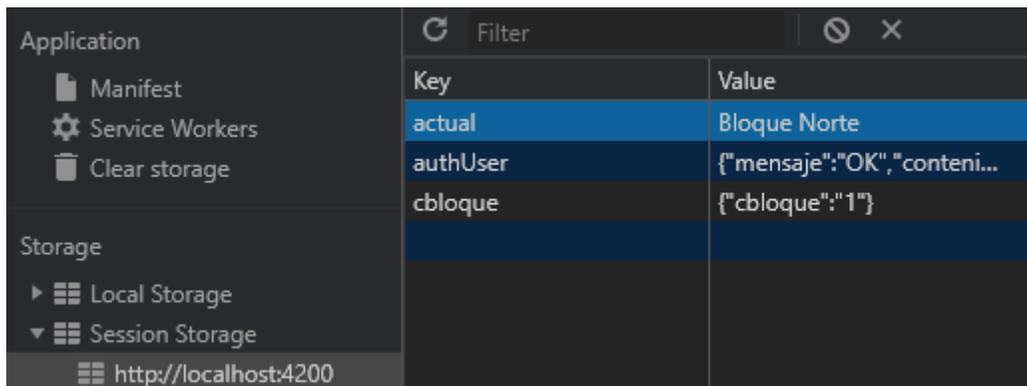
**Figura 64**

Selección del bloque



*Nota.* Imagen que muestra todos los bloques existentes en la aplicación móvil.

**Figura 65**  
*Cambio de valores en Session Storage*



*Nota.* Imagen que muestra el cambio de la variable de sesión en la aplicación móvil.

El resumen también cuenta con una sección de búsqueda como muestra la Figura 66, que permite al usuario información mensual dependiendo el valor ingresado.

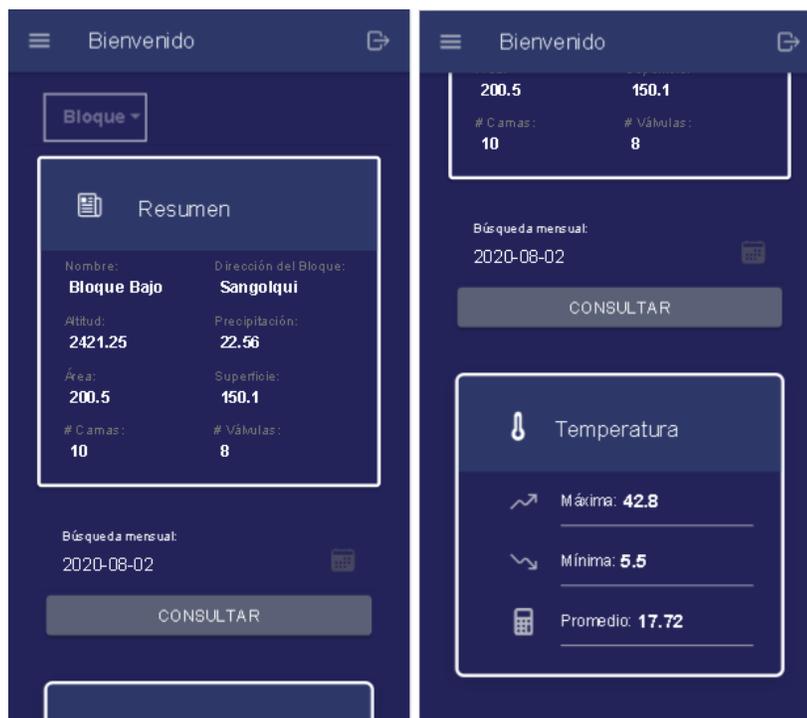
**Figura 66**  
*Búsqueda Mensual*



*Nota.* Imagen que muestra una gráfica del último mes de los grados días desarrollo en la aplicación web.

En la Figura 67 podemos observar la pantalla de nuestro Resumen de manera completa, así como la distribución de cada uno de los componentes antes mencionados.

**Figura 67**  
*Pantalla completa del Resumen*



*Nota.* Imagen que muestra la pantalla completa del resumen en la aplicación móvil.

#### 3.4.3.2.5. Humedad y Temperatura

Las opciones de Humedad y Temperatura comparten la misma presentación ya que estas poseen únicamente un filtro de búsqueda tipo fecha el mismo que devolverá registros máximos, mínimos en la fecha seleccionada como lo muestra la Figura 68 y Figura 69 respectivamente.

**Figura 68**  
*Sección Humedad*



*Nota.* Imagen que muestra la pantalla humedad en la aplicación móvil.

**Figura 69**  
*Sección Temperatura*

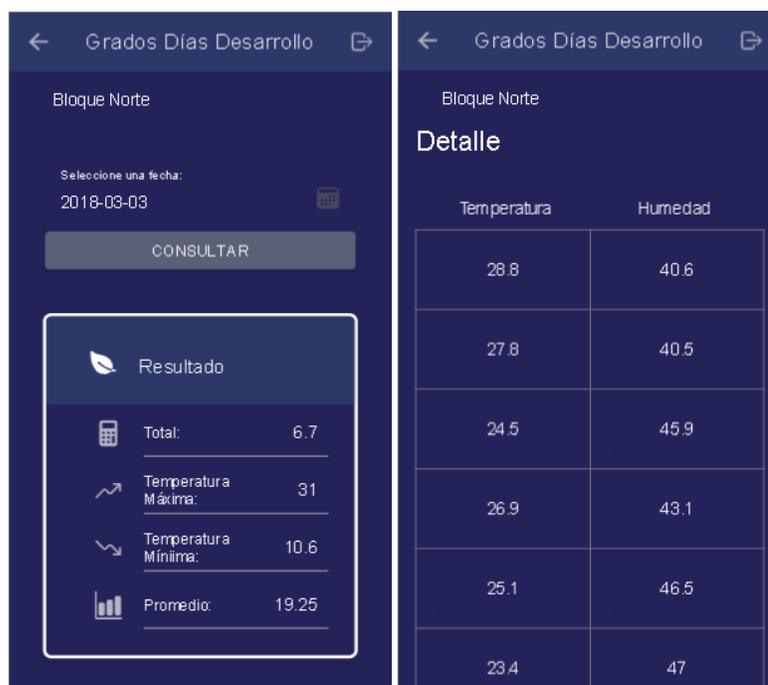


*Nota.* Imagen que muestra la pantalla temperatura en la aplicación móvil.

### 3.4.3.2.6. Grados días desarrollo

La opción Grados día desarrollo cuenta con un filtro de búsqueda tipo fecha el mismo que devolverá información de temperatura, grados día desarrollo, acompañada del detalle de registros de temperatura y humedad en dicha fecha, como lo muestra la Figura 70.

**Figura 70**  
Sección Grados Días Desarrollo

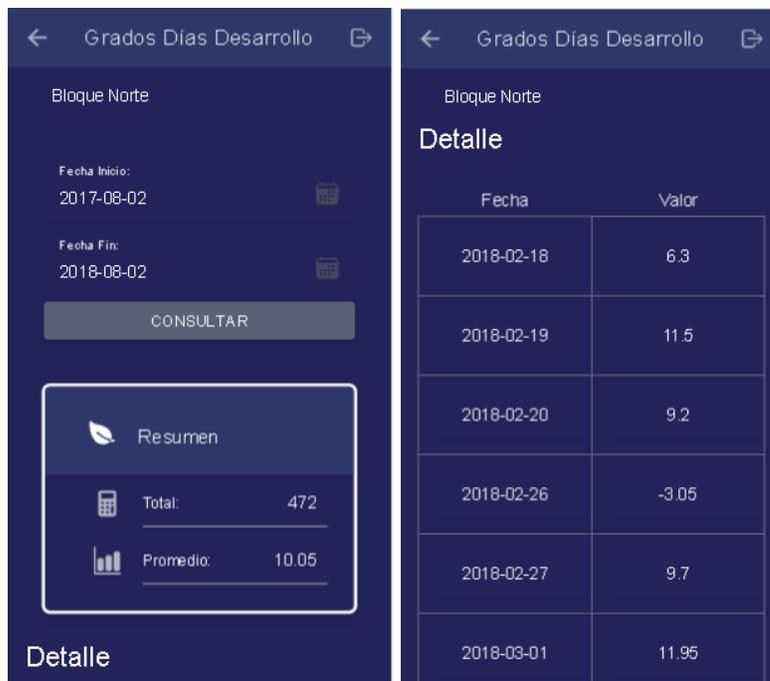


*Nota.* Imagen que muestra la pantalla Grados Días Desarrollo en la aplicación móvil.

### 3.4.3.2.7. Grados días desarrollo por rango

La opción Grados día Desarrollo por Rango cuenta con dos filtros de búsqueda, ya que para esta sección se emplea un rango de fechas como filtros, los mismos que devolverán información relacionada a Humedad, Temperatura (máxima, mínima y promedio) además del total y promedio de grados día desarrollo en ese periodo. En la misma pantalla se compone un detalle de las fechas pertenecientes al periodo seleccionado con el grado día desarrollo promedio respectivo como lo muestra la Figura 71.

**Figura 71**  
*Sección Grados Días Desarrollo por Rango*



*Nota.* Imagen que muestra la pantalla Grados Días Desarrollo por Rango en la aplicación móvil.

### 3.5. Validación y pruebas

Todo el sistema fue probado en conjunto de manera local y también en un servidor pagado de AWS, siguiendo las directrices solicitadas las cuales se detallan en el siguiente capítulo.

## Capítulo IV

### Pruebas y Resultados

En este capítulo se muestra las pruebas que se efectuaron sobre el sistema trabajando de manera conjunta, los resultados obtenidos y la evaluación de satisfacción, interacción y usabilidad del sistema.

#### 4.1. Pruebas

##### 4.1.1. Backend

##### 4.1.1.1. Servicios

Todos los servicios del Backend fueron testeados con la herramienta “Postman” así como directamente con las peticiones del Frontend, en esta sección se mostrará las pruebas correspondientes a la herramienta “Postman” y como fue el comportamiento de los servicios expuestos por la aplicación.

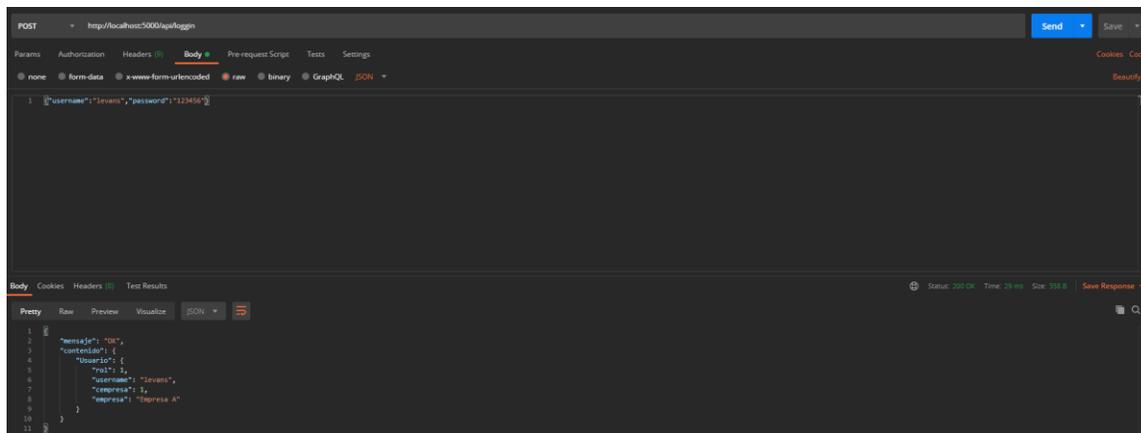
##### 4.1.1.1.1. Casos de Éxito

A continuación, se muestra una lista con las rutas de acceso a los diferentes servicios y la respuesta recibida a la petición de la herramienta “Postman” a los mismos, los servicios se encuentran desplegados de manera local en la dirección <http://localhost:5000/api>.

