

Tera Ed.  
2024

 **PUERTO MADERO  
EDITORIAL**

# **METODOLOGÍA DWEP APLICADA AL SISTEMA DE ANÁLISIS DE ALERTAS TEMPRANAS FITOSANITARIAS**



**PAMELA ALEXANDRA BUÑAY GUIÑAN  
GUIDO JAVIER MAZÓN FIERRO  
RENÉ DAVID CARRANZA GUIJARRO  
DANNY JOEL BUCAY SHUCAD**



[puertomaderoeditorial.com.ar](http://puertomaderoeditorial.com.ar)

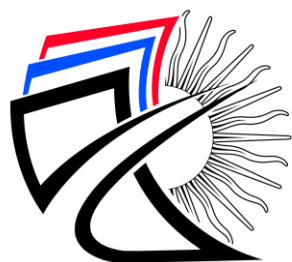


La Plata - Argentina



**METODOLOGÍA DWEP  
APLICADA AL SISTEMA DE  
ANÁLISIS DE ALERTAS  
TEMPRANAS  
FITOSANITARIAS**





**PUERTO MADERO  
EDITORIAL**

# **METODOLOGÍA DWEP APLICADA AL SISTEMA DE ANÁLISIS DE ALERTAS TEMPRANAS FITOSANITARIAS**

**DWEP METHODOLOGY APPLIED TO THE PHYTOSANITARY EARLY  
WARNING ALERT SYSTEM**

**AUTORES:**

Pamela Alexandra Buñay Guisñan  
Guido Javier Mazón Fierro  
René David Carranza Guijarro  
Danny Joel Bucay Shucad





Metodología DWEP aplicada al sistema de análisis de alertas tempranas fitosanitarias /  
Pamela Alexandra Buñay Guisñan ... [et al.] ; Editado por Juan Carlos Santillán Lima  
;  
Fernando Tiverio Molina Granja. - 1a ed - La Plata : Puerto Madero Editorial  
Académica,  
2024.  
Libro digital, PDF/A

Archivo Digital: descarga y online  
ISBN 978-631-6557-45-2

1. Computación. I. Buñay Guisñan, Pamela Alexandra II. Santillán Lima, Juan Carlos,  
ed. III. Molina Granja, Fernando Tiverio, ed.  
CDD 004.02



**Licencia Creative Commons:**

Atribución-NoComercial-SinDerivar 4.0 Internacional (CC BY-NC-SA 4.0)





Primera Edición, Septiembre 2024

METODOLOGÍA DWEP APLICADA AL SISTEMA DE ANÁLISIS DE ALERTAS TEMPRANAS  
FITOSANITARIAS

ISBN: 978-631-6557-45-2

**Editado por:**

**Sello editorial:** ©Puerto Madero Editorial Académica  
**Nº de Alta:** 933832

**Editorial:** © Puerto Madero Editorial Académica

**CUIL:** 20630333971

Calle 45 N491 entre 4 y 5

Dirección de Publicaciones Científicas Puerto Madero Editorial

Académica

La Plata, Buenos Aires, Argentina

**Teléfono:** +54 9 221 314 5902

+54 9 221 531 5142

**Código Postal:** AR1900

**Este libro se sometió a arbitraje bajo el sistema de doble ciego (peer review)**

**Corrección y diseño:**

Puerto Madero Editorial Académica

Diseñador Gráfico: José Luis Santillán Lima

**Diseño, Montaje y Producción Editorial:**

Puerto Madero Editorial Académica

Diseñador Gráfico: Santillán Lima, José Luis

**Director del equipo editorial:** Santillán Lima, Juan Carlos

**Editor:** Santillán Lima, Juan Carlos

Hecho en Argentina

Made in Argentina



## AUTORES:

### ***Pamela Alexandra Buñay Guisñan***

Universidad Nacional de Chimborazo (Unach), Facultad de Ingeniería, ModSim,  
Semillero Aprendizaje inmersivo y Servicios TI, Av. Antonio José de Sucre Km 1 1/2  
vía a Guano, 060, Riobamba, Chimborazo, Ecuador

pbunay@unach.edu.ec,

 <http://orcid.org/0000-0002-4320-6899>

### ***Guido Javier Mazón Fierro***

Escuela Superior Politécnica de Chimborazo (ESPOCH), Facultad de Administración  
de Empresas, Panamericana Sur km 1 ½, 060155, Riobamba, Chimborazo, Ecuador,  
guido.mazon@epoch.edu.ec,

 <https://orcid.org/0000-0001-8745-2373>

### ***René David Carranza Guijarro***

Universidad Nacional de Chimborazo, Facultad de Ingeniería, Avenida Alfonso  
Chávez, entre Ignacio de Veintimilla y Placido Caamaño, Pje. 1. Barrio Madrid,  
Manzana C, Casa N°6., 060108, Riobamba, Chimborazo, Ecuador

rene.carranza@unach.edu.ec,

 <https://orcid.org/0000-0002-3022-8378>

### ***Danny Joel Bucay Shucad***

Universidad Nacional de Chimborazo, Facultad de Ingeniería, Av. Ecuador y Pasaje  
San Jose de Macaji., 060106, Riobamba, Chimborazo, Ecuador

danny.bucay@unach.edu.ec,

 <https://orcid.org/0009-0008-9187-9845>



## CONTENIDO

<b>CONTENIDO</b>	<b>xii</b>
<b>ÍNDICE DE FIGURAS</b>	<b>xv</b>
<b>ÍNDICE DE TABLAS</b>	<b>xviii</b>
<b>RESUMEN</b>	<b>1</b>
<b>ABSTRACT</b>	<b>2</b>
<b>INTRODUCCIÓN</b>	<b>3</b>
<b>CAPÍTULO 1.</b>	<b>4</b>
<b>1 DATA WAREHOUSE.</b>	<b>4</b>
<b>1.1 Concepto</b>	<b>4</b>
<b>1.2 Características de un almacén de datos</b>	<b>4</b>
<b>1.3 Objetivos de un almacén de datos</b>	<b>5</b>
<b>1.4 Funciones de un almacén de datos</b>	<b>7</b>
<b>1.5 Almacén de datos frente a base de datos</b>	<b>7</b>
<b>1.6 Ventajas</b>	<b>8</b>
<b>1.7 Desventajas</b>	<b>8</b>
<b>1.8 Arquitectura</b>	<b>9</b>
<b>1.9 ETL</b>	<b>10</b>
1.9.1 Extracción	10
1.9.2 Limpieza	11
1.9.3 Causas de los datos sucios:	11
1.9.4 Transformación	11
1.9.5 Integración	11
1.9.6 Objetivos ETL	11
1.9.7 Procesamiento analítico en línea relacional	12
1.9.8 Modelo Dimensional	13
1.9.9 Constelación de Hechos	13
<b>1.10 Tablas</b>	<b>15</b>
1.10.1 Hechos	15
1.10.2 Dimensiones	15
<b>CAPITULO 2.</b>	<b>16</b>
<b>2 METDOLOGÍAS DE UN DATA WAREHOUSE</b>	<b>16</b>
<b>2.1 Metodologías de implementación de un data warehouse</b>	<b>16</b>
<b>2.2 Metodología Hefesto</b>	<b>16</b>
<b>2.3 Metodología Inmon</b>	<b>17</b>

<b>2.4</b>	<b>Metodología Kimball</b>	<b>18</b>
<b>2.5</b>	<b>Metodología DWEP</b>	<b>18</b>
<b>2.6</b>	<b>Metodología Data Warehouse Engineering Process (DWEP)</b>	<b>19</b>
<b>2.7</b>	<b>Ventajas y desventajas</b>	<b>20</b>
<b>2.8</b>	<b>Fases de desarrollo</b>	<b>21</b>
2.8.1	Flujos de trabajo	21
<b>CAPITULO 3.</b>		<b>26</b>
<b>3</b>	<b>APLICACIÓN METODOLOGÍA DWEP</b>	<b>26</b>
<b>3.1</b>	<b>Fase de Inicio</b>	<b>26</b>
	Requisitos	26
	Requerimientos Funcionales	26
	<b>Requerimientos No Funcionales</b>	<b>28</b>
	Diagrama de Actividades	29
	Diagrama de Caso de uso	31
	Especificaciones de Caso de uso del Diagrama General	33
	Especificaciones de Caso de uso del Diagrama Estaciones	51
	Especificaciones de Caso de uso del Diagrama Severidad	70
<b>3.2</b>	<b>Análisis</b>	<b>71</b>
	Esquema Conceptual de las Fuentes de Datos	71
	Esquema Lógico de las Fuentes de Datos	73
	Diccionario de Datos	73
	Diccionario de Datos	77
	Esquema Físico de las Fuentes de Datos	80
	Reglas de Negocio	80
<b>3.3</b>	<b>Fase de Elaboración</b>	<b>85</b>
	Diseño	85
	<b>Esquema Conceptual del Data Warehouse</b>	<b>85</b>
	Dimensiones	89
	Dim_Estudio	89
	Dim_Finca	89
	Dim_Plantas	91
	Dim_Tiempo	91
	Dim_Estacion	92
	Hechos	93
	Fact_Severidad	93
	Hechos y Dimensiones	95
	Diagrama de Mapeo de Datos	95
<b>3.4</b>	<b>Implementación</b>	<b>101</b>
	<b>Esquema lógico del Data Warehouse</b>	<b>101</b>
	<b>Esquema Físico del Data Warehouse</b>	<b>102</b>
	Proceso ETL	104
	Proceso ETL Dimensiones	110
	Proceso ETL Hechos	115
<b>3.5</b>	<b>Fase de Construcción</b>	<b>118</b>

Implementación	118
<b>Diagrama de Casos de Uso</b>	<b>118</b>
<b>Diagrama de Clases</b>	<b>127</b>
<b>Diagrama de Objetos</b>	<b>128</b>
<b>Diagrama de Secuencia</b>	<b>130</b>
<b>Diagrama de Colaboración</b>	<b>130</b>
<b>Diagrama de Actividades</b>	<b>131</b>
<b>Diagrama de Componentes</b>	<b>133</b>
<b>3.6 Pruebas</b>	<b>133</b>
Planeación	133
Diseño e Implementación	133
ETL	133
Incidencia	191
Severidad	214
<b>3.7 Evaluación de los Resultados</b>	<b>250</b>
ETL	250
<b>Front End</b>	<b>251</b>
Back End	255
<b>3.8 Fase de Transición</b>	<b>258</b>
Despliegue	258
Instalación de Pentaho Data Integration	258
Despliegue del Sitio Web	259
<b>3.9 Mantenimiento</b>	<b>266</b>
<b>Referencias</b>	<b>267</b>
<b>DE LOS AUTORES</b>	<b>271</b>
<i>Pamela Alexandra Buñay Guisñan.</i>	<b>271</b>
<i>Guido Javier Mazón Fierro</i>	<b>272</b>
<i>René David Carranza Guijarro</i>	<b>273</b>
<i>Danny Joel Bucay Shucad</i>	<b>274</b>

## ÍNDICE DE FIGURAS

<b>Figura 1.</b> Diagrama de actividades .....	10
<b>Figura 2.</b> Proceso ETL.....	12
<b>Figura 3.</b> Modelos Dimensionales.....	13
<b>Figura 4.</b> Tabla de Hecho. ....	15
<b>Figura 5.</b> Tabla de Hecho. ....	15
<b>Figura 6.</b> Metodología DWEP.....	20
<b>Figura 7:</b> Diagrama de Actividades DWEP .....	29
<b>Figura 8:</b> Diagrama de Casos de uso General - Análisis Descriptivo Relación Severidad, Incidencia y Elementos del Clima .....	31
<b>Figura 9:</b> Diagrama de Casos de uso Estaciones – Análisis Descriptivo Relación Severidad y Elementos del Clima en función de las Estaciones .....	32
<b>Figura 10:</b> Diagrama de Casos de uso Severidad– Análisis Descriptivo Severidad en función de las Fincas y Estudios.....	32
<b>Figura 11:</b> Modelo Relacional - Estudios.....	71
<b>Figura 12:</b> Esquema conceptual de la fuente de datos – Base De Datos Relacional Estudios .....	72
<b>Figura 13:</b> Modelo Relacional - Estaciones .....	72
<b>Figura 14:</b> Esquema conceptual de la fuente de datos – Base de Datos Relacional Estaciones .....	73
<b>Figura 15:</b> Esquema lógico de la fuente de datos – Base de Datos Relacional Estaciones .....	73
<b>Figura 16:</b> Esquema lógico de la fuente de datos – Base de Datos Relacional Estaciones. ....	77
<b>Figura 17:</b> Esquema físico de la fuente de datos – Base de Datos Relacional Estaciones .....	80
<b>Figura 18:</b> Esquema físico de la fuente de datos – Base de Datos Relacional Estaciones. .....	80
<b>Figura 19:</b> Esquema conceptual – Nivel 1 .....	86
<b>Figura 20:</b> Esquema conceptual – Nivel 2 .....	86
<b>Figura 21:</b> Esquema conceptual – Nivel 3 .....	88
<b>Figura 22:</b> Dimensiones vs. Hechos .....	95