- /login

### Figura 72

*Pruebas Backend de la ruta login*

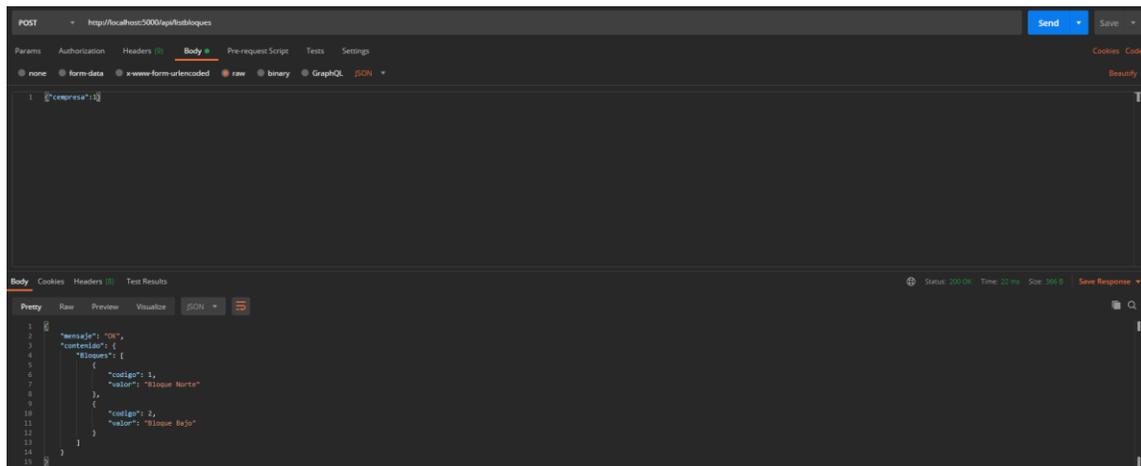


*Nota.* Imagen que muestra la prueba response de la ruta login en el backend.

- /listbloques

### Figura 73

*Pruebas Backend de la ruta listbloques*

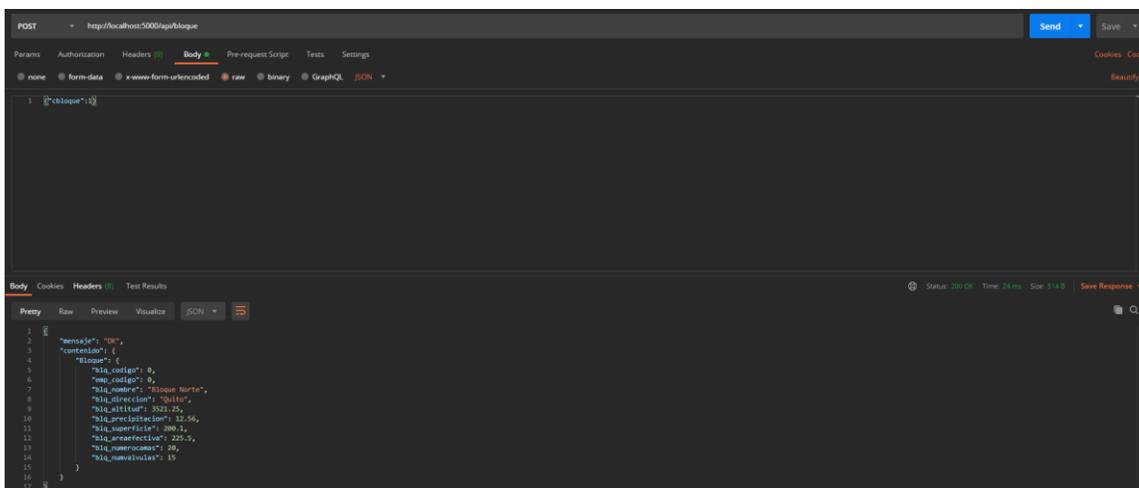


*Nota.* Imagen que muestra la prueba response de la ruta listbloques en el backend.

- /bloque

**Figura 74**

*Pruebas Backend de la ruta bloque*

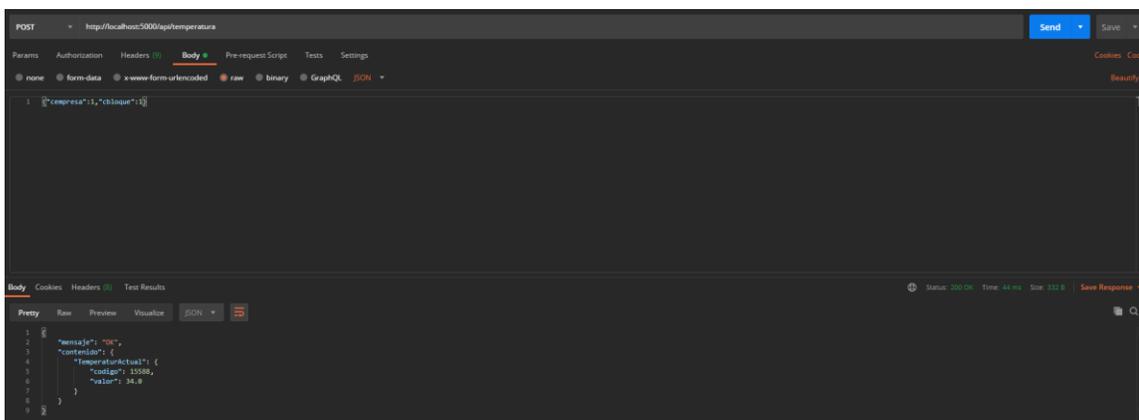


*Nota.* Imagen que muestra la prueba response de la ruta bloque en el backend.

- /temperatura

**Figura 75**

*Pruebas Backend de la ruta temperatura*

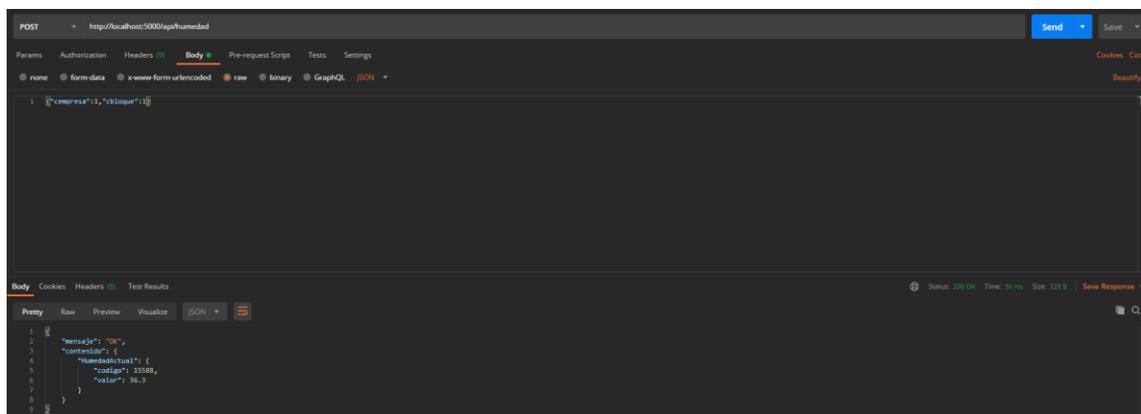


*Nota.* Imagen que muestra la prueba response de la ruta temperatura en el backend.

- /humedad

### Figura 76

Pruebas Backend de la ruta humedad

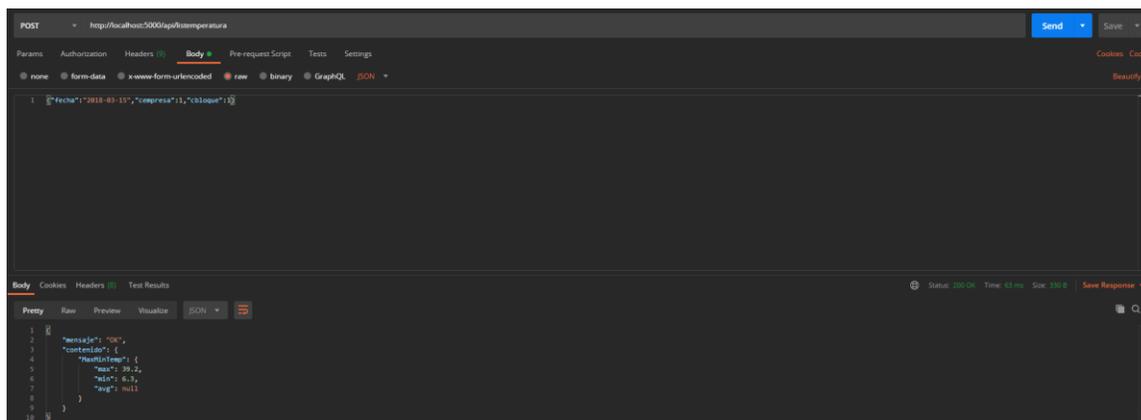


*Nota.* Imagen que muestra la prueba response de la ruta humedad en el backend.

- /listemperatura

### Figura 77

Pruebas Backend de la ruta listemperatura

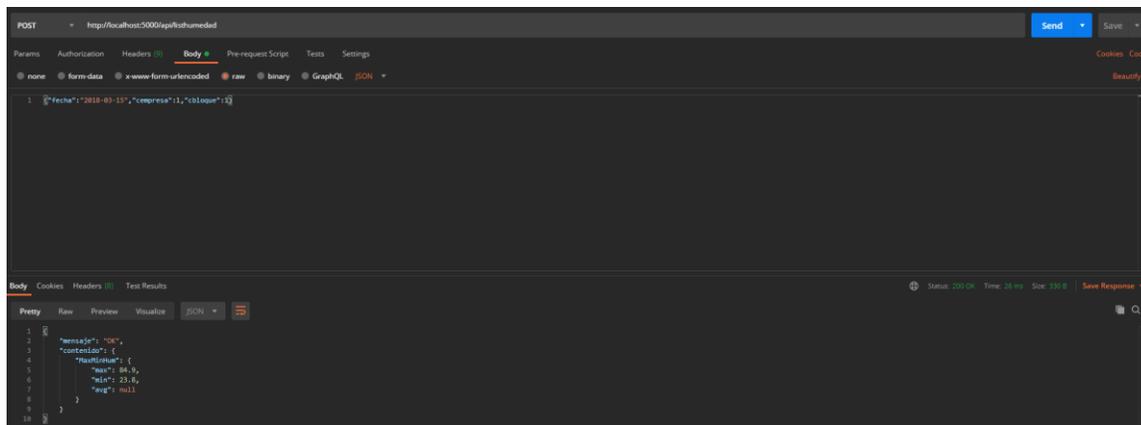


*Nota.* Imagen que muestra la prueba response de la ruta listemperatura en el backend.

- /listhumedad

### Figura 78

*Pruebas Backend de la ruta listhumedad*

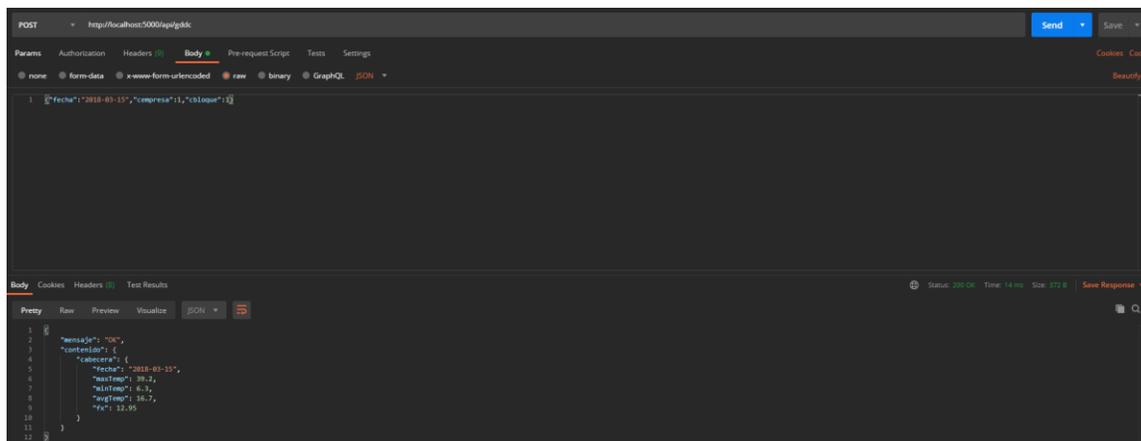


*Nota.* Imagen que muestra la prueba response de la ruta listhumedad en el backend.

- /gddc

### Figura 79

*Pruebas Backend de la ruta gddc*

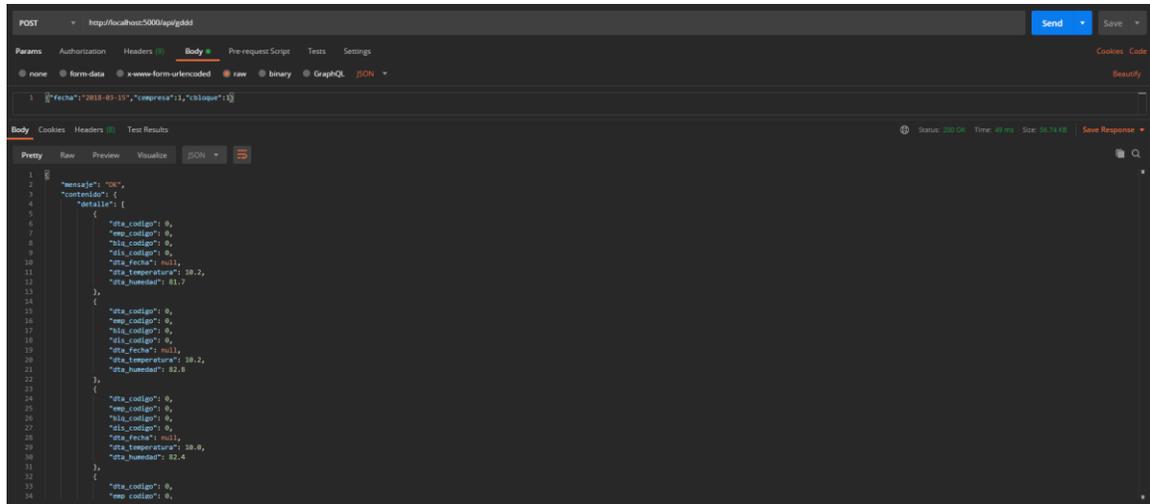


*Nota.* Imagen que muestra la prueba response de la ruta gddc en el backend.

- /gddd

**Figura 80**

*Pruebas Backend de la ruta gddd*

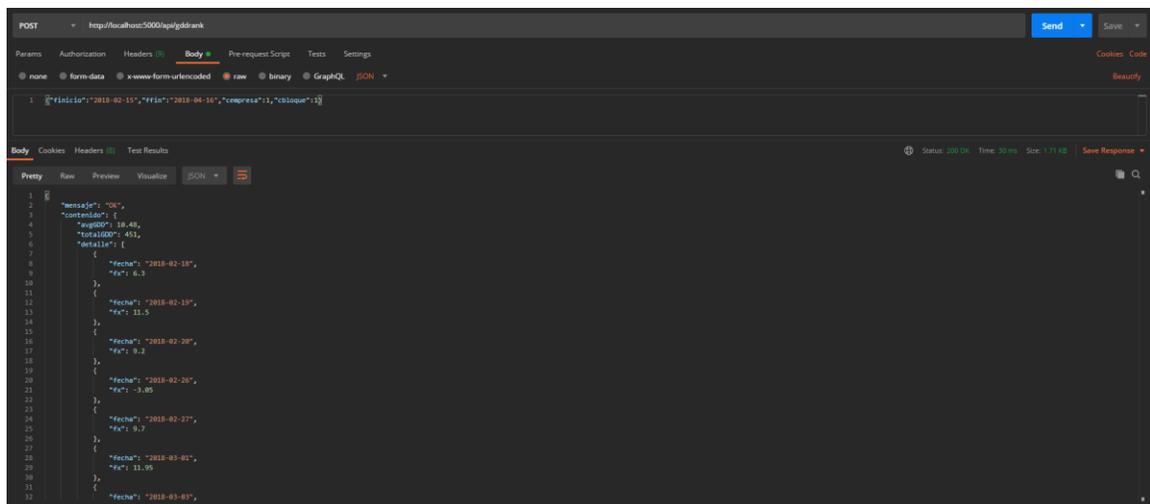


*Nota.* Imagen que muestra la prueba response de la ruta gddd en el backend.

- /gddrank

**Figura 81**

*Pruebas Backend de la ruta gddrank*

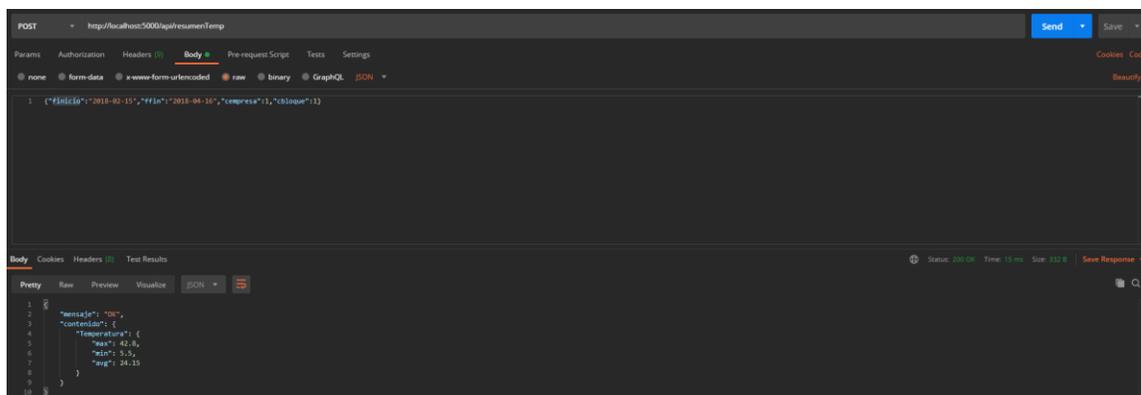


*Nota.* Imagen que muestra la prueba response de la ruta gddrank en el backend.

- `/resumenTemp`

### Figura 82

*Pruebas Backend de la ruta resumenTemp*

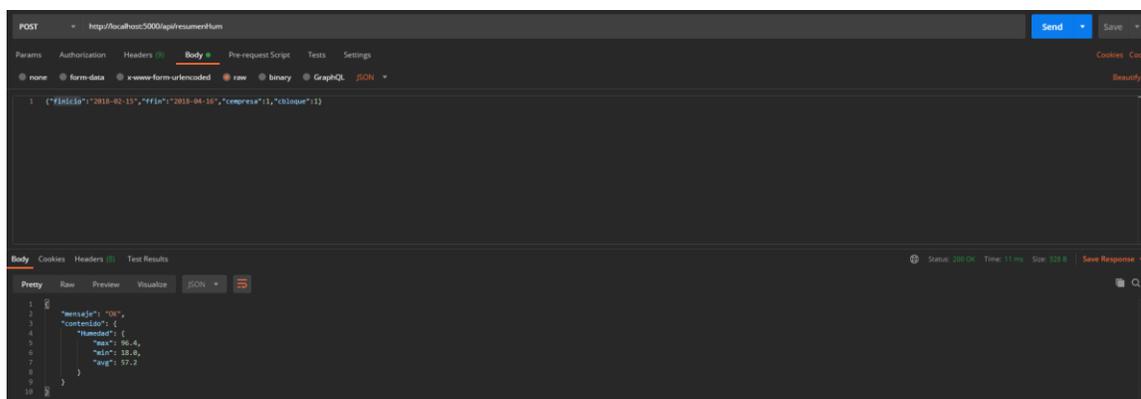


*Nota.* Imagen que muestra la prueba response de la ruta resumenTemp en el backend.

- `/resumenHum`

### Figura 83

*Pruebas Backend de la ruta resumenHum*

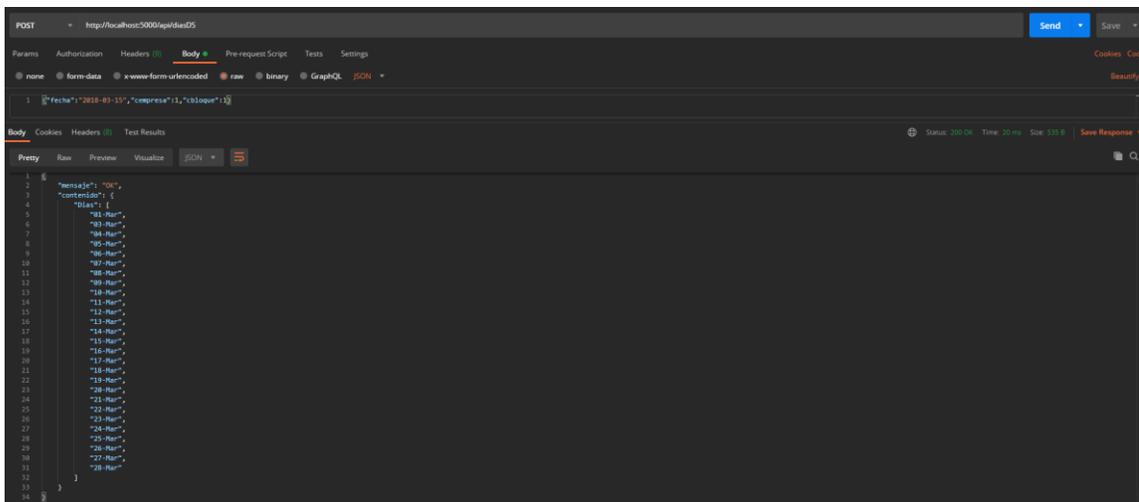


*Nota.* Imagen que muestra la prueba response de la ruta resumenHum en el backend.

- /diasDS

### Figura 84

Pruebas Backend de la ruta diasDS

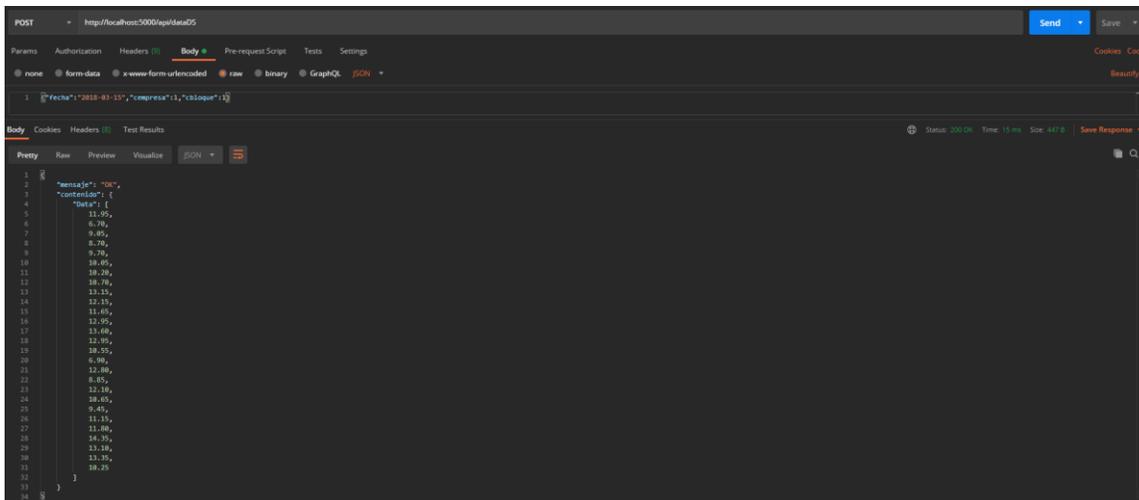


Nota. Imagen que muestra la prueba respuesta de la ruta diasDS en el backend.