<b>Figura 23:</b> Mapeo nivel de base de datos – Nivel 0 .....	96
<b>Figura 24:</b> Mapeo nivel de flujo de datos – Nivel 1 .....	96
<b>Figura 25:</b> Mapeo dimensión estudio – Nivel 2 .....	97
<b>Figura 26:</b> Mapeo dimensión finca – Nivel 2 .....	97
<b>Figura 27:</b> Mapeo dimensión planta – Nivel 2 .....	97
<b>Figura 28:</b> Mapeo dimensión estación – Nivel 2 .....	97
<b>Figura 29:</b> Mapeo hecho severidad – Nivel 2 .....	97
<b>Figura 30:</b> Mapeo hecho clima – Nivel 2 .....	98
<b>Figura 31:</b> Mapeo dimensión estudio – Nivel 3 .....	98
<b>Figura 32:</b> Mapeo dimensión finca – Nivel 3 .....	99
<b>Figura 33:</b> Mapeo dimensión planta – Nivel 3 .....	99
<b>Figura 34:</b> Mapeo dimensión estación – Nivel 3 .....	100
<b>Figura 35:</b> Mapeo hecho severidad – Nivel 3 .....	100
<b>Figura 36:</b> Mapeo hecho clima – Nivel 3 .....	101
<b>Figura 37:</b> Esquema lógico Data Warehouse .....	102
<b>Figura 38:</b> Esquema físico - componentes del Data Warehouse .....	103
<b>Figura 39:</b> Esquema físico - despliegue del Data Warehouse .....	104
<b>Figura 40:</b> Staging area. ....	104
<b>Figura 41:</b> Tmp_Estudios – Staging area. ....	105
<b>Figura 42:</b> Tmp_Fincas – Staging area. ....	105
<b>Figura 43:</b> Tmp_Plantas – Staging area. ....	106
<b>Figura 44:</b> Tmp_Monitoreos – Staging area. ....	106
<b>Figura 45:</b> Tmp_Variedades – Staging area.....	106
<b>Figura 46:</b> Tmp_Finca_Variedad – Staging area. ....	107
<b>Figura 47:</b> Tmp_Datos – Staging area.....	107
<b>Figura 48:</b> Tmp_Estaciones – Staging area.....	108
<b>Figura 49:</b> Tmp_Estacion_1 – Staging area. ....	108
<b>Figura 50:</b> Tmp_Estacion_2 – Staging area. ....	109
<b>Figura 51:</b> Tmp_Estacion_3 – Staging area. ....	109
<b>Figura 52:</b> Tmp_Estacion_4 – Staging area. ....	110
<b>Figura 53:</b> Dimensiones. ....	110
<b>Figura 54:</b> Dim_Estudios – ETL Dimensiones. ....	111

<b>Figura 55:</b> Dim_Finca – ETL Dimensiones. ....	111
<b>Figura 56:</b> Dim_Planta – ETL Dimensiones. ....	112
<b>Figura 57:</b> Dim_Estacion – ETL Dimensiones. ....	112
<b>Figura 58:</b> Dim_Tiempo – ETL Dimensiones.....	113
<b>Figura 59:</b> Hechos. ....	115
<b>Figura 60:</b> Proceso ETL – Fact_Severidad .....	115
<b>Figura 61:</b> Proceso ETL – Fact_Severidad .....	117
<b>Figura 62:</b> Caso de uso Sitio Web.....	119
<b>Figura 63:</b> Conexión a la Base de Datos .....	128
<b>Figura 64:</b> Clases Aisladas .....	128
<b>Figura 65:</b> Conexión a Base de Datos .....	129
<b>Figura 66:</b> Clases Aisladas .....	129
<b>Figura 67:</b> Diagrama de Secuencia.....	130
<b>Figura 68:</b> Diagrama de Colaboración .....	130
<b>Figura 69:</b> Diagrama de Actividades.....	131
<b>Figura 70:</b> Diagrama de Estados .....	132
<b>Figura 71:</b> Diagrama de Componentes.....	133

## ÍNDICE DE TABLAS

<b>Tabla 1:</b> Diferencias entre el almacén de datos y la base de datos.....	7
<b>Tabla 2:</b> Descripción del Proceso ETL.....	12
<b>Tabla 3:</b> Ventajas y Desventajas .....	12
<b>Tabla 4:</b> Característica del Modelo Constelación.....	13
<b>Tabla 5:</b> Comparación de las metodologías para la implementación de un Data Warehouse. ....	19
<b>Tabla 6:</b> Ventajas y Desventajas de la metodología DWEP. ....	20
<b>Tabla 7:</b> Fases de Desarrollo - DWEP.....	21
<b>Tabla 8:</b> Fases de Desarrollo - DWEP.....	21
<b>Tabla 9:</b> Actividades de las fases y flujos de trabajo - DWEP.....	24
<b>Tabla 10:</b> Descripción del Dashboard .....	26
<b>Tabla 12:</b> Especificación Caso de Uso Análisis Descriptivo - General .....	33
<b>Tabla 13:</b> Especificación Caso de Uso Severidad - General .....	34
<b>Tabla 14:</b> Especificación Caso de Uso Incidencia - General .....	35
<b>Tabla 15:</b> Especificación Caso de Uso Elementos del Clima - General.....	36
<b>Tabla 16:</b> Especificación Caso de Uso Temperatura - General.....	37
<b>Tabla 17:</b> Especificación Caso de Uso Precipitación - General .....	39
<b>Tabla 18:</b> Especificación Caso de Uso Ráfaga de Viento - General .....	40
<b>Tabla 19:</b> Especificación Caso de Uso Punto de Rocío - General .....	41
<b>Tabla 20:</b> Especificación Caso de Uso Velocidad de Viento - General .....	42
<b>Tabla 21:</b> Especificación Caso de Uso Humedad Relativa - General .....	43
<b>Tabla 22:</b> Especificación Caso de Uso Dirección del Viento - General.....	44
<b>Tabla 23:</b> Especificación Caso de Uso Análisis de la Relación - General .....	45
<b>Tabla 24:</b> Especificación Caso de Uso Periodo - General.....	47
<b>Tabla 25:</b> Especificación Caso de Uso Mensual – General.....	48
<b>Tabla 26:</b> Especificación Caso de Uso Semanal – General.....	49
<b>Tabla 27:</b> Especificación Caso de Uso Diario – General .....	50
<b>Tabla 28:</b> Especificación Caso de Uso Análisis Descriptivo - Estaciones.....	51
<b>Tabla 29:</b> Especificación Caso de Uso Estaciones - Estaciones.....	53
<b>Tabla 30:</b> Especificación Caso de Uso General - Estaciones .....	54
<b>Tabla 30:</b> Especificación Caso de Uso Severidad - Estaciones.....	55

<b>Tabla 31:</b> Especificación Caso de Uso Elementos del Clima - Estaciones .....	57
<b>Tabla 32:</b> Especificación Caso de Uso Temperatura - Estaciones .....	58
<b>Tabla 33:</b> Especificación Caso de Uso Precipitación - Estaciones.....	59
<b>Tabla 34:</b> Especificación Caso de Uso Ráfaga de Viento - Estaciones.....	61
<b>Tabla 35:</b> Especificación Caso de Uso Punto de Rocío - Estaciones .....	62
<b>Tabla 36:</b> Especificación Caso de Uso Velocidad de Viento - Estaciones .....	63
<b>Tabla 37:</b> Especificación Caso de Uso Humedad Relativa - Estaciones.....	64
<b>Tabla 38:</b> Especificación Caso de Uso Dirección del Viento - Estaciones .....	65
<b>Tabla 39:</b> Especificación Caso de Uso Análisis de Comportamiento (Severidad) - Estaciones .....	66
<b>Tabla 40:</b> Especificación Caso de Uso Monitoreos - Estaciones .....	67
<b>Tabla 41:</b> Especificación Caso de Uso Análisis de Comportamiento (Elementos del Clima) - Estaciones.....	69
<b>Tabla 42:</b> Especificación Caso de Uso Análisis Descriptivo Severidad - Severidad....	70
<b>Tabla 43:</b> Descripción de la tabla fincas .....	73
<b>Tabla 44:</b> Descripción de la tabla variedades.....	74
<b>Tabla 45:</b> Descripción de la tabla estudios .....	74
<b>Tabla 46:</b> Descripción de la tabla monitoreos .....	75
<b>Tabla 47:</b> Descripción de la tabla plantas.....	75
<b>Tabla 48:</b> Descripción de la tabla datos.....	76
<b>Tabla 49:</b> Descripción de la tabla estacion_iniap.....	77
<b>Tabla 50:</b> Descripción de la tabla estacion_loreto.....	78
<b>Tabla 51:</b> Descripción de la tabla estacion_sacha .....	78
<b>Tabla 52:</b> Descripción de la tabla estacion_labelleza.....	79
<b>Tabla 53:</b> Descripción de la tabla datos_estaciones .....	79
<b>Tabla 54:</b> Análisis de la calidad de los datos.....	82
<b>Tabla 55:</b> Representación UML de los estereotipos de la metodología DWEP.....	85
<b>Tabla 56:</b> Simbología de los atributos de los hechos y dimensiones .....	87
<b>Tabla 57:</b> Jerarquía de la dimensión estudio .....	89
<b>Tabla 58:</b> Jerarquía de la dimensión finca.....	90
<b>Tabla 59:</b> Jerarquía de la dimensión planta .....	91
<b>Tabla 60:</b> Jerarquía de la dimensión tiempo.....	91

<b>Tabla 61:</b> Jerarquía de la dimensión estaciones .....	92
<b>Tabla 62:</b> Granularidad del hecho severidad .....	93
<b>Tabla 63:</b> Medidas del hecho severidad .....	94
<b>Tabla 64:</b> Granularidad del hecho clima .....	94
<b>Tabla 65:</b> Medidas del hecho clima.....	94
<b>Tabla 66:</b> Resumen tmp_estudios.....	105
<b>Tabla 67:</b> Resumen tmp_fincas .....	105
<b>Tabla 68:</b> Resumen tmp_plantas .....	106
<b>Tabla 69:</b> Resumen tmp_monitoreos.....	106
<b>Tabla 70:</b> Resumen tmp_variedades .....	106
<b>Tabla 71:</b> Resumen tmp_finca_variedad .....	107
<b>Tabla 72:</b> Resumen tmp_datos .....	107
<b>Tabla 73:</b> Resumen tmp_estaciones .....	108
<b>Tabla 74:</b> Resumen tmp_estacion_1 .....	108
<b>Tabla 75:</b> Resumen tmp_estacion_2.....	109
<b>Tabla 76:</b> Resumen tmp_estacion_3.....	109
<b>Tabla 77:</b> Resumen tmp_estacion_3.....	110
<b>Tabla 78:</b> Resumen ETL dim_estudios .....	111
<b>Tabla 79:</b> Resumen ETL dim_finca .....	111
<b>Tabla 80:</b> Resumen ETL dim_planta.....	112
<b>Tabla 81:</b> Resumen ETL dim_estacion .....	112
<b>Tabla 82:</b> Resumen ETL dim_tiempo .....	113
<b>Tabla 83:</b> Resumen ETL Fact_Severidad.....	115
<b>Tabla 84:</b> Resumen ETL Fact_Clima .....	117
<b>Tabla 85:</b> Especificación Caso de Uso Ingresar al Sistema – Sitio Web .....	119
<b>Tabla 86:</b> Especificación Caso de Uso Estaciones – Sitio Web.....	120
<b>Tabla 87:</b> Especificación Caso de Uso Estación 1 – Sitio Web .....	121
<b>Tabla 88:</b> Especificación Caso de Uso Modulo General – Sitio Web.....	122
<b>Tabla 89:</b> Especificación Caso de Uso Severidad – Sitio Web .....	123
<b>Tabla 90:</b> Especificación Caso de Uso General – Sitio Web .....	124
<b>Tabla 91:</b> Especificación Caso de Uso Finca – Sitio Web .....	125
<b>Tabla 92:</b> Especificación Caso de Uso Estudio – Sitio Web.....	126