- /dataDS

### Figura 85

Pruebas Backend de la ruta dataDS

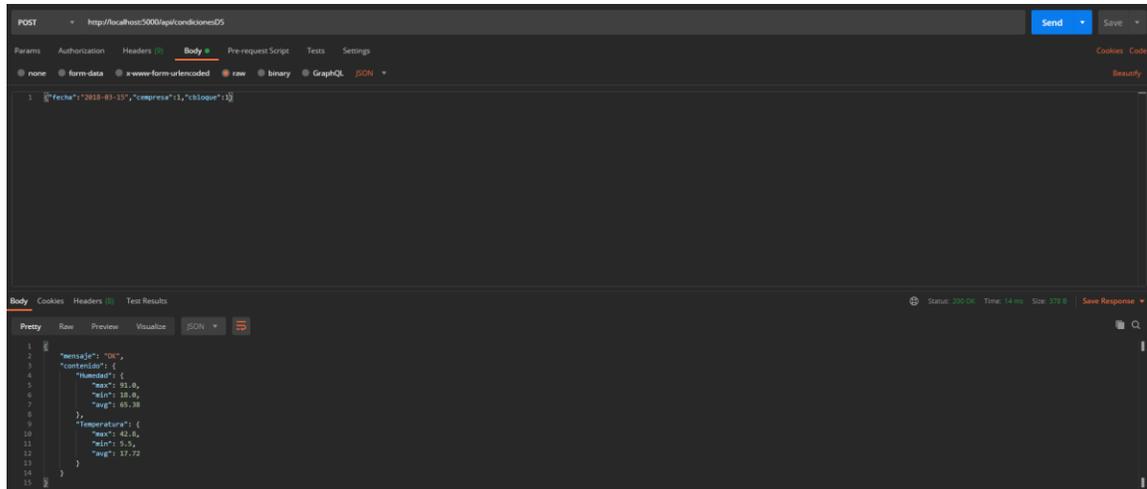


Nota. Imagen que muestra la prueba respuesta de la ruta dataDS en el backend.

- /condicionesDS

**Figura 86**

*Pruebas Backend de la ruta condicionesDS*



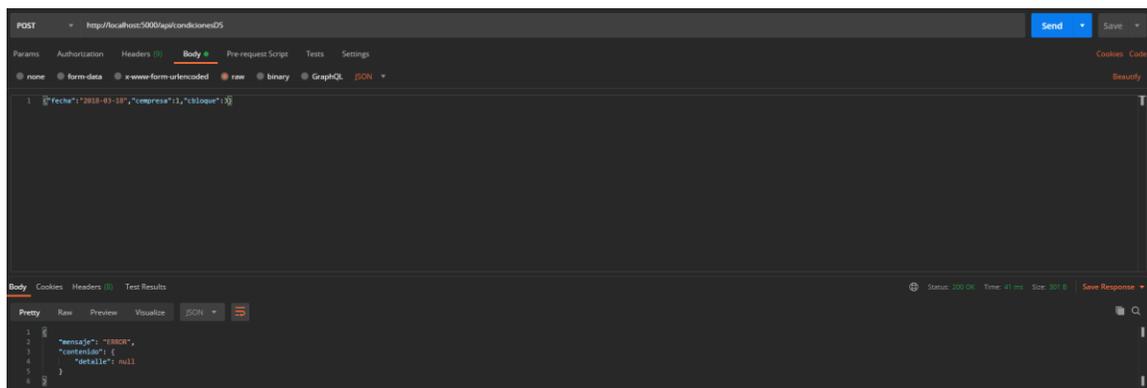
*Nota.* Imagen que muestra la prueba response de la ruta condicionesDS.

#### 4.1.1.1.2. Casos Fallidos

Al contar nuestra aplicación con un nivel personalizable en los objetos de respuesta, no importa cuál sea el servicio que se desee consumir, el objeto de respuesta al ser este un caso fallido siempre será el mismo y se observa en la parte inferior de la Figura 87.

**Figura 87**

*Caso fallido*



*Nota.* Imagen que muestra la prueba response errónea en el backend.

## 4.1.2. Frontend

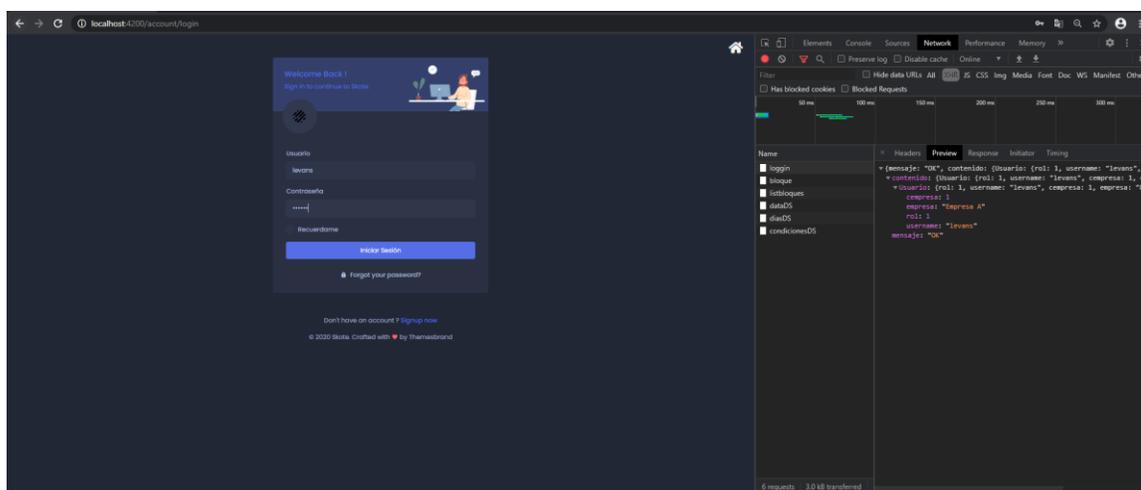
### 4.1.2.1. Web

Las pruebas referentes al frontend se realizaron con el sistema desplegado de manera local y extrayendo la información directamente de la consola del navegador.

- Inicio de Sesión

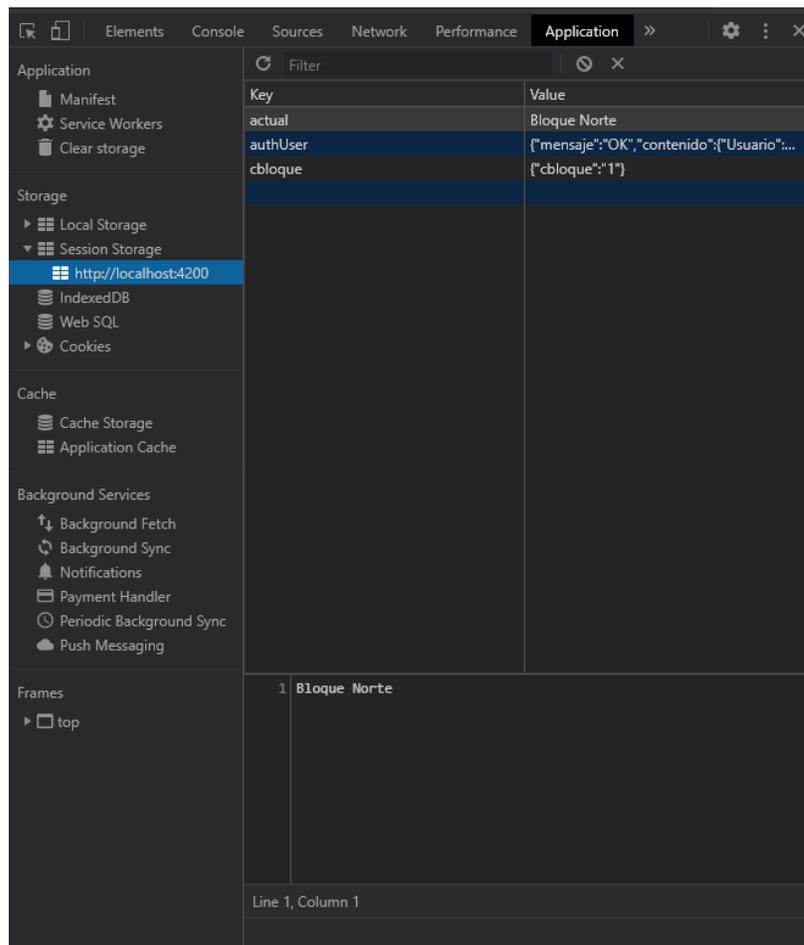
#### Figura 88

##### *Pruebas FrontEnd Login*



*Nota.* Imagen que muestra la prueba response de login en el frontend.

**Figura 89**  
*Session Storage cuando inicia sesión*

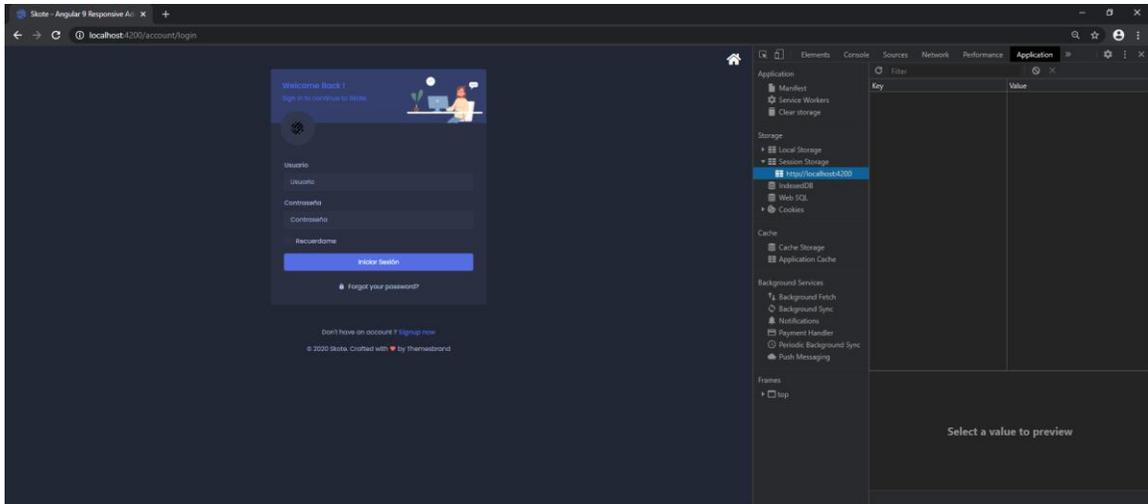


*Nota.* Imagen que muestra la variable de sesión al ingresar a la aplicación en el frontend.

- Cierre de Sesión

**Figura 90**

*Session Storage cuando se cierra sesión*

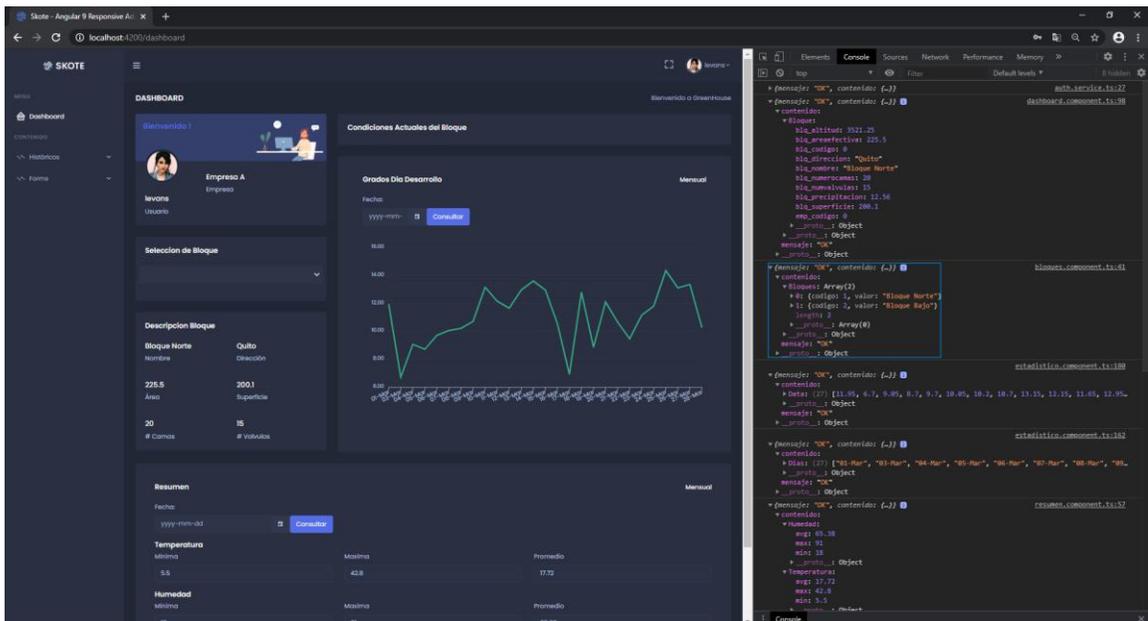


*Nota.* Imagen que muestra la variable de sesión al cerrar la aplicación en el frontend.