<b>Tabla 54:</b> Planificación de las pruebas. ....	133
<b>Tabla 94:</b> Caso de Prueba Carga Variedades - Staging.....	133
<b>Tabla 95:</b> Caso de Prueba Carga Estudios - Staging .....	134
<b>Tabla 96:</b> Caso de Prueba Carga Estacion 1 - Staging .....	135
<b>Tabla 97:</b> Caso de Prueba Carga Dim_Estudio - Data Warehouse .....	136
<b>Tabla 98:</b> Caso de Prueba Carga Dim_Finca - Data Warehouse.....	137
<b>Tabla 99:</b> Caso de Prueba Carga Fact_Severidad – Data Warehouse .....	138
<b>Tabla 100:</b> Caso de Prueba Carga Fact_Clima – Data Warehouse .....	140
<b>Tabla 101:</b> Caso de Prueba Sp_Proceso_ETL_1 – Data Warehouse .....	141
<b>Tabla 102:</b> Caso de Prueba Información_Graficos_General.....	142
<b>Tabla 103:</b> Caso de Prueba Información_Graficos_Estaciones .....	143
<b>Tabla 104:</b> Caso de Prueba Elementos_Clima_Estaciones .....	144
<b>Tabla 105:</b> Caso de Prueba Periodo_Existente_Estaciones .....	145
<b>Tabla 106:</b> Caso de Prueba Rango_Fechas_General.....	146
<b>Tabla 107:</b> Caso de Prueba Temperatura_Gráfico_Estaciones .....	147
<b>Tabla 108:</b> Caso de Prueba Get_Estaciones .....	148
<b>Tabla 109:</b> Caso de Prueba Post_CS_Mensual .....	149
<b>Tabla 110:</b> Caso de Prueba Post_CS_Mensual_1 .....	150
<b>Tabla 111:</b> Caso de Prueba Post_CS_Mensual_2 .....	152
<b>Tabla 112:</b> Caso de Prueba Post_CS_Mensual_3 .....	153
<b>Tabla 113:</b> Caso de Prueba Post_CS_Mensual_4 .....	154
<b>Tabla 114:</b> Caso de Prueba Post_CS_Semanal .....	156
<b>Tabla 115:</b> Caso de Prueba Post_CS_Semanal_1 .....	158
<b>Tabla 116:</b> Caso de Prueba Post_CS_Semanal_2 .....	159
<b>Tabla 117:</b> Caso de Prueba Post_CS_Semanal_3 .....	161
<b>Tabla 118:</b> Caso de Prueba Post_CS_Semanal_4 .....	162
<b>Tabla 119:</b> Caso de Prueba Post_CS_Monitoreos.....	164
<b>Tabla 120:</b> Caso de Prueba Post_CS_Monitoreos_1.....	165
<b>Tabla 121:</b> Caso de Prueba Post_CS_Monitoreos_2.....	167
<b>Tabla 122:</b> Caso de Prueba Post_CS_Monitoreos_3.....	168
<b>Tabla 123:</b> Caso de Prueba Post_CS_Monitoreos_4.....	170
<b>Tabla 124:</b> Caso de Prueba Post_CS_Monitoreos_Semanas .....	171

<b>Tabla 125:</b> Caso de Prueba Post_CS_Monitoreos_Semanas_1 .....	173
<b>Tabla 126:</b> Caso de Prueba Post_CS_Monitoreos_Semanas_2 .....	174
<b>Tabla 127:</b> Caso de Prueba Post_CS_Monitoreos_Semanas_3 .....	176
<b>Tabla 128:</b> Caso de Prueba Post_CS_Monitoreos_Semanas_4 .....	177
<b>Tabla 129:</b> Caso de Prueba Post_CS_Diario .....	179
<b>Tabla 130:</b> Caso de Prueba Post_CS_Diario_1 .....	181
<b>Tabla 131:</b> Caso de Prueba Post_CS_Diario_2 .....	182
<b>Tabla 132:</b> Caso de Prueba Post_CS_Diario_3 .....	184
<b>Tabla 133:</b> Caso de Prueba Post_CS_Diario_4 .....	185
<b>Tabla 134:</b> Caso de Prueba Post_CS_Datos .....	187
<b>Tabla 135:</b> Caso de Prueba Post_CS_Datos_1 .....	188
<b>Tabla 136:</b> Caso de Prueba Post_CS_Datos_2 .....	190
<b>Tabla 137:</b> Caso de Prueba Get_Incidencia .....	191
<b>Tabla 138:</b> Caso de Prueba Post_Incidencia_Mensual .....	192
<b>Tabla 139:</b> Caso de Prueba Post_Incidencia_Mensual_1 .....	194
<b>Tabla 140:</b> Caso de Prueba Post_Incidencia_Mensual_2 .....	195
<b>Tabla 141:</b> Caso de Prueba Post_Incidencia_Mensual_3 .....	197
<b>Tabla 142:</b> Caso de Prueba Post_Incidencia_Mensual_4 .....	198
<b>Tabla 143:</b> Caso de Prueba Post_Incidencia_Semanal .....	199
<b>Tabla 144:</b> Caso de Prueba Post_Incidencia_Semanal_1 .....	201
<b>Tabla 145:</b> Caso de Prueba Post_Incidencia_Semanal_2 .....	202
<b>Tabla 146:</b> Caso de Prueba Post_Incidencia_Semanal_3 .....	204
<b>Tabla 147:</b> Caso de Prueba Post_Incidencia_Semanal_4 .....	205
<b>Tabla 148:</b> Caso de Prueba Post_Incidencia_Diario .....	207
<b>Tabla 149:</b> Caso de Prueba Post_Incidencia_Diario_1 .....	208
<b>Tabla 150:</b> Caso de Prueba Post_Incidencia_Diario_2 .....	210
<b>Tabla 151:</b> Caso de Prueba Post_Incidencia_Diario_3 .....	211
<b>Tabla 152:</b> Caso de Prueba Post_Incidencia_Diario_4 .....	213
<b>Tabla 153:</b> Caso de Prueba Get_Severidad .....	214
<b>Tabla 154:</b> Caso de Prueba Post_Severidad_Mensual .....	215
<b>Tabla 155:</b> Caso de Prueba Post_Severidad_Mensual_1 .....	217
<b>Tabla 156:</b> Caso de Prueba Post_Severidad_Mensual_2 .....	218

<b>Tabla 157:</b> Caso de Prueba Post_Severidad_Mensual_3 .....	219
<b>Tabla 158:</b> Caso de Prueba Post_Severidad_Anual .....	220
<b>Tabla 159:</b> Caso de Prueba Post_Severidad_Anual_1 .....	221
<b>Tabla 160:</b> Caso de Prueba Post_Severidad_Anual_2 .....	222
<b>Tabla 161:</b> Caso de Prueba Post_Severidad_Anual_3 .....	223
<b>Tabla 162:</b> Caso de Prueba Post_Severidad_Fincas.....	224
<b>Tabla 163:</b> Caso de Prueba Post_Severidad_Fincas_1.....	226
<b>Tabla 164:</b> Caso de Prueba Post_Severidad_Fincas_2.....	227
<b>Tabla 165:</b> Caso de Prueba Post_Severidad_Fincas_3.....	228
<b>Tabla 166:</b> Caso de Prueba Post_Severidad_Plantas.....	229
<b>Tabla 167:</b> Caso de Prueba Post_Severidad_Plantas_1.....	230
<b>Tabla 168:</b> Caso de Prueba Post_Severidad_Plantas_2.....	232
<b>Tabla 169:</b> Caso de Prueba Post_Severidad_Plantas_3.....	233
<b>Tabla 170:</b> Caso de Prueba Post_Severidad_Frutos.....	234
<b>Tabla 171:</b> Caso de Prueba Post_Severidad_Frutos_1 .....	236
<b>Tabla 172:</b> Caso de Prueba Post_Severidad_Frutos_2.....	237
<b>Tabla 173:</b> Caso de Prueba Post_Severidad_Plantas_4.....	238
<b>Tabla 174:</b> Caso de Prueba Post_Severidad_Estudio.....	240
<b>Tabla 175:</b> Caso de Prueba Post_Severidad_Estudio_1 .....	241
<b>Tabla 176:</b> Caso de Prueba Post_Severidad_Estudio_2.....	243
<b>Tabla 177:</b> Caso de Prueba Post_Severidad_Estudio_3.....	244
<b>Tabla 178:</b> Caso de Prueba Post_Severidad_Monitoreo .....	245
<b>Tabla 179:</b> Caso de Prueba Post_Severidad_Monitoreo_1 .....	246
<b>Tabla 180:</b> Caso de Prueba Post_Severidad_Monitoreo_2 .....	248
<b>Tabla 181:</b> Caso de Prueba Post_Severidad_Monitoreo_3 .....	249
<b>Tabla 182:</b> Evaluación de los resultados ETL.....	250
<b>Tabla 183:</b> Evaluación de los resultados Front end.....	251
<b>Tabla 184:</b> Evaluación de los resultados Back End.....	255
<b>Tabla 185:</b> Evaluación de resultados general.....	258

## **RESUMEN**

En la provincia de Orellana, el cacao es el cultivo que más hectáreas utiliza junto con el café. Uno de los principales riesgos asociados con el cultivo de este árbol, es la posibilidad de que las enfermedades o plagas ataquen los cultivos, en consecuencia, existen centros meteorológicos y de monitoreo para cultivos generando una gran cantidad de datos, que requieren ser analizados. La presente obra tiene como objetivo indagar la metodología “Data Warehouse Engineering Process”, implementar un Data Warehouse para almacenar y analizar los datos históricos recogidos por los centros meteorológicos y de control de cultivos, con el fin de contribuir con la toma de decisiones para la prevención o disminución del impacto de posibles plagas o enfermedades en la producción.

La metodología DWEP contempla el ciclo de vida completa de un Data Warehouse, desde la obtención de requerimientos hasta el despliegue y mejora del software, generando como resultado de su aplicación un sistema completo, de calidad y bien documentado. Un aspecto importante de la metodología es la generación de esquemas y diagramas mediante el estándar UML, como resultado de la implementación del Data Warehouse para la toma de decisiones fitosanitarias agrícolas, se desarrollaron diferentes diagramas y esquemas, propuestos por la metodología.

**Palabras clave:** Data Warehouse, DWEP, metodología, ETL, propuesta DWEP.

## ABSTRACT

In the province of Orellana, cocoa is the crop that occupies the most hectares along with coffee. One of the main risks associated with the cultivation of this tree is the possibility of diseases or pests attacking the crops. Consequently, there are meteorological and crop monitoring centers that generate a large amount of data requiring analysis. The purpose of this work is to investigate the 'Data Warehouse Engineering Process' methodology, implement a Data Warehouse to store and analyze the historical data collected by meteorological and crop control centers, aiming to contribute to decision-making for the prevention or reduction of the impact of potential pests or diseases on production.

The DWEP methodology encompasses the complete life cycle of a Data Warehouse, from requirements gathering to software deployment and improvement, resulting in a comprehensive, quality, and well-documented system. An important aspect of the methodology is the generation of schemas and diagrams using the UML standard. As a result of the implementation of the Data Warehouse for agricultural phytosanitary decision-making, various diagrams and schemas were developed, as proposed by the methodology.

**Keywords:** Data Warehouse, DWEP, methodology, ETL, DWEP proposal.

## **INTRODUCCIÓN**

En la provincia de Orellana, el cacao es el cultivo que más hectáreas utiliza junto con el café. Uno de los principales riesgos asociados con el cultivo de este árbol, es la posibilidad de que las enfermedades o plagas ataquen los cultivos, en consecuencia, existen centros meteorológicos y de monitoreo para cultivos generando una gran cantidad de datos, que requieren ser analizados.

Un Data Warehouse es una colección de datos enfocada a una sola temática proveniente de diferentes orígenes de datos, con la finalidad de que sean analizados y sirvan de soporte en la toma de decisiones, su importancia radica en la organización y estructuración de los datos, de tal forma que facilite el acceso y la generación de información. (Inmon, 2002)

La metodología de desarrollo de software es importante porque permite organizar y estructurar cada una de las actividades del proceso de desarrollo. La metodología DWEP contempla el ciclo completo de desarrollo de un Data Warehouse, permitiendo crear todos los aspectos fundamentales en los modelos de datos (lógico, conceptual y físico); iterativo e incremental es una de las principales características que contempla esta metodología y se basa en 4 fases de desarrollo (Inicio, Elaboración, Construcción, Transición) y 7 flujos de trabajo (Requisitos, Análisis, Diseño, Implementación, Pruebas, Mantenimiento, Revisión Post Desarrollo). (Luján & Trujillo, 2003)

El proyecto de investigación se realizó en la provincia de Orellana, específicamente en los cantones Francisco de Orellana, Loreto y Joya de los Sachas; enfocado en la implementación de un Data Warehouse, mediante la metodología Data Warehouse Engineering Process (DWEP), que almacene datos históricos de los centros meteorológicos y de control de cultivos, estos datos requieren ser analizados y presentados de manera simplificada para la toma de decisiones, con el fin de prevenir daños y afectaciones en los cultivos de cacao.

## CAPÍTULO 1.

### 1 DATA WAREHOUSE.

#### 1.1 Concepto

Un almacén de datos (Data Warehouse, DW) sirve como la base central donde se almacenan los datos recopilados de sistemas transaccionales y convencionales. Se convierte en una nueva fuente de la cual se pueden aplicar procesos de minería de datos e inteligencia de negocios. Sin embargo, el desarrollo de un Data Warehouse no es una tarea simple; se necesita la orientación de una metodología que dirija cada una de las fases y actividades involucradas en el proceso. El objetivo fundamental es lograr la creación de un almacén de datos coherente. (Caicedo & Tello, 2018)

Los Almacenes de Datos o Data Warehouse (DW), surgieron en la década de 1990, definidos como: una colección de datos orientada a un ámbito específico (empresa u organización), integrada, no volátil y que varía en el tiempo, con el propósito de facilitar el proceso de toma de decisiones en los sistemas de soporte. El diseño y la construcción de Data Warehouse están ganando cada vez más popularidad en las organizaciones, considerando las ventajas de analizar datos históricos de manera multidimensional para respaldar la toma de decisiones. Sin embargo, este proceso se vuelve complejo debido a desafíos en la recolección de requerimientos, el análisis y el diseño, que a menudo no se abordan con la metodología adecuada. (Brizuela & Blanco, 2013)

Un Data Warehouse es un sistema diseñado para recopilar, gestionar datos de diversas fuentes, así como analizar y conectar los datos empresariales. Funciona como el núcleo del sistema de Business Intelligence, permitiendo el análisis de grandes cantidades de información y la generación de informes para consultas y toma de decisiones.

Al desarrollar una solución de Data Warehouse, las organizaciones se encuentran con diversas metodologías entre las cuales elegir, cada una con sus ventajas particulares y algunas limitaciones. Estas características deben ser evaluadas cuidadosamente para seleccionar la metodología que mejor se adapte a los requisitos del proyecto y de la organización. (Mendoza, 2022)

#### 1.2 Características de un almacén de datos

Las siguientes son las cuatro características de un almacén de datos:

- **Orientado al sujeto:** Un almacén de datos se centra en un tema particular y ofrece información sobre ese tema específico en lugar de abordar las operaciones actuales

de una empresa. En otras palabras, el proceso de almacenamiento de datos está mejor preparado para gestionar un área o tema específico. Ejemplos de estos temas incluyen ventas, distribuciones, marketing, entre otros.

- **Integrado:** La integración se define como el establecimiento de una conexión entre una amplia cantidad de datos provenientes de diversas bases de datos o fuentes. No obstante, es igualmente esencial que los datos se almacenen de manera unificada en el almacén de datos. El proceso de almacenamiento de datos integra información de múltiples fuentes, que pueden incluir mainframes, bases de datos relacionales, archivos planos, entre otros. Además, contribuye a mantener la consistencia en códigos, medidas de atributos, convenciones de nomenclatura y formatos.
- **Variante de tiempo:** **La variante de tiempo en un Data Warehouse es más extensa en comparación con otros sistemas operativos. Los datos almacenados en un almacén de datos se recuperan con un período de tiempo específico y proporcionan información desde una perspectiva histórica.**
- **No volátil:** En un almacén de datos no volátiles, los datos son permanentes, lo que significa que, al insertar nuevos datos, no se reemplazan, omiten ni eliminan los datos anteriores. En este tipo de almacén de datos, la información es de solo lectura y solo se actualiza en intervalos específicos. Las dos operaciones fundamentales en el almacén de datos son el acceso a los datos y la carga de datos.

### 1.3 Objetivos de un almacén de datos

1. El Data Warehouse debe facilitar el acceso a la información de la organización. El contenido del Data Warehouse debe ser claro, intuitivo y evidente para los usuarios de negocios. La comprensión implica legibilidad, por lo tanto, el contenido del Data Warehouse debe estar etiquetado de manera significativa. Los usuarios de negocios deben poder extraer porciones del Data Warehouse y combinar esta información de todas las formas posibles, utilizando herramientas simples y fáciles de usar, con un tiempo de respuesta mínimo.

2. El Data Warehouse debe presentar la información de la organización de manera coherente. La información almacenada en el Data Warehouse debe ser confiable. Los datos deben ser recolectados cuidadosamente desde diversas fuentes en toda la organización, deben ser depurados, asegurando su calidad, y estar listos para ser utilizados por los usuarios. La información relacionada con un proceso de negocio debe coincidir con la información de otro proceso dentro del Data Warehouse. Si dos métricas no representan lo mismo, deben recibir nombres distintos. Para garantizar la consistencia, la información debe ser de alta

calidad, lo que significa que todos los datos sean precisos y completos. La consistencia también implica que las definiciones comunes del contenido de un Data Warehouse estén disponibles para todos los usuarios.

3. El Data Warehouse debe ser adaptable y resistente a cambios, ya que resulta imposible evitarlos. Las necesidades de los usuarios, las condiciones del negocio, los datos y la tecnología están sujetos a cambios con el tiempo. El diseño del Data Warehouse debe ser capaz de gestionar estas inevitables modificaciones. Los cambios en el Data Warehouse no deben invalidar los datos ni las aplicaciones existentes, ni deben ser alterados o comprometidos cuando la comunidad de usuarios realice nuevas consultas o se añadan nuevos datos al Data Warehouse. Si se realizan cambios en los datos descriptivos del Data Warehouse, estos deben ser gestionados de manera adecuada.

4. El Data Warehouse debe funcionar como un respaldo que resguarde la información de la organización. La información recopilada de la organización se almacena en el Data Warehouse. Por lo tanto, es probable que el Data Warehouse contenga detalles sensibles, como qué se vendió, a quién y a qué precio, información potencialmente riesgosa en manos indebidas. Es imperativo que el Data Warehouse controle de manera efectiva el acceso a la información confidencial de la organización.

5. El Data Warehouse debe desempeñar un papel fundamental para mejorar la toma de decisiones. Debe contener la información adecuada para respaldar los procesos relacionados con la toma de decisiones, permitiendo así la generación de decisiones basadas en los resultados proporcionados por el Data Warehouse. Estas decisiones contribuyen al negocio generando impacto y valor atribuible al uso del Data Warehouse.

6. El éxito del Data Warehouse se define por la aceptación de la organización. Aunque se haya construido una solución sofisticada utilizando los mejores productos y plataformas disponibles, si los usuarios de la organización no adoptan el Data Warehouse y dejan de utilizarlo después de un tiempo posterior a la capacitación, entonces no se ha superado la prueba de aceptación. A diferencia de las nuevas implementaciones de sistemas operativos, donde los usuarios no tienen otra opción más que utilizar el nuevo sistema, el uso del Data Warehouse a veces es opcional. La aceptación de los usuarios está más relacionada con la simplicidad que con cualquier otro factor.

Esto destaca claramente que la creación de un Data Warehouse no solo requiere un alto nivel de conocimientos técnicos, sino también un entendimiento profundo de los procesos de negocio, un cambio cultural en la organización y una visión muy clara con objetivos bien definidos. (Gutiérrez, 2013)

### 1.4 Funciones de un almacén de datos

El almacén de datos opera como un repositorio que permite a las organizaciones evitar los costos asociados con sistemas de almacenamiento y respaldo de datos a nivel empresarial.

Las funciones clave del almacén de datos incluyen (Astera, 2023):

- Limpieza de datos
- Integración de datos
- Asignación de datos
- Extracción de datos
- Transformación de datos
- Carga de datos
- Actualización de datos

### 1.5 Almacén de datos frente a base de datos

Las principales diferencias entre el almacén de datos y la base de datos se resumen en la siguiente tabla (Astera, 2023):

**Tabla 1:** Diferencias entre el almacén de datos y la base de datos

Base de datos	Almacenamiento de Datos
Una base de datos es una mezcla de datos interrelacionados.	El almacén de datos funciona como un sistema de información que alberga datos históricos y cambiantes provenientes de una o varias fuentes.
Una base de datos se emplea para registrar y almacenar datos.	Un almacén de datos se utiliza para analizar datos.
Una base de datos es un conjunto de datos orientado a aplicaciones.	El Data Warehouse es la agrupación de datos centrada en el sujeto.
Una base de datos hace uso de procesamiento transaccional en línea, conocido como OLTP.	El Data Warehouse emplea el procesamiento analítico en línea (OLAP).
Las tablas y las uniones en bases de datos se encuentran normalizadas, lo que las complica.	Las tablas y relaciones en el Data Warehouse no están normalizadas, lo que las hace más sencillas.

---

En el diseño, se emplean técnicas de modelado de Entidad-Relación (ER).

En el diseño, se aplican técnicas de modelado de datos.

---

## **1.6 Ventajas**

Las ventajas son (Bernabeu, 2009):

- Convierte datos provenientes de aplicaciones en información orientada a la toma de decisiones.
- Unifica y consolida diversas fuentes de datos, tanto internas como externas, así como departamentos empresariales previamente aislados, en una plataforma única, sólida y centralizada.
- Proporciona la capacidad de analizar y explorar diferentes áreas de trabajo de forma inmediata.
- Facilita la adaptación rápida a los cambios del mercado.
- Mejora la competitividad en el mercado.
- Elimina la producción y procesamiento de datos innecesarios derivados de aplicaciones mal diseñadas o ya no utilizadas.
- Mejora la entrega de información al proporcionar datos completos, correctos, consistentes, oportunos y accesibles, en el momento adecuado y en el formato apropiado.
- Genera un impacto positivo en los procesos de toma de decisiones al ofrecer a los usuarios acceso a una mejor calidad de información.
- Incrementa la eficiencia de los encargados de tomar decisiones.
- Permite a los usuarios acceder directamente a la información en línea, facilitando la realización de tareas rutinarias y proporcionando una amplia gama de información multidimensional en un formato coherente y confiable.
- Facilita el uso de herramientas familiares a los usuarios, como hojas de cálculo, procesadores de texto, software de análisis de datos, software de análisis estadístico, informes, paneles, etc.
- Facilita la toma de decisiones estratégicas y tácticas.

## **1.7 Desventajas**

Las desventajas son (Bernabeu, 2009):

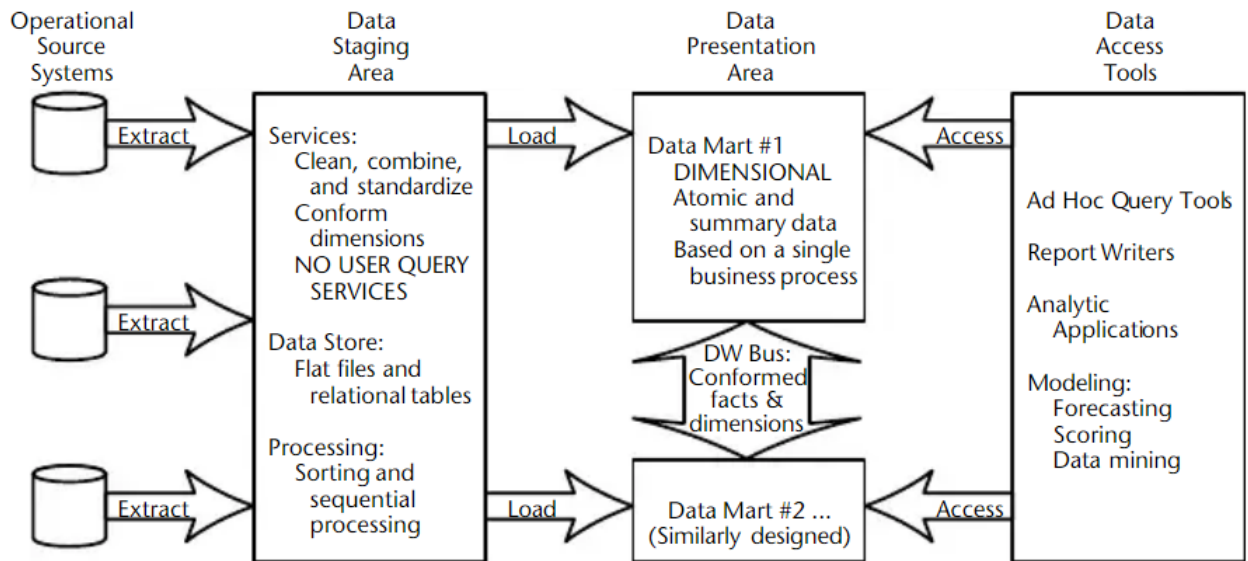
- Implica una inversión significativa, ya que su construcción adecuada no es una tarea fácil y requiere muchos recursos. Además, su implementación abarca desde la adquisición de herramientas de consulta y análisis hasta la formación de los usuarios.
- Hay resistencia al cambio entre los usuarios.

- Los beneficios del Data Warehouse se evidencian a medio y largo plazo.
- Este punto está relacionado con el anterior y, fundamentalmente, indica que no todos los usuarios confiarán en el Data Warehouse inicialmente, pero lo harán después de comprobar su efectividad y ventajas. Además, su correcto uso surge de la experiencia propia.

## 1.8 Arquitectura

La arquitectura de un Data Warehouse tiene los siguientes componentes (Kimball & Ross 2002):

- **Fuentes de Datos de Sistemas Operacionales:** En el marco de la estructura, este elemento abarca las fuentes de información destinadas a alimentar el Data Warehouse, mayormente vinculadas a los sistemas transaccionales empresariales, como, por ejemplo, los sistemas de ventas o la gestión de inventarios. Estas fuentes de datos pueden ser tanto internas como externas, y su almacenamiento puede ser gestionado mediante sistemas de archivos o bases de datos.
- **Área de preparación de datos:** Este componente hace mención a los procedimientos ETL y constituye la parte del Data Warehouse destinada a la retención temporal de datos, a la espera de su procesamiento y traslado posterior al almacenamiento principal del Data Warehouse o a los distintos Data Marts.
- **Área de presentación de datos:** En este componente, los datos procedentes de diversas fuentes se almacenan conforme a la estructura definida para el Data Warehouse. Estos datos están accesibles y a la vista para su consulta, facilitando la realización de diversos análisis y la generación de informes y reportes.
- **Herramientas de Acceso a Datos:** Este componente hace referencia a todas las herramientas que utilizarán los datos almacenados en el Data Warehouse para generar información. Estas herramientas pueden variar desde software básico destinado a formular consultas hasta sistemas sofisticados de minería de datos.