- Dashboard

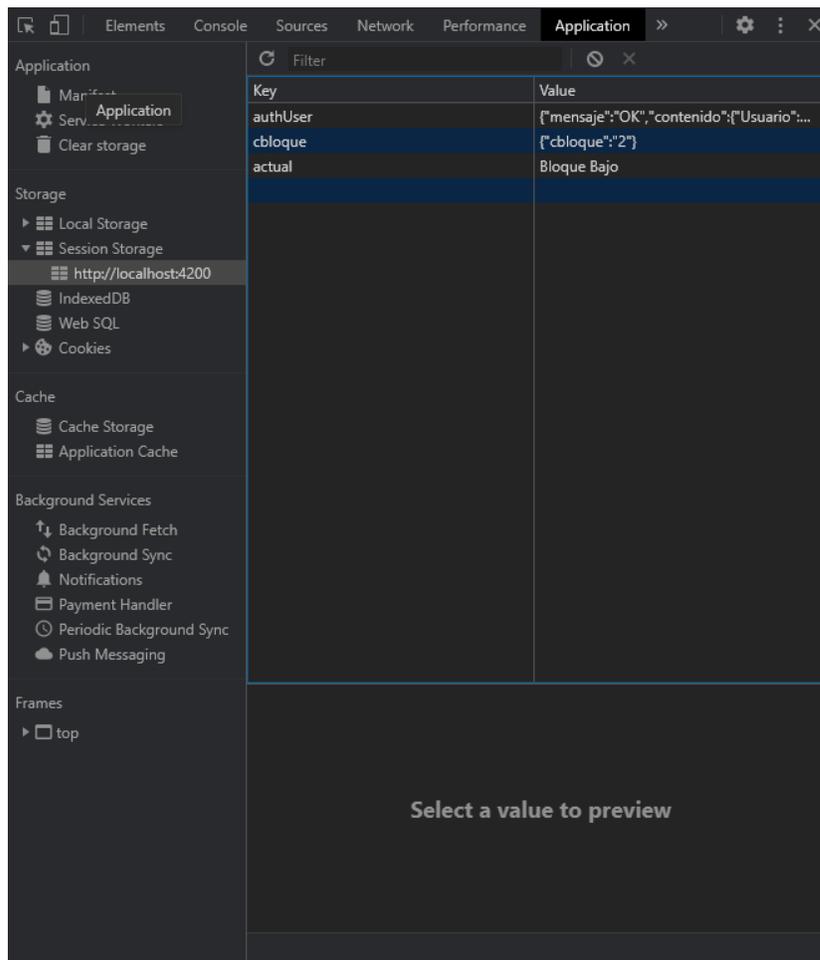
**Figura 91**

*Pruebas en la sección Dashboard*



*Nota.* Imagen que muestra las impresiones en consola de las respuestas obtenidas del frontend.

**Figura 92**  
*Session Storage en la sección Dashboard*

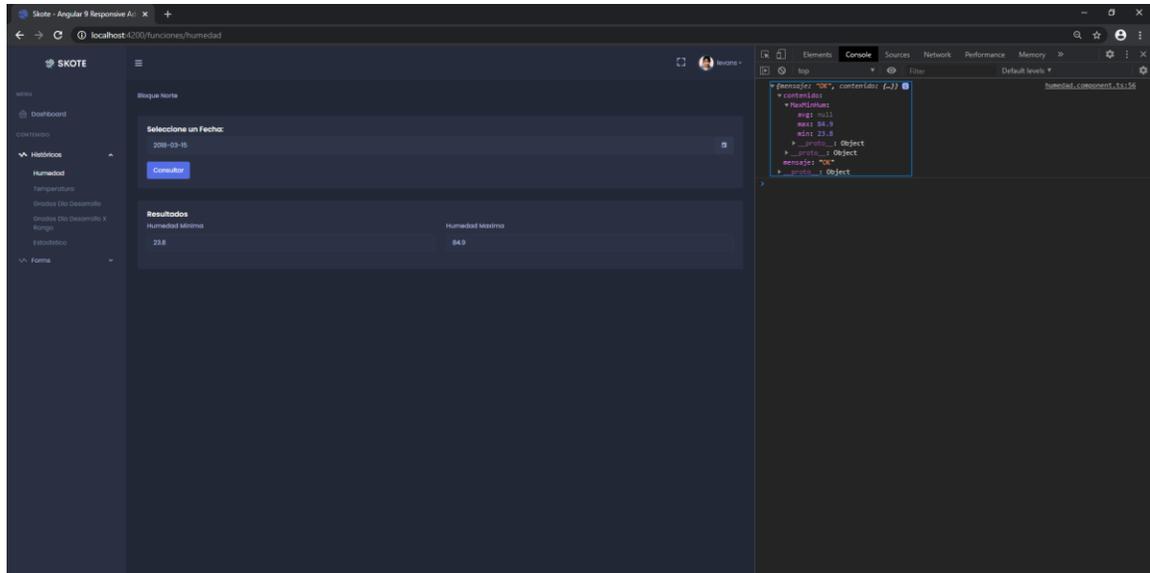


*Nota.* Imagen que muestra los valores del session storage.

- **Humedad**

**Figura 93**

*Pruebas en la sección Humedad*

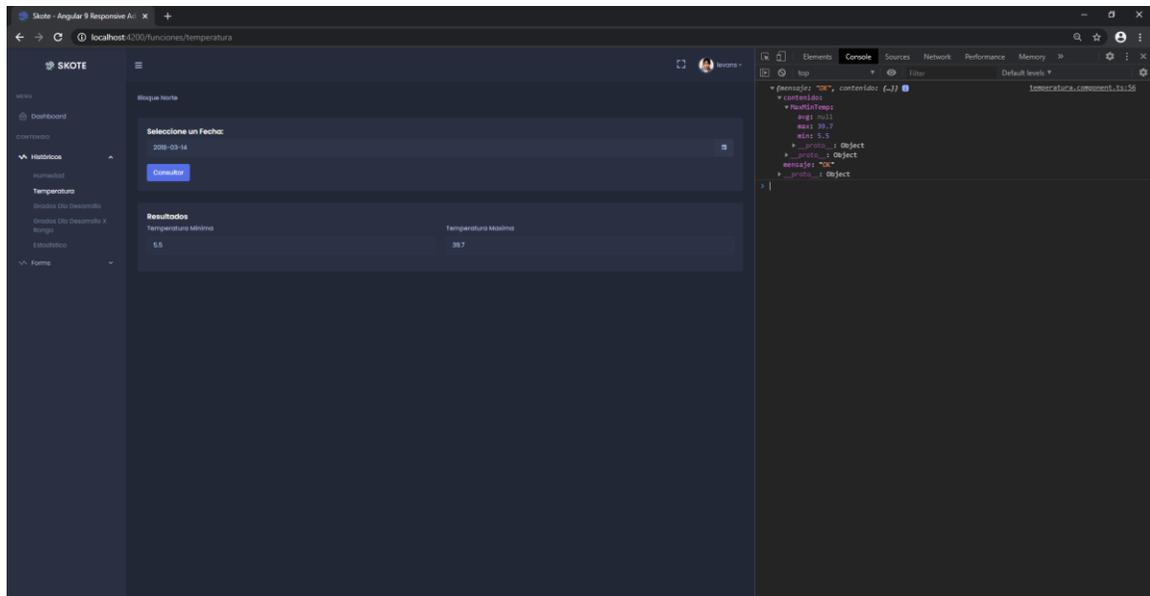


*Nota.* Imagen que muestra las impresiones en consola de las respuestas obtenidas del frontend.

- **Temperatura**

**Figura 94**

*Pruebas en la sección Temperatura*



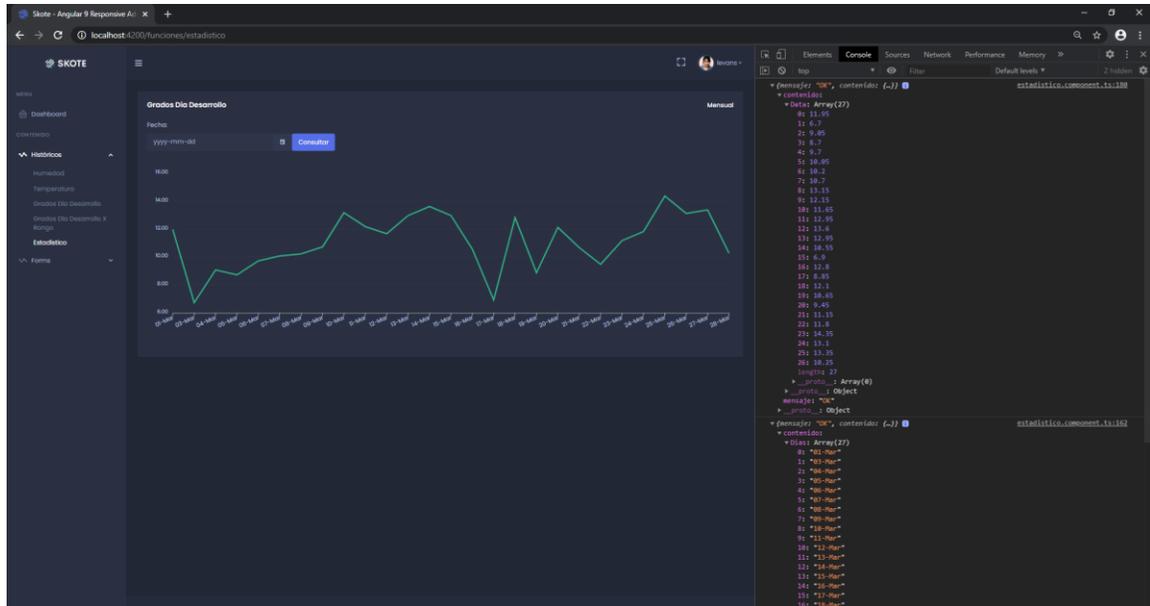
*Nota.* Imagen que muestra las impresiones en consola de las respuestas obtenidas del frontend.



- Estadístico

### Figura 97

Pruebas en la sección Estadístico



Nota. Imagen que muestra las impresiones en consola de las respuestas obtenidas del frontend.

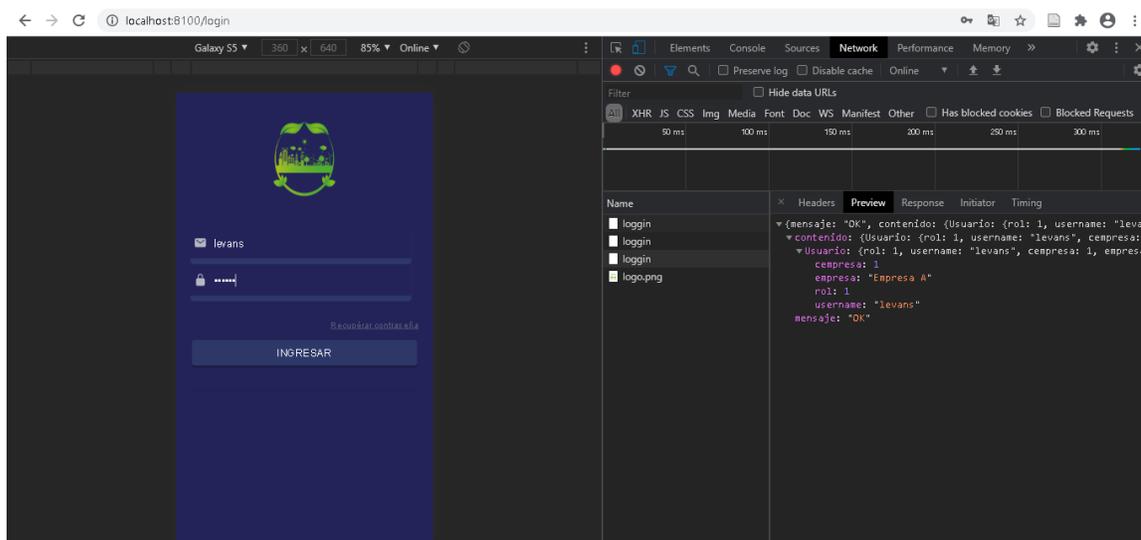
#### 4.1.2.2. App

Las pruebas referentes al frontend se realizaron con el sistema desplegado de manera local y extrayendo la información directamente de la consola del navegador.