**Figura 1.** Diagrama de actividades

**Fuente:** (Kimball & Ross, 2002)

## 1.9 ETL

Facilita la extracción de datos desde el entorno de origen, adapta dichos datos según los requisitos comerciales de integración de datos y los carga en el entorno de destino. Los entornos de origen y destino suelen ser bases de datos y/o archivos, aunque ocasionalmente también pueden ser colas de mensajes de algún middleware u otras fuentes con estructuras, semiestructuras o no estructuradas. (Conesa, 2012)

En la realidad, las herramientas ETL realizan el movimiento o transferencia de datos entre los entornos de origen y destino. Además, registran el proceso de transformación de los datos entre estos entornos y almacenan esta información en sus catálogos de metadatos. Intercambian estos metadatos con otras aplicaciones que puedan necesitarlos y administran todos los procesos de ejecución y ETL. (Curto Díaz, 2017)

La ETL, que forma parte del desarrollo del data warehouse, ocupa una porción considerable de todos los procesos, demandando recursos, habilidades, tecnologías y estrategias. Los pasos para extraer datos de fuentes de información al (DW) están divididos en 5 subprocesos:

### 1.9.1 Extracción

Recopila datos de diversas fuentes de información, ya sea físicamente o mediante herramientas ETL. La recuperación de datos se realiza con un lenguaje de programación, y el objetivo principal es extraer solo los datos necesarios del sistema transaccional y prepararlos para el resto del flujo ETL. La principal ventaja de la extracción es su capacidad para obtener información de diversas fuentes de bases de datos, lo que reduce el tiempo

dedicado a descubrir procesos no documentados, minimiza los posibles errores y facilita una mayor flexibilidad. (Esteso, 2021)

### **1.9.2 Limpieza**

Este procedimiento restaura los datos y verifica su calidad al eliminar duplicados, corregir valores incorrectos y rellenar campos nulos. Así, transforma los datos para minimizar los errores de carga, garantizando la obtención de datos limpios y de alta calidad. (Rayón, 2015)

### **1.9.3 Causas de los datos sucios:**

- Ausencia de valores
- Valores encriptados
- Utilización incorrecta de campos
- Reciclaje de la clave principal con un valor previamente empleado
- Identificador no exclusivo
- Integración obsoleta entre sistemas
- Valores predeterminados

### **1.9.4 Transformación**

Los datos de alta calidad son extraídos y estructurados, y la fase de transformación incluye cambios de formato, sustitución de códigos, valores derivados y agregación. En este proceso, se ajustan los detalles, y aunque los datos descriptivos a nivel de líneas de facturación se pueden tener por separado, los datos de ventas semanales o mensuales se almacenan en el Data Warehouse (DW). En la etapa de integración, los datos se incorporan al DW instantáneamente, y se debe verificar que los totales de ventas almacenados sean coherentes con la información del sistema de origen. Es crucial asegurar que el sistema esté desarrollado de manera correcta y proporcione información precisa al usuario.

### **1.9.5 Integración**

El último paso implica la integración en el Data Warehouse (DW) tan pronto como se cargan los datos, y se debe asegurar que los totales de ventas almacenados sean coherentes con la información presente en el sistema de origen. Es esencial verificar que el sistema esté correctamente desarrollado y proporcionar información veraz al usuario.

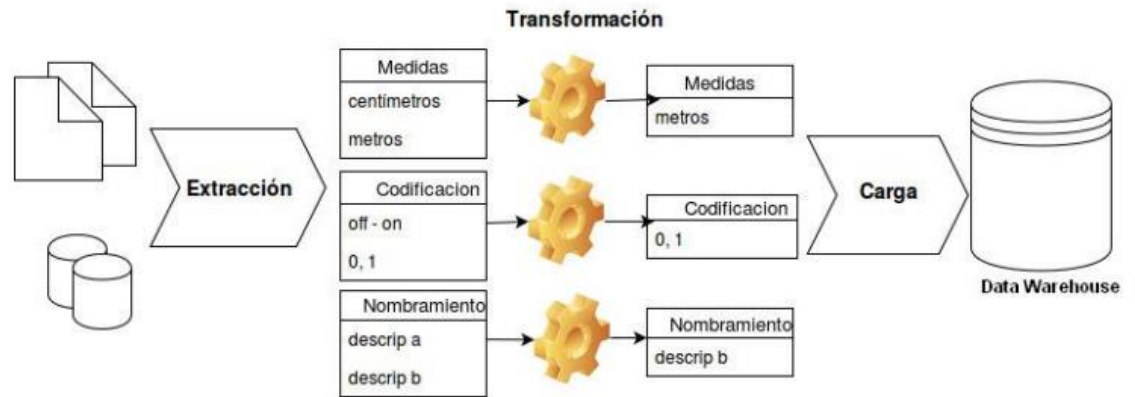
### **1.9.6 Objetivos ETL**

Las operaciones ETL se encuentran en el área de preparación de datos y tienen como objetivo extraer, transformar y cargar datos desde diversas fuentes identificadas. (Rolón, 2021).

Según Becker (2009), los objetivos principales del ETL son los siguientes:

- Corregir inexactitudes, asignar y reemplazar información ausente.

- Preparar datos de diversas fuentes para su uso conjunto.
- Establecer criterios de calidad para los datos.
- Diseñar una organización de datos que facilite su consulta mediante varias herramientas.



**Figura 2.** Proceso ETL

**Fuente:** (Rolón, 2021)

**Tabla 2:** Descripción del Proceso ETL

Extracción	Transformación	Carga
Se extraen los datos desde las diversas fuentes.	En esta etapa, se depura los datos con algún criterio de calidad para eliminar posibles errores en los datos y uniformizar su formato.	Durante esta fase, los datos se introducen en la configuración definida para el Data Warehouse o los distintos Data Marts.

### 1.9.7 Procesamiento analítico en línea relacional

Identificado como ROLAP, este enfoque arquitectónico se concentra en la utilización de Bases de Datos Relacionales para la construcción y modelado del Data Warehouse. Este modelo consta de tres elementos esenciales: la Base de Datos Relacional, el motor OLAP y una herramienta destinada a la presentación de la información (Tamayo & Moreno, 2006).

**Tabla 3:** Ventajas y Desventajas

Ventajas	Desventajas
La estructura está diseñada para gestionar grandes volúmenes de datos.	Al realizar consultas en conjuntos de datos extensos, es posible que experimentemos una disminución en el rendimiento.

La elección de un motor de base de datos relacional permite el almacenamiento eficiente de la información.

ROLAP enfrenta los mismos desafíos vinculados al motor de base de datos relacional que esté en uso y al lenguaje SQL.

Integrar un RDBMS (Sistema Gestor de Bases de Datos Relacionales) con el uso de un motor de base de datos relacional facilita la interacción y administración de la base de datos implementando ROLAP.

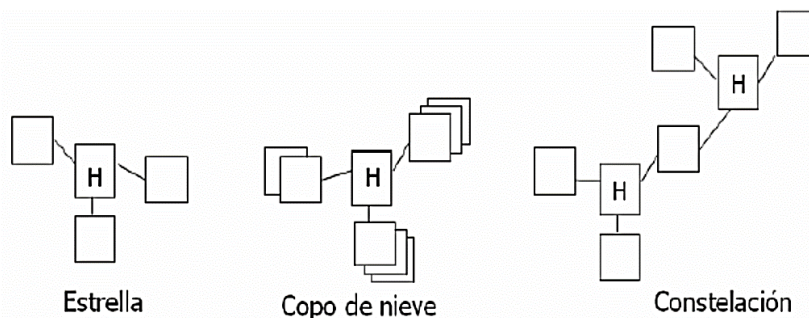
**Fuente:** (Mbaabu, 2021)

### 1.9.8 Modelo Dimensional

Define la disposición en la que los datos serán almacenados en el Data Warehouse, con la finalidad de facilitar la presentación de datos para su transformación en información valiosa mediante consultas realizadas por los usuarios finales. (Jaramillo, 2016)

### 1.9.9 Constelación de Hechos

El modelo adoptado para el proyecto es el de constelación de hechos, ya que se requiere la presencia de dos hechos en el Data Warehouse desarrollado para este proyecto en particular. Esta configuración se logra al combinar los modelos de copo de nieve y estrella, dado que pueden existir en el esquema dos modelos estrella, dos modelos de copo de nieve, o un modelo estrella y uno de copo de nieve, todos compartiendo hechos.



**Figura 3.** Modelos Dimensionales

**Fuente:** (Méndez & Regalado, 2009)

**Tabla 4:** Característica del Modelo Constelación

Características	Descripción
-----------------	-------------

<b>Entendimiento del Modelo</b>	El modelo constelación surge de la combinación de los modelos estrella y copo de nieve, posibilitando que una dimensión esté formada por múltiples tablas y que los hechos compartan dimensiones. Esto añade complejidad al modelo cuando se analiza y comprende. (IBM, 2021)
<b>Número de Tablas</b>	Puede o no contener varias tablas y está directamente vinculado al modelo utilizado para construir cada hecho. En otras palabras, depende de si cada hecho que compone el modelo constelación corresponde a un modelo estrella o a un modelo copo de nieve.
<b>Complejidad de la Consulta</b>	La dificultad de las consultas en este modelo es significativa, pues requiere de numerosas uniones para consultar sobre los hechos y dimensiones de la estructura del modelo.
<b>Desempeño de las Consultas y Procesamiento del Cubo</b>	La estructura del modelo puede intrincada y desafiante de mantener según aumenta el número de tablas relacionadas. Además, el rendimiento de las consultas se verá afectado por la cantidad de tablas vinculadas en el modelo.(Jarrín & Belalcázar, 2014)
<b>Almacenamiento</b>	Dado que se centra en hechos que comparten dimensiones, posibilita la utilización eficiente de tablas presentes en el modelo, generando un ahorro en el espacio de almacenamiento. (Jarrín & Belalcázar, 2014)

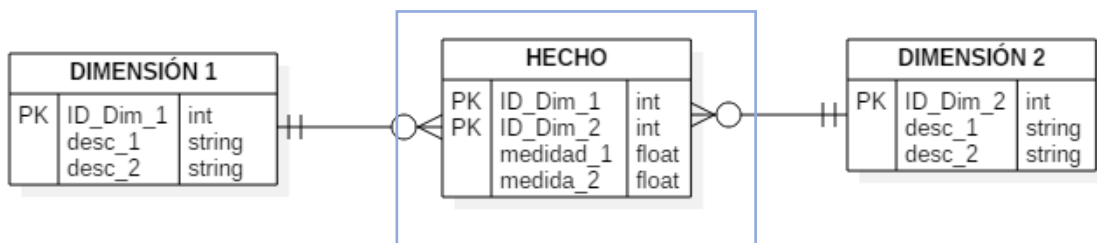
---

**Fuente:** Los Autores

## 1.10 Tablas

### 1.10.1 Hechos

Esta tabla alberga los datos que se pretenden analizar y representa una actividad del negocio, como pedidos, ventas, ingresos, egresos, entre otras. Por lo general, incluye valores numéricos conocidos como medidas, que posibilitan obtener la suma total, cantidad de ventas, costos, precios, etc., relacionados con el hecho o actividad en análisis. Además, esta tabla incorpora claves que establecen conexiones con diversas. (Méndez & Regalado, 2009)



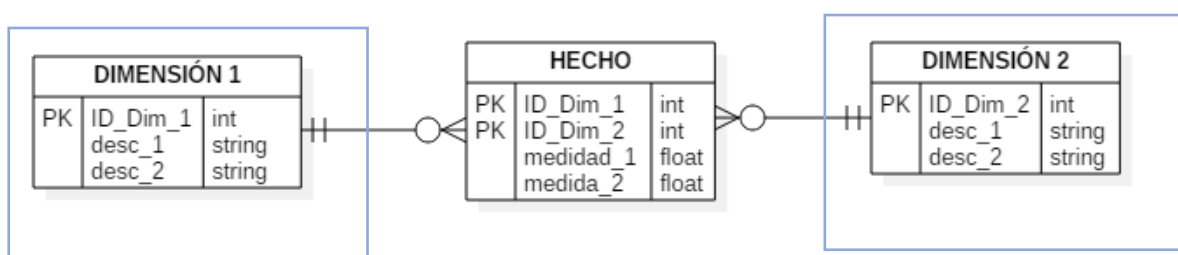
**Figura 4.** Tabla de Hecho.