- Inicio de Sesión

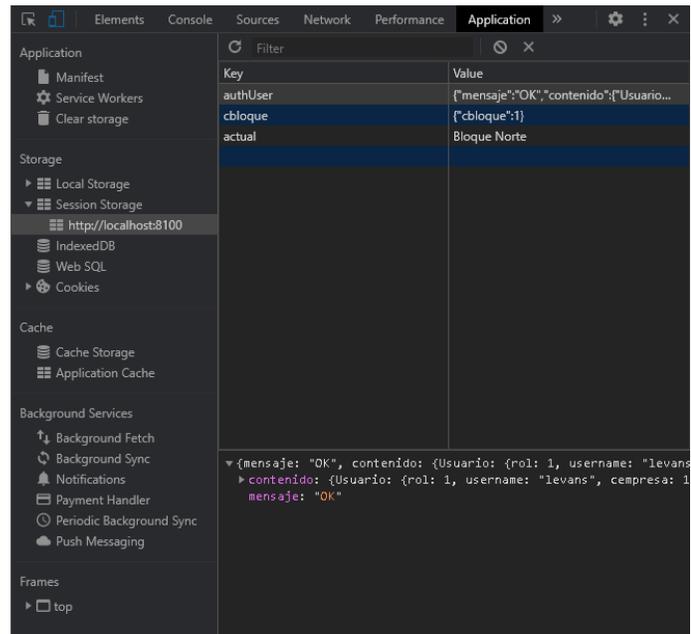
#### Figura 98

*Pruebas FrontEnd Login en la aplicación móvil*



*Nota.* Imagen que muestra la prueba response de login en el frontend.

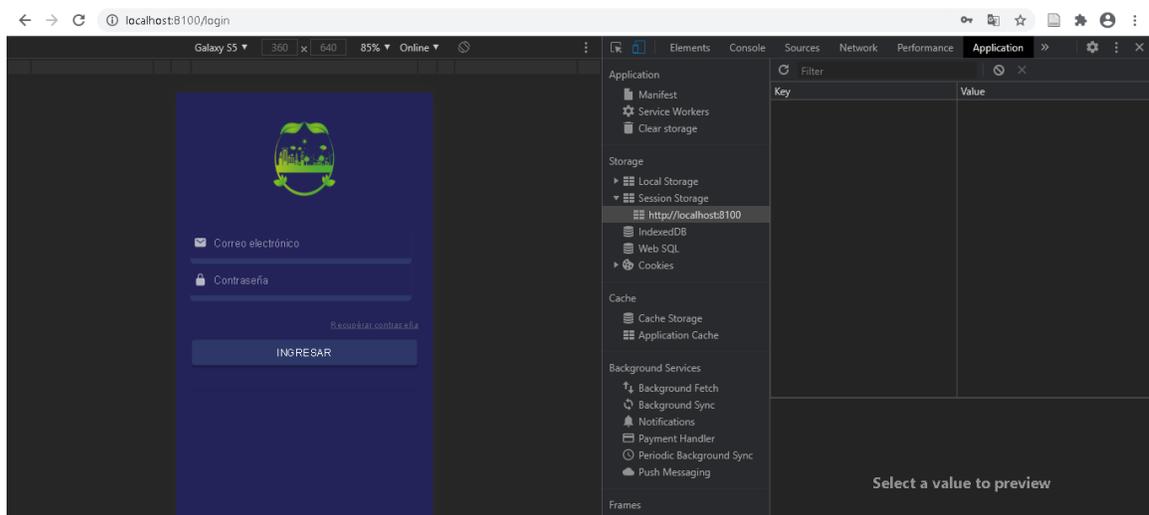
**Figura 99**  
*Session Storage cuando inicia sesión en la aplicación móvil*



*Nota.* Imagen que muestra la variable de sesión al ingresar a la aplicación móvil en el frontend.

- **Cierre de Sesión**

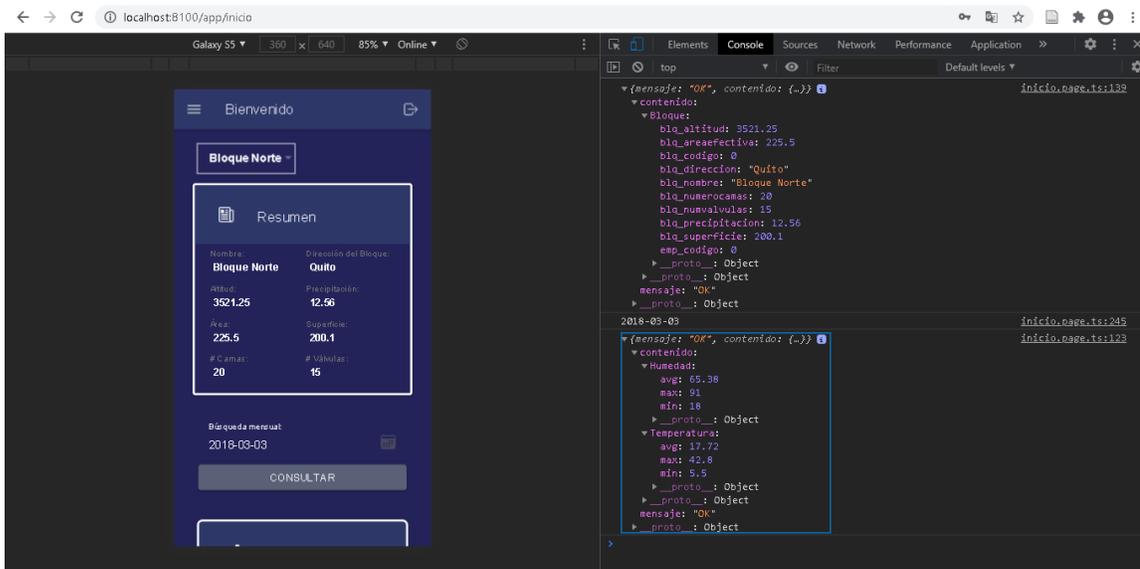
**Figura 100**  
*Session Storage cuando se cierra sesión en la aplicación móvil*



*Nota.* Imagen que muestra la variable de sesión al cerrar la aplicación en el frontend.

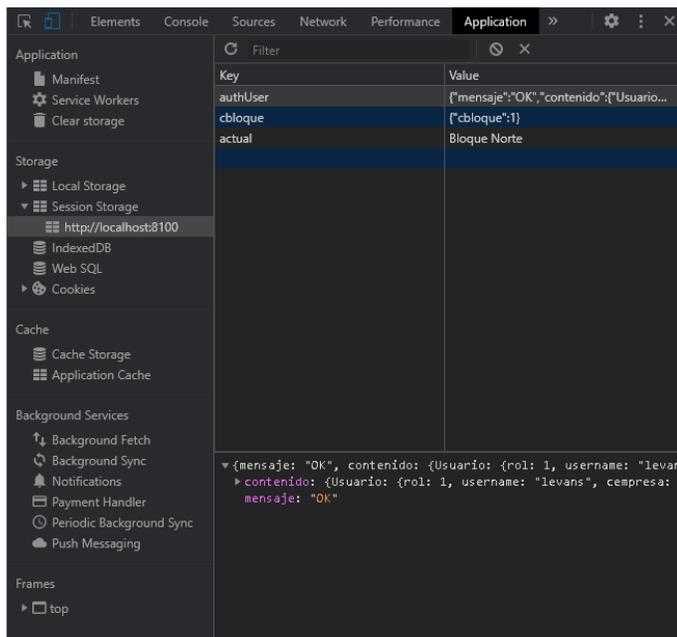
- **Resumen**

**Figura 101**  
Pruebas en la sección Resumen de la aplicación móvil



*Nota.* Imagen que muestra las impresiones en consola de las respuestas obtenidas del frontend.

**Figura 102**  
Session Storage en la sección Resumen

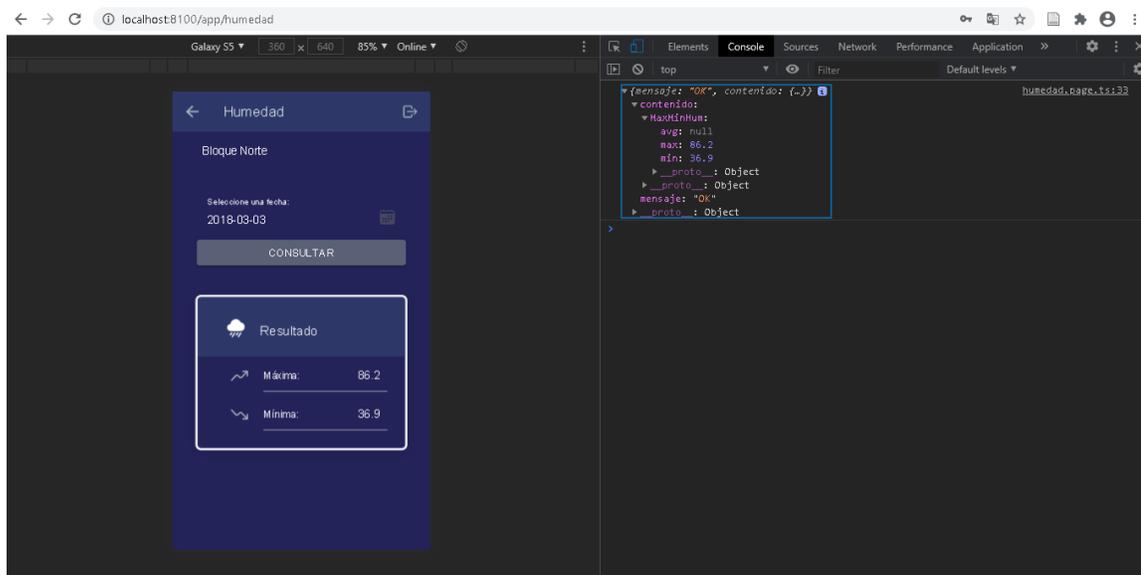


*Nota.* Imagen que muestra los valores del session storage.

- **Humedad**

**Figura 103**

*Pruebas en la sección Humedad de la aplicación móvil*

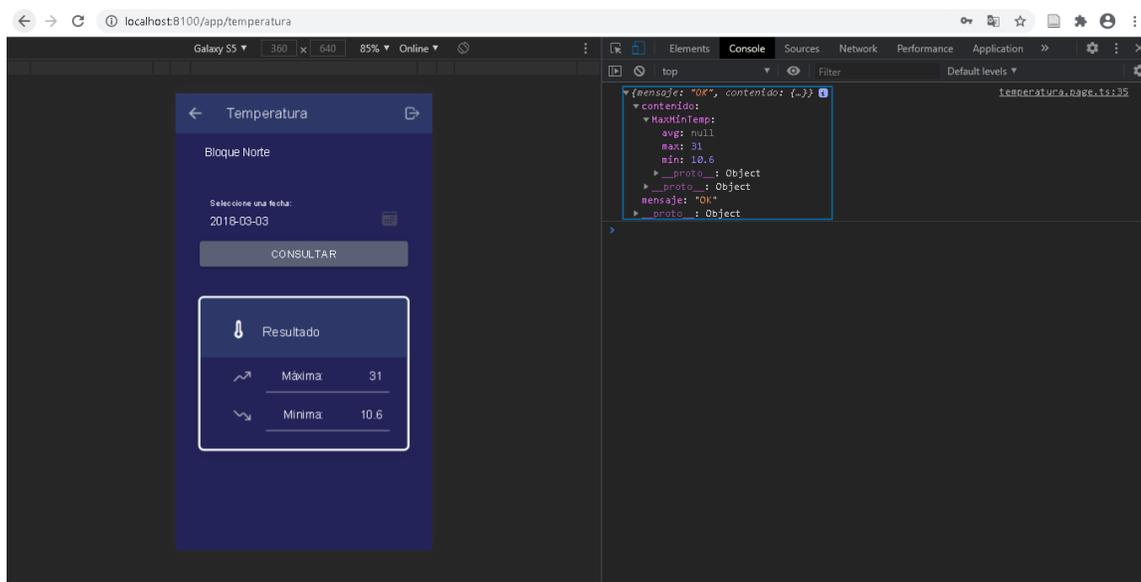


*Nota.* Imagen que muestra las impresiones en consola de las respuestas obtenidas del frontend.