Otra característica de estas tablas, como se ilustra en la Figura 4, es que tienen una clave primaria formada por claves foráneas de dimensiones.

### 1.10.2 Dimensiones

Las tablas de dimensiones tienen como propósito describir la información contenida en la tabla de hechos, permitiendo que los usuarios finales analicen el contexto del tema focalizado en el Data Warehouse o Data Mart. Por lo general, los valores almacenados en esta tabla son descriptivos, como códigos empresariales, nombres, direcciones, categorías y posiciones geográficas. Además de los valores descriptivos, esta tabla guarda una identificación única generada automáticamente al insertar registros. (Jaramillo, 2016)

La presencia de valores descriptivos en la tabla de dimensión permite filtrar, buscar y manipular los datos almacenados en las tablas de hechos, facilitando la construcción de información útil para la toma de decisiones. (Jaramillo, 2016)



**Figura 5.** Tabla de Hecho.

## CAPITULO 2.

### 2 METDOLOGÍAS DE UN DATA WAREHOUSE

#### 2.1 Metodologías de implementación de un data warehouse

Las metodologías empleadas para llevar a cabo la creación de un Data Warehouse engloban una serie de enfoques, instrumentos y procedimientos destinados a asistir a los programadores en la creación de sistemas de información innovadores. Estas estrategias aseguran tanto la eficiencia como la eficacia durante la edificación de software. Cada metodología consta de fases, las cuales pueden subdividirse en subfases, y presentan atributos distintivos que las distinguen entre sí.(Leonard & Castro, 2013)

La definición de metodología para algunos autores es: una colección de procedimientos, técnicas, herramientas y documentos auxiliares que ayudan a los desarrolladores de software en sus esfuerzos por implementar nuevos sistemas de información. Una metodología está formada por fases, cada una de las cuales se puede dividir en sub-fases, que guiarán a los desarrolladores de sistemas a elegir las técnicas más apropiadas en cada momento del proyecto y también a planificarlo, gestionarlo, controlarlo y evaluarlo.

El propósito fundamental de una metodología de desarrollo consiste en asegurar la eficacia, como, por ejemplo, la satisfacción de los requisitos iniciales, y la eficiencia, como la minimización de pérdidas de tiempo, en el proceso de generar software. Sin embargo, estas metodologías no siempre han sido reconocidas por su éxito, y mucho menos por su popularidad. La crítica más común hacia ellas es que suelen ser burocráticas, repetitivas y ambiguas, llegando a ralentizar todo el ritmo del desarrollo por la cantidad de tareas asociadas con su seguimiento. En la actualidad, se presentan diversas propuestas metodológicas que inciden en diferentes aspectos del proceso de desarrollo de un almacén de datos, con el objetivo de facilitar la ejecución de nuevos proyectos y reducir su complejidad. Cada metodología posee características particulares que las distinguen, por lo que la elección de una o varias de ellas para construir un almacén de datos debe evaluarse cuidadosamente. (Brizuela et al., 2013)

#### 2.2 Metodología Hefesto

La metodología HEFESTO facilita la construcción de Data Warehouse de manera simple, organizada e intuitiva, tomando inspiración de Hefesto, el dios griego de la construcción y el fuego. HEFESTO es una metodología propia que se fundamenta en una extensa investigación, comparación de metodologías existentes y experiencias propias en la creación

de almacenes de datos. El enfoque principal es comprender cada paso del proceso, evitando la rigidez de seguir un método de manera mecánica sin comprender completamente las acciones realizadas. La construcción e implementación del Data Warehouse puede adaptarse a cualquier ciclo de vida de desarrollo de software, aunque algunas fases particulares requerirán acciones específicas. Es crucial evitar metodologías que exijan extensas reuniones de requerimientos y análisis, desarrollo monolítico que consuma demasiado tiempo y despliegues prolongados. La meta es proporcionar una primera implementación que satisfaga parte de las necesidades, demostrando las ventajas del Data Warehouse y motivando a los usuarios. La metodología HEFESTO puede integrarse en cualquier ciclo de vida que cumpla con esta condición. (Peñafiel et. al, 2013)

### **2.3 Metodología Inmon**

Esta metodología proporciona los procedimientos necesarios para llevar a cabo la correcta implementación de un Data Warehouse. Según Bill Inmon, el diseño de un Data Warehouse inicia con los procesos que determinarán cómo se almacenarán los datos en el sistema, ya que se manejarán grandes cantidades de información y la eficiencia en el acceso a los datos depende en gran medida de este proceso. Además, la perspectiva de Inmon respalda uno de los principios esenciales en el desarrollo de un Data Warehouse, que establece que el entorno donde se originan los datos y el entorno de acceso a los datos deben estar físicamente separados en distintas bases de datos y equipos. Por último, los sistemas actuales albergan una enorme cantidad de datos, lo que hace poco realista intentar realizar cargas de información con frecuencia. Si el volumen de datos no se gestiona y condensa de manera cuidadosa, dicho volumen de información obstaculiza el logro de los objetivos del Data Warehouse.

A Bill Inmon se le vincula comúnmente con los Data Warehouse a nivel empresarial, los cuales abarcan todo el ámbito corporativo desde el principio, sin enfocarse en un incremento específico hasta después de completar completamente el diseño del Data Warehouse. Según su filosofía, un DataMart constituye simplemente una de las capas del Data Warehouse y depende del depósito central de datos o Data Warehouse Corporativo, por lo que se construyen posteriormente a este. En el enfoque de Inmon, es imperativo desarrollar una estrategia de Data Warehouse e identificar las áreas principales desde el inicio del proyecto, anticipando así con precisión la estructura que tendrán los núcleos de desarrollo fundamentales en la empresa. Esto garantiza una solución integral y previene la aparición de situaciones imprevistas que podrían poner en riesgo el proyecto. (Gardner, 1998)

## **2.4 Metodología Kimball**

Kimball propone la adopción de un modelo dimensional para Data Marts, utilizando un esquema de estrella, con una arquitectura centrada en áreas de interés y Data Marts, poblando el Data Warehouse con datos detallados y sumariados. Kimball da una base metodológica y empírica adecuada para implementaciones de almacenes de datos, gracias a su versatilidad y a su enfoque progresivo que permite construir los almacenes de datos de forma escalonada, integrando fases y artefactos para obtener mejores resultados. Estas fases incluyen análisis de requerimientos, análisis de sistemas transaccionales, modelo lógico del ETL, procesos ETL, rendimiento y mantenimiento del Data Warehouse.

El enfoque de Kimball se caracteriza por ser una metodología Bottom-up, que consiste en conectar diferentes Data Marts estructurados, crear Data Marts adicionales y luego ir agregando otros que compartan dimensiones establecidas o introduzcan nuevas. (Guerrero & Rodas, 2022).

## **2.5 Metodología DWEP**

La metodología denominada proceso de ingeniería para el desarrollo de almacenes de datos (DWEP) se fundamenta en el Proceso Unificado (PU) y engloba los flujos de trabajo que abarcan desde los requerimientos y análisis hasta el diseño, pruebas, mantenimiento y revisiones posteriores al desarrollo. Sus características clave incluyen ser un enfoque iterativo, orientado por casos de uso y se apoya en las etapas de desarrollo del Proceso Unificado. Además, utiliza UML como lenguaje para el modelado gráfico. (Luján-Mora & Trujillo, 2005)

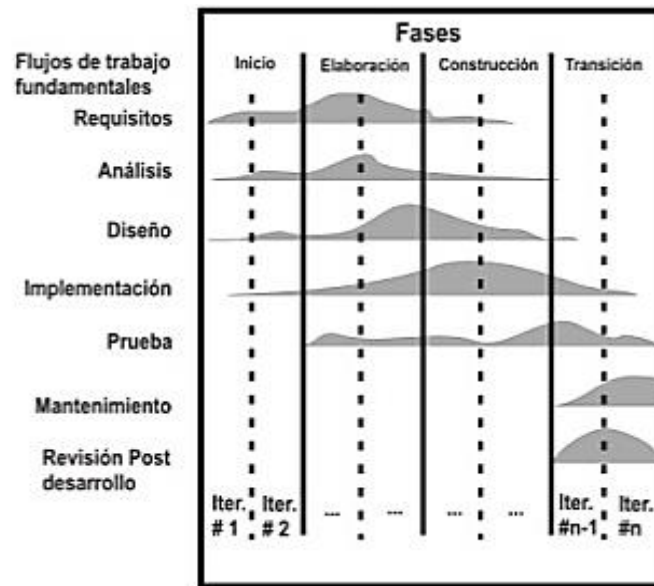
Leonard & Castro (2013), realizan un análisis de diversas metodologías empleadas para el desarrollo de un almacén de datos. Cada una de estas metodologías se adapta a proyectos con características específicas, y se han examinado las cualidades fundamentales empleadas en el desarrollo del Data Warehouse (DW). La conclusión es que las metodologías HEFESTO y DWEP son las más idóneas para la construcción del DW. Esto se debe a que la metodología HEFESTO establece cómo recopilarse los requerimientos y las necesidades de los usuarios para definir la arquitectura de datos, mientras que DWEP detalla la elaboración del modelo conceptual, lógico y físico. (Castro, 2022)

**Tabla 5:** Comparación de las metodologías para la implementación de un Data Warehouse.

<b>Metodologías</b>	<b>Kimball</b> (Cáceres & Cevallos, 2016)	<b>Hefesto</b> (Bustos & Mosquera, 2013)	<b>Inmon</b> (Bustos & Mosquera, 2013)	<b>DWEP</b> (Leonard & Castro, 2013)
Iterativa e Incremental	<b>x</b>	<b>x</b>	<b>x</b>	<b>ü</b>
Contempla la recolección y análisis de requisitos	<b>ü</b>	<b>ü</b>	<b>ü</b>	<b>ü</b>
Contempla la generación de reportes del Data Warehouse.	<b>ü</b>	<b>x</b>	<b>ü</b>	<b>ü</b>
Contempla la facilidad de inserción de cambios en los requisitos.	<b>x</b>	<b>x</b>	<b>x</b>	<b>ü</b>
Es estándar aceptado por la industria	<b>ü</b>	<b>ü</b>	<b>ü</b>	<b>ü</b>

## 2.6 Metodología Data Warehouse Engineering Process (DWEP)

Esta metodología, conocida como DWEP y propuesta por Juan Trujillo y Sergio Lujan-Mora, se orienta al desarrollo tanto del Data Warehouse como del Data Mart. Está fundamentada en UML (Unified Modeling Language) y en la metodología RUP (Rational Unified Process), lo que proporciona a los desarrolladores una guía integral a lo largo de todo el proceso de construcción del Data Warehouse. (Duran, 2017)



**Figura 6.** Metodología DWEP.

*Fuente:* (Leonard & Castro, 2013)

Esta metodología se centra en 4 fases de desarrollo y 7 flujos de trabajo, sus características según Leonard & Castro (2013) son:

- Emplea el Lenguaje Unificado de Modelado (UML).
- Se fundamenta en el Proceso Unificado (UP).
- Representa un estándar reconocido tanto en el ámbito científico como industrial.
- Se caracteriza por ser un enfoque iterativo e incremental.
- Se apoya en el uso de componentes.

## 2.7 Ventajas y desventajas

Las ventajas y desventajas de la metodología DWEP se aprecia en la tabla:

**Tabla 6:** Ventajas y Desventajas de la metodología DWEP.

Ventajas	Desventajas
Es una metodología que promueve buenas prácticas, se fundamenta en la metodología RUP.	Falta de documentación.
Es fácil de comprender gracias a la estructura de etapas dentro de cada fase y flujos de trabajo	La obtención de requisitos establece el fundamento para análisis y diseño subsiguientes, subrayando la importancia de obtener requisitos nítidos y precisos al inicio de la metodología para garantizar el éxito.

Se basa en UML, lo que simplifica la creación de entregables específicos.

Se destaca por su exhaustividad, abarca desde la recopilación de requerimientos hasta la implementación de un Data Warehouse, incluyendo fases críticas del ETL.

---

**Fuente:** (Duran, 2017); (Ochoa, 2017)

## **2.8 Fases de desarrollo**

Dentro de esta metodología existen 4 fases de desarrollo que se detalla a continuación:

**Tabla 7:** Fases de Desarrollo - DWEP.

<b>Fase</b>	<b>De</b>	<b>Descripción</b>
<b>Desarrollo</b>		
Inicio		Examina el proyecto mediante una visión general, identifica posibles riesgos críticos que puedan impactar en el proyecto y se esfuerza por establecer las nuevas funcionalidades fundamentales del software.
Elaboración		Define la base lógica mediante casos de uso y establece los artefactos que se emplearán dentro del sistema.
Construcción		Comienza desde la línea base de arquitectura establecida en la fase de elaboración y tiene como objetivo el desarrollo de un producto listo para su operación inicial en el entorno del usuario final.
Transición		Cuando el proyecto entra en la fase de transición, el sistema ha alcanzado su capacidad operativa inicial. Durante esta etapa, se busca implementar el producto en su entorno de operación.

---

**Fuente:** (Leonard & Castro, 2013)

### **2.8.1 Flujos de trabajo**

Dentro de esta metodología existen 7 flujos de trabajo que se detalla a continuación:

Dentro de esta metodología existen 7 flujos de trabajo que se detalla a continuación:

**Tabla 8:** Fases de Desarrollo - DWEP.

<b>Fase de desarrollo</b>	<b>Descripción</b>
---------------------------	--------------------

---

Requisitos	Se detallan los requisitos, se lleva a cabo el análisis dimensional, se diseña las posibles consultas para la generación de informes y se establece la frecuencia de actualización de los datos. <b>Entregables:</b> <ul style="list-style-type: none"><li>• Diagrama de actividades.</li><li>• Diagrama de casos de uso.</li></ul>
Análisis	Se examinan los requisitos previamente recopilados y se determinan los sistemas que proporcionarán datos al Data Warehouse. <b>Entregables:</b> <ul style="list-style-type: none"><li>• Esquema conceptual de las fuentes de datos.</li><li>• Esquema lógico de las fuentes de datos.</li></ul>
Diseño	Se lleva a cabo la creación del modelo conceptual. <b>Entregables:</b> <ul style="list-style-type: none"><li>• Esquema conceptual del Data Warehouse.</li><li>• Diagrama de mapeo de datos.</li></ul>
Implementación	Se implementa la construcción del Data Warehouse, comenzando a adquirir información desde diversas fuentes de datos. <b>Entregables:</b> <ul style="list-style-type: none"><li>• Esquema lógico del Data Warehouse.</li><li>• Esquema físico del Data Warehouse.</li><li>• Proceso ETL.</li></ul>
Pruebas	Se verifica el rendimiento del sistema a través de pruebas.
Mantenimiento	Se establecen los procedimientos de actualización y carga para mantener el Data Warehouse constantemente actualizado.
Revisión post desarrollo	Se realiza una revisión del sistema para identificar posibles mejoras.

---

**Fuente:** (Leonard & Castro, 2013); (Duran, 2017); (Lujan-Mora & Trujillo, 2005)

- Diagrama de Actividades: Presenta de manera secuencial las actividades y las decisiones necesarias para ejecutar un proceso, que podría ser una actividad empresarial.
- Diagrama de Casos de Uso: Expone los requisitos funcionales o las necesidades identificadas por los usuarios.

- Esquema Conceptual de Datos Fuente: Ilustra de manera conceptual la organización de la estructura que constituye la información de origen para el Datamart.
- Esquema Lógico de Datos Fuente: Presenta de forma lógica la configuración de la estructura de la información de origen para el Datamart.
- Esquema Conceptual del Datamart: Expone de manera conceptual el modelo multidimensional del Datamart.
- Diagrama de Mapeo de Datos: Ilustra la relación entre la estructura de la información de origen y el propio Datamart.
- Esquema Lógico del Datamart: Presenta de forma lógica el modelo multidimensional del Datamart.
- Esquema Físico del Datamart: Muestra de manera física el modelo multidimensional del Datamart, incluyendo la configuración de los servidores y cómo está almacenado en los discos.
- Proceso ETL: Ilustra el proceso de extracción de la información fuente, su transformación y carga en el Datamart, comúnmente conocido como ETL (Extracción, Transformación y Carga). (Zambrano, 2011)

A continuación se detallan las actividades que se realizan tanto en las fases como en los flujos de trabajo:

**Tabla 9:** Actividades de las fases y flujos de trabajo - DWEP.

<b>Fase</b>	<b>Inicio</b>	<b>Elaboración</b>	<b>Construcción</b>
<b>Flujos</b>			
Requisitos	<ul style="list-style-type: none"> <li>• Requisitos funcionales y no funcionales.</li> <li>• Identificación de las medidas, agregaciones y dimensiones.</li> <li>• Análisis de los reportes utilizados actualmente.</li> <li>• Elaboración de los diagramas de casos de uso.</li> </ul>	<ul style="list-style-type: none"> <li>• Revisión y recolección de nuevos requerimientos.</li> <li>• Revisión e identificación de nuevas medidas, agregaciones y dimensiones.</li> <li>• Revisión y elaboración de nuevos casos de uso.</li> </ul>	
Análisis	<ul style="list-style-type: none"> <li>• Determinación de las posibles fuentes de datos.</li> <li>• Elaboración de los diagramas lógico y físico de las fuentes de datos.</li> </ul>	<ul style="list-style-type: none"> <li>• Revisión y selección de fuentes de datos.</li> <li>• Revisión y actualización del diagrama lógico y físico de las fuentes de datos.</li> <li>• Elaboración del diagrama conceptual de las fuentes de datos.</li> </ul>	<ul style="list-style-type: none"> <li>• Actualización tanto del diagrama conceptual, lógico y físico de la fuente de datos.</li> </ul>
Diseño	<ul style="list-style-type: none"> <li>• Definición de la estructura del Data Warehouse.</li> <li>• Definición del diagrama conceptual del Data Warehouse.</li> </ul>	<ul style="list-style-type: none"> <li>• Elaboración del proceso ETL a nivel conceptual.</li> <li>• Actualización del diagrama conceptual.</li> <li>• Elaboración del diagrama mapeo de datos.</li> </ul>	<ul style="list-style-type: none"> <li>• Definición y actualización del proceso ETL a nivel conceptual.</li> </ul>

- 
- Actualización de los diagramas conceptual del Data Warehouse.

---

Implementación

- Elaboración de las estructuras físicas del Data Warehouse.
- Elaboración del diagrama lógico.
- Elaboración del diagrama físico.
- Elaboración del diagrama del proceso ETL.

---

Pruebas

- Planeación de pruebas.
- Diseño de los casos de prueba.
- Realización de las pruebas según casos de pruebas.
- Resultados y correcciones.

---

**Fuente:** (Romero, 2014)

## CAPITULO 3.

### 3 APLICACIÓN METODOLOGÍA DWEP

En este apartado se describe la utilización de la metodología Data Warehouse Engineering Process para la solución del presente proyecto, donde se da a conocer de manera detallada tanto las fases de desarrollo como los flujos de trabajo, en la construcción del Data Warehouse.

#### 3.1 Fase de Inicio

##### Requisitos

Los requerimientos fueron obtenidos en base a las necesidades propuestas, para el componente del Data Warehouse, por parte del director del proyecto PATFA.

Después de realizar reuniones con el director del proyecto PATFA, se logró identificar los siguientes requerimientos:

##### Requerimientos Funcionales

- a) **Desarrollar una herramienta que permita realizar la toma de decisiones a partir del análisis descriptivo de los datos:** Representa la necesidad principal. Permitirá el análisis descriptivo del comportamiento de las tres variables más importantes relacionadas a los cultivos de cacao: severidad, incidencia y elementos del clima (temperatura, precipitación, punto de rocío, ráfaga de viento, humedad relativa y dirección del viento).

Los reportes serán presentados mediante gráficos estadísticos configurables. Los gráficos mencionados tienen el objetivo de representar la siguiente información:

**Tabla 10:** Descripción del Dashboard

Objetivo	Descripción
Relacionar las variables de severidad, incidencia y elementos del clima en el tiempo.	Representa el comportamiento de las variables en un periodo mensual, semanal y diario.
Relacionar las variables de la severidad y elementos del clima, en función de las estaciones o	Representa el comportamiento de las variables en un periodo mensual, quincenal y semanal, clasificadas mediante las

monitoreos, en función de un periodo de tiempo.	estaciones o monitoreos existentes.
Representar la variable de severidad en función de las fincas o estudios, en un periodo de tiempo.	Representa el comportamiento de la variable con un promedio mensual, en un periodo máximo de hasta 5 años, con la utilidad de clasificar en función de las fincas (detalladas en plantas y frutos) o estudios (detallados en monitoreos).

Los gráficos mencionados contarán con las siguientes características de configuración:

- b)** Los gráficos que incluyan como variable a los elementos del clima permitirán su elección.
- c)** Los gráficos permitirán modificar el periodo de tiempo en que se genera la información, fijando una fecha de inicio y fin.
- d)** Los gráficos enfocados a la severidad permitirán seleccionar la forma de representación de la información (mensual o anual) y cambiar la forma de visualización del gráfico (barras o líneas).
- e)** **Las medidas a analizar deben permitir describir el comportamiento de la Monilla en el cultivo del cacao:** Las medidas hacen referencia a los hechos presentes dentro del Data Warehouse. En este proyecto es relevante el análisis de la severidad (porcentaje de enfermedad en el fruto) y elementos climáticos, ya que son los principales factores que permitirán describir adecuadamente el comportamiento de la Monilla, una de las principales enfermedades del cacao.  
Según el precedente anterior, los datos que servirán para describir el fenómeno corresponderán a la severidad y elementos climáticos, considerando que la incidencia está directamente relacionada con la medida de la severidad.
- f)** **El contexto de análisis debe contener datos relevantes, que permitan identificar aspectos importantes y descriptivos de las medidas:** El contexto de análisis hace referencia a las dimensiones presentes dentro de un Data Warehouse, tomando en cuenta que dos de las medidas importantes son la severidad e incidencia, se considera que

las dimensiones deben contener datos relacionados a las fincas donde se han realizado los respectivos estudios de las diferentes plantas, con el fin de determinar los niveles de severidad e incidencia de cada uno de sus frutos.

Otra medida relevante se refiere a los elementos climáticos, donde se toma datos relativos a la temperatura, precipitación, punto de rocío, ráfaga de viento, velocidad viento, humedad relativa y dirección del viento. Para esto solo hay datos descriptivos relacionados con las estaciones que tomaron dicha información.

Según el precedente anterior, los datos útiles para describir las diferentes medidas corresponderán a las fincas, plantas, frutos, estudios, monitoreos y estaciones.

### **Requerimientos No Funcionales**

a) **Desarrollo de un Sitio Web para la presentación del Dashboard:** La herramienta de visualización de reportes será realizada mediante un Sitio Web.

b) **Implementación de las herramientas ROLAP y el modelo constelación de hechos:** Las herramientas ROLAP (Relational Online Analytical Processing) hace referencia a la forma en la que se desarrollara el cubo de datos, en este caso se utilizara una base de datos relacional con un modelo dimensional.

El modelo de constelación hace referencia al modelo que se implementará para crear el Data Warehouse, en el presente proyecto se utilizará un modelo de constelación.

c) **Utilización de software libre:** El presente proyecto únicamente será desarrollado con software libre, tomando en cuenta las siguientes tecnologías:

1. **Pentaho Data Integration:** Utilizado para el desarrollo de los procesos de Extracción, Transformación y Carga.

2. **PostgreSQL:** Motor de Base de datos.

#### **3. Front End:**

a. **Angular-Ionic:** Desarrollo de Single Web Application para la Interfaz de Usuario.

b. **ChartJS:** Librería para la implementación de gráficos estadísticos dentro del Dashboard.

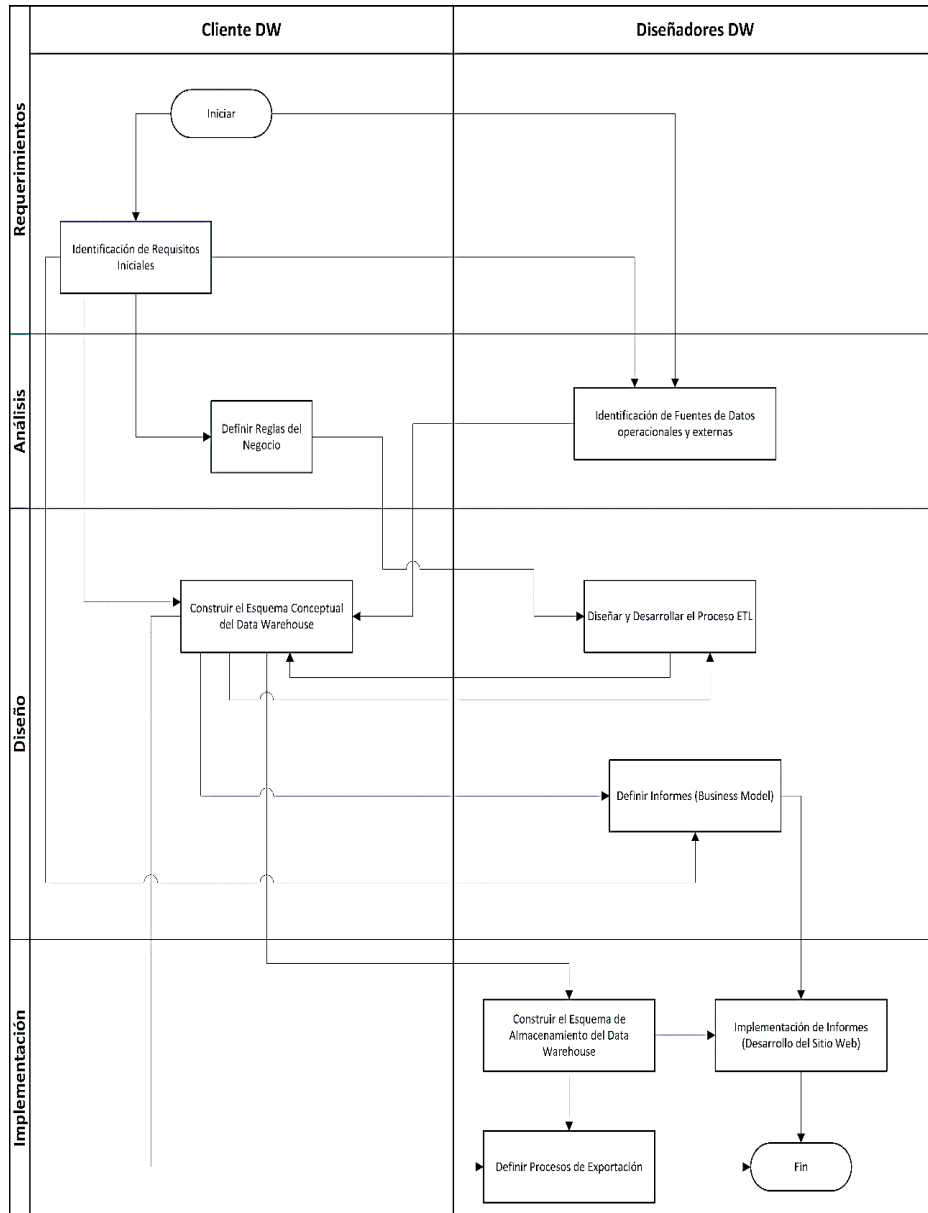
c. **Bootstrap-AdminLTE:** Librería de CSS para la implementación de estilos prediseñados.