- **Temperatura**

**Figura 104**

*Pruebas en la sección Temperatura de la aplicación móvil*



*Nota.* Imagen que muestra las impresiones en consola de las respuestas obtenidas del frontend.



## 4.2. Resultados

### 4.2.1. Análisis de resultados sobre la satisfacción del usuario

Se realizó una encuesta de satisfacción a 14 personas que tienen conocimientos básicos de informática. Cada una de ellas tuvo una introducción de la usabilidad de la aplicación móvil y web.

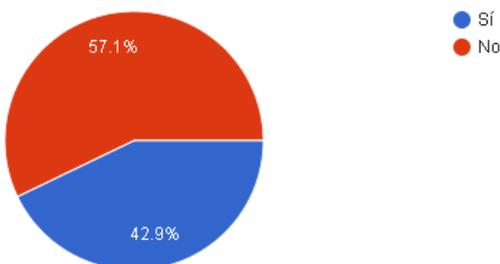
Luego de esta pequeña introducción cada persona tuvo la oportunidad de entrar y navegar en las aplicaciones con un usuario de prueba obteniendo los siguientes resultados.

De las 14 personas encuestadas solo un 42.9% tiene conocimiento de aplicaciones que permiten la visualización y monitoreo de información como muestra la figura 107.

#### Figura 107

*Resultado de análisis de usabilidad de aplicaciones de monitoreo y visualización de datos*

¿Ha usado anteriormente una aplicación para monitoreo y visualización de datos?



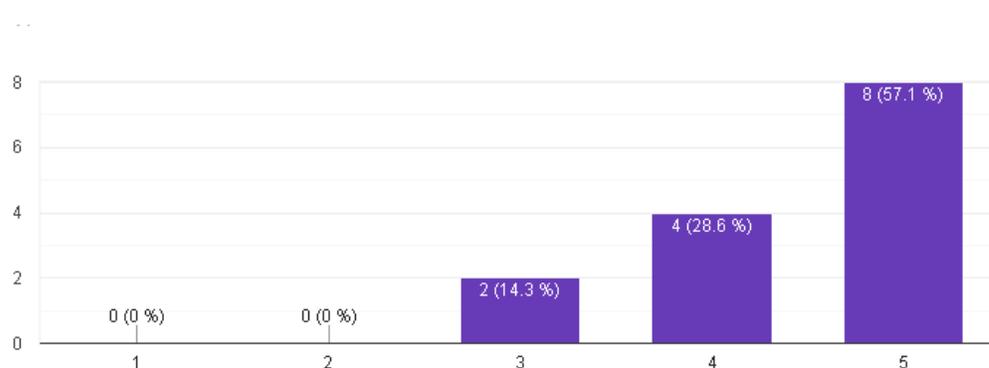
*Nota.* Esta imagen muestra la pregunta y el resultado de los análisis luego de aplicar la encuesta.

Se realizó una parametrización de valores del uno al cinco siendo cinco todo y uno nada. Un 57.1% de los encuestados entiende totalmente los datos presentados por la aplicación móvil y web, un 28.6% entiende la mayoría de la información y un 14.3% no entiende algunos datos. Esta información se muestra en la figura 108.

**Figura 108**

*Resultado de análisis de monitoreo y visualización de datos.*

¿La información que muestra la aplicación móvil y web es entendible?



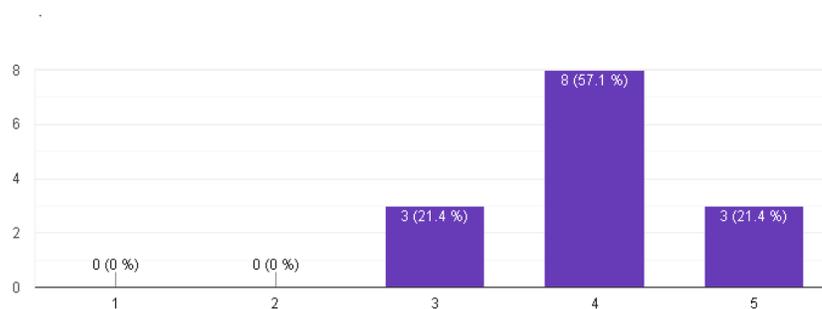
*Nota.* Esta imagen muestra la pregunta y el resultado de los análisis luego de aplicar la encuesta.

Se refleja en un 57.1% que las respuestas brindadas en la aplicación móvil y web, en su mayoría son exactas y sin ninguna inconsistencia. Mientras que en un 21.4% no tuvieron ningún inconveniente como muestra la figura 109.

**Figura 109**

*Resultado de análisis de consultas de datos.*

Las consultas y respuestas que brinda la aplicación móvil y web ¿son exactas y sin inconsistencias?

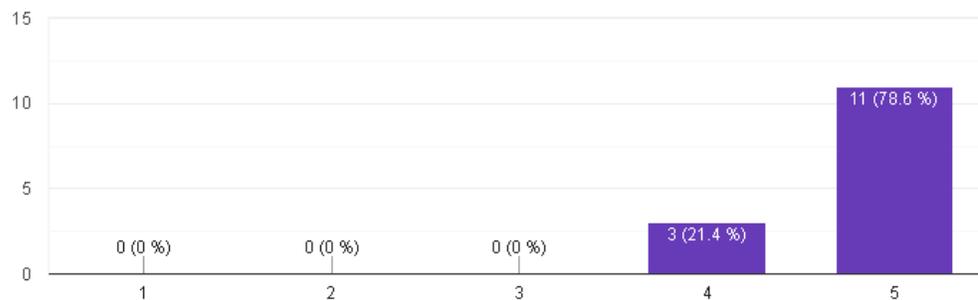


*Nota.* Esta imagen muestra la pregunta y el resultado de los análisis luego de aplicar la encuesta.

La navegabilidad de la aplicación móvil y web resultó muy fácil para los encuestados reflejando los siguientes resultados de la figura 110. Con un 78.6% indican una navegabilidad fácil y sin problemas mientras que en un 21,4% muestran un mínimo disgusto.

**Figura 110**  
*Resultado de análisis de navegabilidad.*

La navegación de la aplicación móvil y web ¿es fácil?

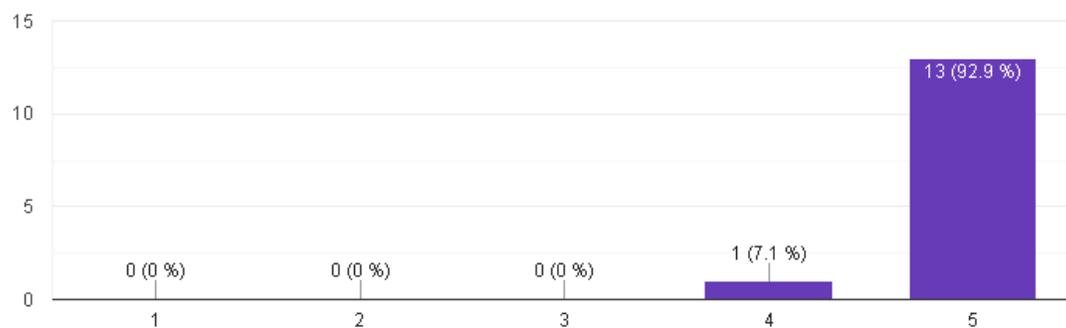


*Nota.* Esta imagen muestra la pregunta y el resultado de los análisis luego de aplicar la encuesta.

La encuesta muestra en un 92.9% lo muy agradable que fue para el usuario la aplicación web y en un 7.1% fue solamente agradable como lo muestra la figura 111.

**Figura 111**  
*Resultado de análisis de estética.*

La apariencia de la aplicación web ¿es estética y agradable?



*Nota.* Esta imagen muestra la pregunta y el resultado de los análisis luego de aplicar la encuesta.

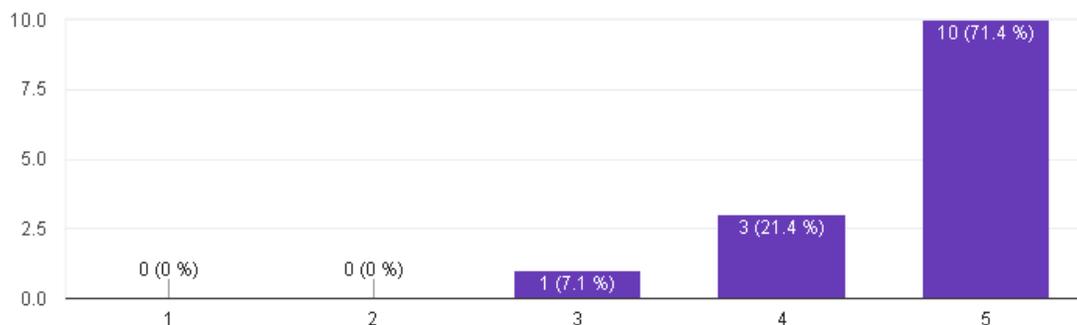
La encuesta muestra en un 71.4% lo muy agradable que fue para el usuario la aplicación móvil, en un 21.4% fue solamente agradable y en un 7.1% normal como lo muestra la figura 112.

**Figura 112**

*Resultado de análisis de estética.*

La apariencia de la aplicación móvil ¿es estética y agradable?

14 respuestas



*Nota.* Esta imagen muestra la pregunta y el resultado de los análisis luego de aplicar la encuesta.

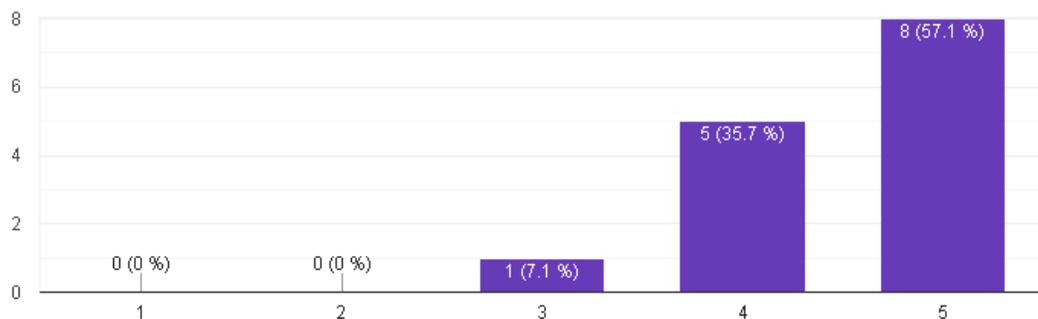
Con un 57.1% los encuestados se encuentran muy satisfechos con el tiempo de espera cuando se realiza una petición, con el 35.7% se encuentran satisfechos y un 7.1% algo satisfecho como muestra la figura 113

**Figura 113**

*Resultado de análisis de tiempo de respuesta.*

Quando solicito información a la aplicación móvil o web, ¿éste despliega información en el tiempo esperado?

14 ...



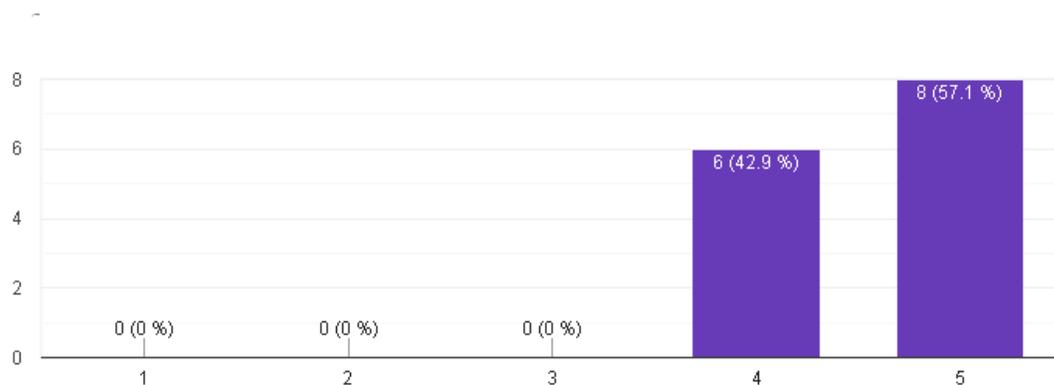
*Nota.* Esta imagen muestra la pregunta y el resultado de los análisis luego de aplicar la encuesta.

En general el 57.1% de los encuestados se encuentran muy satisfechos con la aplicación web y móvil mientras que el 42.9% se encuentran satisfechos como lo indica la figura 114

**Figura 114**

*Resultado de análisis de satisfacción.*

En general, ¿Me encuentro satisfecho con la aplicación web y móvil?



*Nota.* Esta imagen muestra la pregunta y el resultado de los análisis luego de aplicar la encuesta.

## Capítulo V

### Conclusiones y Recomendaciones

El manejo de herramientas las cuales nos permita separar la parte del cliente manteniendo una arquitectura sólida y escalable, ayuda no solo al mantenimiento del software, sino también es un punto clave para la escalabilidad de este.

A nivel comparativo el software desarrollado muestra una gran ventaja sobre otros, ya que en la actualidad aún se siguen usando sistemas basados en arquitecturas tradicionales o en su defecto usando software o servicios de terceros.

El nivel de integración que se obtuvo con el sistema fue satisfactorio ya que tanto su versión web como aplicación móvil son estables y permiten el intercambio de información con el servidor, cabe recalcar que cada una de estas es independiente y no afecta el funcionamiento de la otra además de mantener cada proceso o cambio transparente para el usuario.

La correcta manipulación y análisis de datos para una aplicación web o móvil, hace fácil de interpretar al usuario final para la toma de decisiones a futuro.

Este sistema web o móvil se lo puede implementar no solo para las flores, sino para los diferentes cultivos y animales de granja, para su monitoreo y control en el proceso del crecimiento.

La framework de trabajo de IONIC permite realizar distintas aplicaciones móviles con funcionalidades que se adaptan a cualquier sistema operativo, sea Android o iOS. Además, una de sus ventajas es la implementación de librerías de JavaScript, Java, JQuery entre otras que permiten a la aplicación móvil satisfacer, en lo mayor posible, la necesidad del usuario.

Para desplegar el sistema en red es necesario saber las versiones que se manejan tanto de parte del servidor como en la parte del cliente ya que esto puede producir ciertas alertas las cuales pueden causar deterioro en el rendimiento del sistema.

El uso de políticas de intercambio de información es totalmente necesario para mantener seguros los servicios teniendo en cuenta que estos se encontraran expuestos en internet, por esta razón se debe mantener actualizado el conocimiento y manejo de las políticas CORS.

La migración de la base de datos local a una base de datos en la nube, para evitar el proceso de traslado de información.

Los futuros desarrollos se manejarán de forma modular, ya que, mediante metodologías ágiles se puede modificar y entender el código fácilmente para posibles cambios y actualizaciones que requiera el usuario.

Mantener la capa de personalización de la aplicación móvil y web para poder obtener de forma más eficiente y eficaz respuestas por parte del servidor.

## Anexos

### Encuesta

## GreenHouse

Encuesta para conocer la satisfacción del usuario.

El presente sistema permite visualizar información relacionada con características de un bloque florícola. Se encuentra desplegado en un servidor de pruebas por lo que la data se encuentra limitada al periodo 2018-02-18 a 2018-04-31. El sistema desplegado cuenta con consultas de éxito como de error si es el caso, el usuario deberá emplear cada uno de los componentes necesarios mostrados en dicho sistema. Lea detenidamente y conteste la encuesta. Muchas Gracias

**\*Obligatorio**

1. Ingrese su correo electrónico \*

---

2. ¿Ha usado anteriormente una aplicación para monitoreo y visualización de datos? \*

Marca solo un óvalo.

Sí

No

3. ¿La información que muestra la aplicación móvil y web es entendible? \*

Marca solo un óvalo.

1    2    3    4    5

---

Muy en desacuerdo      Muy de acuerdo

4. Las consultas y respuestas que brinda la aplicación móvil y web ¿son exactas y sin inconsistencias? \*

Marca solo un óvalo.

	1	2	3	4	5	
Muy en desacuerdo	<input type="radio"/>	Muy de acuerdo				

5. La navegación de la aplicación móvil y web ¿es fácil? \*

Marca solo un óvalo.

	1	2	3	4	5	
Muy en desacuerdo	<input type="radio"/>	Muy de acuerdo				

6. La apariencia de la aplicación web ¿es estética y agradable? \*

Marca solo un óvalo.

	1	2	3	4	5	
Muy en desacuerdo	<input type="radio"/>	Muy de acuerdo				

7. La apariencia de la aplicación móvil ¿es estética y agradable? \*

Marca solo un óvalo.

	1	2	3	4	5	
Muy en desacuerdo	<input type="radio"/>	Muy de acuerdo				

8. Cuando solicito información a la aplicación móvil o web, ¿éste despliega información en el tiempo esperado? \*

Marca solo un óvalo.

	1	2	3	4	5	
Muy en desacuerdo	<input type="radio"/>	Muy de acuerdo				

9. En general, ¿Me encuentro satisfecho con la aplicación web y móvil? \*

Marca solo un óvalo.

	1	2	3	4	5	
Muy en desacuerdo	<input type="radio"/>	Muy de acuerdo				

10. ¿Usted tiene alguna observación referente a la aplicación móvil y web?

Marca solo un óvalo.

No

Otros: \_\_\_\_\_

## Bibliografía

✓ Qué es Ionic | Quality Devs | Somos Desarrolladores. (2019, mayo 31). *Quality Devs*.

<https://www.qualitydevs.com/2019/05/31/que-es-ionic-desarrollador-web/>

Almenara, J. C. (2015). Reflexiones educativas sobre las tecnologías de la información y la comunicación (TIC). *Revista Tecnología, Ciencia y Educación*, 0(1), Article 1.

<https://www.revistasocitec.org/index.php/TCE/article/view/27>

Arsys. (2017, enero 26). ¿Por qué usar Tomcat sobre Servidores Cloud? *Blog de arsys.es*.

<http://blog3.arsysdesarrollo.lan/programacion/tomcat-servidores-cloud/>

Ayala, E., & Gonzales Sánchez, S. (2015). Tecnologías de la Información y la Comunicación.

*Repositorio Institucional - UIGV*.

<http://repositorio.uigv.edu.pe/handle/20.500.11818/1189>

Calvo, D. (2018, abril 7). Metodología XP Programación Extrema (Metodología ágil). *Diego Calvo*.

<http://www.diegocalvo.es/metodologia-xp-programacion-extrema-metodologia-agil/>

Castillo, P. A. del. (2018, septiembre 28). *Desarrollo de aplicaciones multiplataforma: ¿nativo o*

*híbrido?* <https://mobivery.com/desarrollo-de-aplicaciones-multiplataforma-nativo-o-hibrido/>

Conislla, F. (2020, febrero 14). *Cómo aplicar seguridad en el ciclo de vida del desarrollo de software*. Belatrix Software Development Blog.

<https://www.belatrixsf.com/blog/seguridad-desarrollo-software>

¿CUÁLES SON LAS VENTAJAS DE UNA API REST? (2017, febrero 9). *Chakray*.

<https://www.chakray.com/es/cuales-son-las-ventajas-de-una-api-rest/>

- Delía, L. N. (2017). *Desarrollo de aplicaciones móviles multiplataforma* [Tesis, Universidad Nacional de La Plata]. <http://sedici.unlp.edu.ar/handle/10915/60497>
- Desarrollo de aplicaciones híbridas con entorno Ionic*. (2019, junio 11). #lanuevaFP. <https://fp.uoc.fje.edu/blog/desarrollo-de-aplicaciones-hibridas-con-entorno-ionic/>
- Devs, Q. (2018, noviembre 26). Aplicaciones móviles multiplataforma. *Quality Devs*. <https://www.qualitydevs.com/2018/11/26/aplicaciones-moviles-multiplataforma/>
- Ionic. (2015). *Ionic—Cross-Platform Mobile App Development*. Ionic Framework. <https://ionicframework.com/>
- IONOS. (2019). *Extreme Programming*. IONOS Digitalguide. <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/extreme-programming/>
- ISO IEC 12207. (2008). *Systems and Software Engineering Software Life Cycle Processes*.
- Luna, N. (2018, febrero 26). *¿Qué son las TICs?* Entrepreneur. <https://www.entrepreneur.com/article/308917>
- Madayeliperezhernandez309. (2017, noviembre 15). Ventajas y desventajas de MySQL, oracle, visual foxpro y access. *DISEÑA BASES DE DATOS OFIMATICAS*. <https://mape309site.wordpress.com/2017/11/15/ventajas-y-desventajas-de-mysql-oracle-visual-foxpro-y-access/>
- Moreno, B. (2018). *Ciclo de vida del software—Curso de Fundamentos de Pruebas de Software*. Platzi.Com. <https://https://platzi.com/clases/1421-pruebas-software/15303-ciclo-de-vida-del-software/>
- MySQL: ¿cómo funciona este sistema de gestión de base de datos? (2019, octubre 8). *TIC Portal*. <https://www.ticportal.es/glosario-tic/mysql>

oblancarte. (2018, julio 17). Que es Spring Boot y su relación con los microservicios. *Oscar*

*Blancarte - Software Architecture.*

<https://www.oscarblancarteblog.com/2018/07/17/spring-boot-relacion-los-microservicios/>

¿Que es Angular y para que sirve? (2017, agosto 5). *Victor Robles.*

<https://victorroblesweb.es/2017/08/05/que-es-angular-y-para-que-sirve/>

Qué es el Framework Spring y las ventajas de utilizarlo. (2019, diciembre 15). ▷ *Cursos de*

*Programación de 0 a Experto @ Garantizados.* <https://unipython.com/que-es-el-framework-spring-y-las-ventajas-de-utilizarlo/>

¿Qué son las APIs REST? (2018, febrero 16). *Tribalyte Technologies.*

<https://tech.tribalyte.eu/blog-que-es-una-api-rest>

Ramírez Vique, R. (2016). *Métodos para el desarrollo de aplicaciones móviles.*

[https://www.exabyteinformatica.com/uoc/Informatica/Tecnologia\\_y\\_desarrollo\\_en\\_dispositivos\\_moviles/Tecnologia\\_y\\_desarrollo\\_en\\_dispositivos\\_moviles\\_\(Modulo\\_4\).pdf](https://www.exabyteinformatica.com/uoc/Informatica/Tecnologia_y_desarrollo_en_dispositivos_moviles/Tecnologia_y_desarrollo_en_dispositivos_moviles_(Modulo_4).pdf)

skrapy95. (21:44:28 UTC). *Power designer-presentación* [Presentaciones y charlas públicas].

<https://es.slideshare.net/skrapy95/power-designerpresentacin>

Sommerville, I. (2005). *Ingeniería del software.* Pearson Educación.

Vaquero, D. (2019, noviembre 9). Arquitectura Cliente Servidor. *Cursos de Desarrollo.*

<https://cursosdedesarrollo.com/2019/11/arquitectura-cliente-servidor/>

*Ventajas de utilizar Angular, un framework JavaScript.* (2018, abril 10). El Blog de Aitana –

Partner Microsoft y Sage en España. <https://blog.aitana.es/2018/04/10/ventajas-de-utilizar-angular/>

*Visual Studio Code: Características principales.* (2018, abril 6). Damián De Luca.

<https://damiandeluca.com.ar/visual-studio-code-caracteristicas-principales>

Watson, T. (2019). *Cordova vs Ionic—What Framework to Choose For Your App*. Skywell Software. <https://skywell.software/blog/cordova-vs-ionic-frameworks/>