#### **4. Back End:**

a. **Python-Flask:** Desarrollo de una REST API que provea los datos de las consultas realizadas por el Front End.

- b. **Psycopg2**: Liberaría para la realización de transacciones con la base de datos de PostgreSQL.

**Diagrama de Actividades**



**Figura 7:**Diagrama de Actividades DWEP

Las actividades para el diseño y desarrollo del Data Warehouse aplicando la metodología DWEP, están establecidas en la **Figura 7**. El diagrama de actividades presente en la **Figura 7**, cuenta con cuatro carriles horizontales y dos carriles verticales; los carriles horizontales representan los flujos de trabajo de la metodología y los carriles verticales

representan los actores presentes en el desarrollo del Data Warehouse. A continuación, se detallan las diferentes actividades presentes en los carriles horizontales:

**a) Requerimientos:**

- En esta fase, el cliente inicia al proyecto identificando los requisitos iniciales, en los que se recogerá la información necesaria para diseñar y desarrollar el producto.

**b) Análisis:**

- En la fase de análisis el paso siguiente a la identificación de requisitos corresponde a establecer las reglas de negocio; actividad que se desarrollada conjuntamente con el cliente.

- Una actividad que se desprende directamente del inicio del proyecto y de la identificación de requisitos es la identificación de las fuentes de datos operacionales y externas a cargo del diseñador del Data Warehouse. En esta actividad el diseñador se encarga de identificar las fuentes de datos necesarias para dar cumplimiento con los requerimientos establecidos.

**c) Diseño:**

- Una de las actividades presentes en esta fase, se enfoca en una de las tareas más importantes del proceso de desarrollo: La construcción del esquema conceptual, esta actividad es desarrollada juntamente con el cliente, con el fin de garantizar que el esquema propuesto cumpla con todas las necesidades del mismo. Para garantizar el cumplimiento de las necesidades del cliente, esta actividad se desprende de la identificación de requisitos y fuentes de datos, logrando tener los aspectos necesarios para construir un esquema adecuado.

- Los procesos ETL son otras de las actividades presentes en la fase de diseño; se encuentran directamente relacionados a la construcción del esquema, fuentes de datos y reglas de negocio, siendo estos últimos los que permitirán definir con claridad las características de los atributos presentes en el Data Warehouse. La actividad y la del

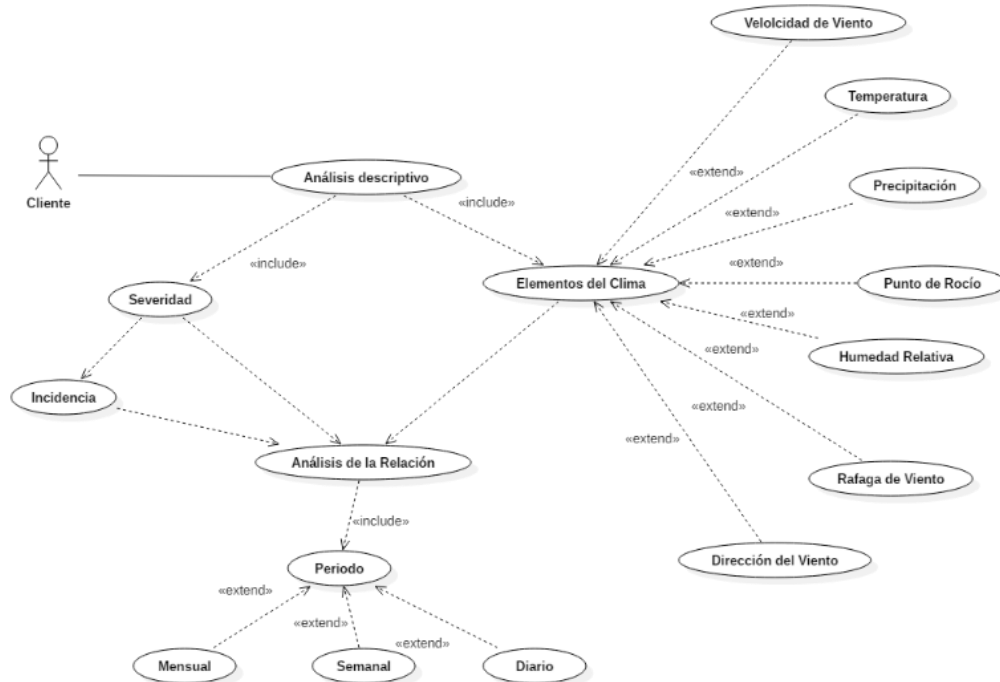
esquema conceptual resultan en un bucle, ya que se pueden identificar fallos en el esquema durante la construcción del ETL.

- La definición de informes tiene el objetivo de establecer las diferentes vistas que serán presentadas al usuario final, es decir, los diferentes reportes que se visualizaran en el dashboard.

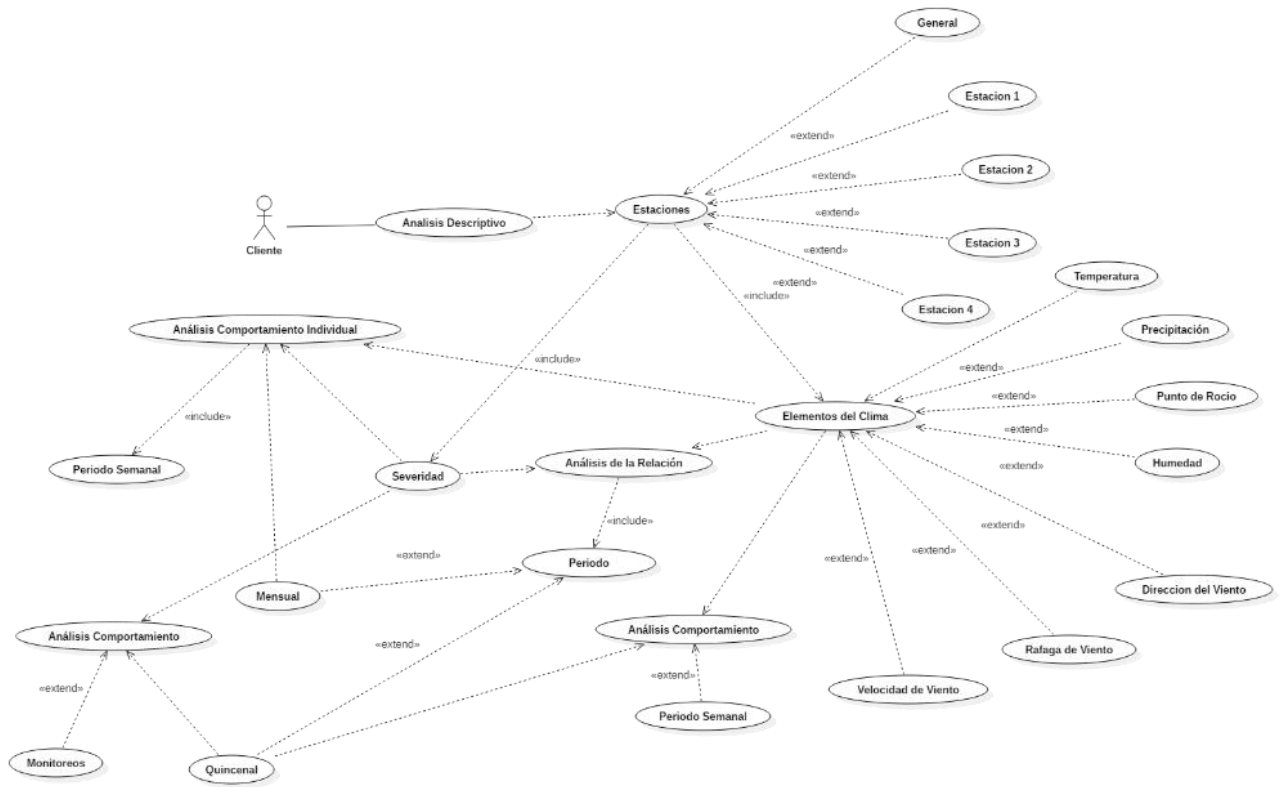
**d) Implementación:**

- En esta fase resulta relevante establecer que tecnología de almacenamiento que será utilizada para el Data Warehouse: Relacional, Multidimensional u Orientada a Objetos. La actividad de definición del almacenamiento genera como resultado un esquema lógico del Data Warehouse que establece la manera de almacenamiento de los datos.
- Para el proceso de implementación de informes se desarrollará un Sitio Web, que muestre los dashboards correspondientes a los requerimientos establecidos.
- Los procesos de exportación se refieren a cómo el esquema conceptual generado se almacenará en la tecnología de almacenamiento seleccionada.

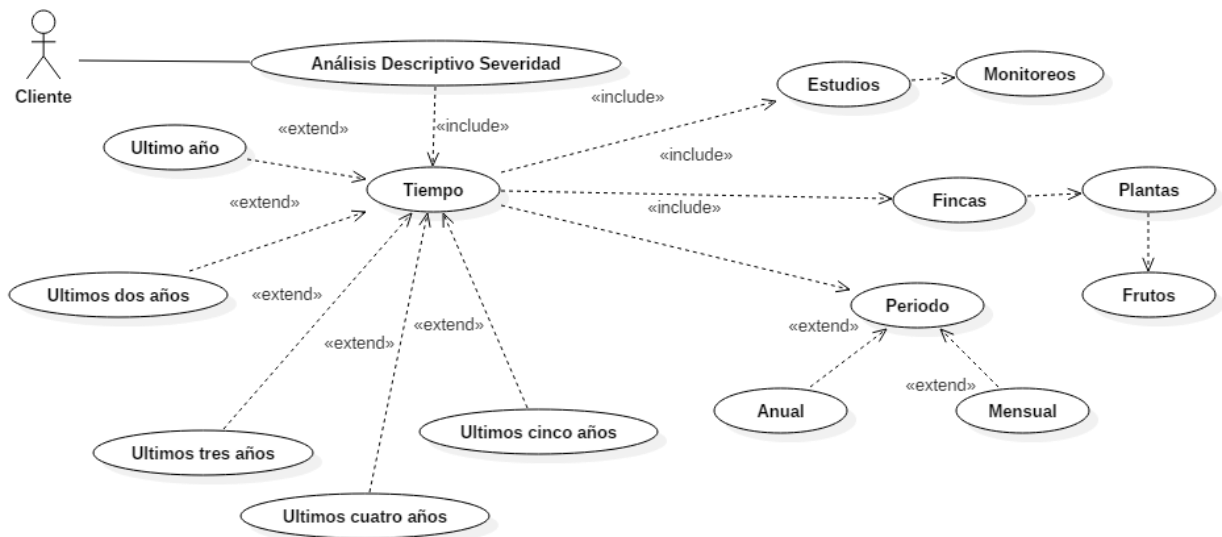
**Diagrama de Caso de uso**



**Figura 8:** Diagrama de Casos de uso General - Análisis Descriptivo Relación Severidad, Incidencia y Elementos del Clima



**Figura 9:** Diagrama de Casos de uso Estaciones – Análisis Descriptivo Relación Severidad y Elementos del Clima en función de las Estaciones



**Figura 10:** Diagrama de Casos de uso Severidad– Análisis Descriptivo Severidad en función de las Fincas y Estudios

## Especificaciones de Caso de uso del Diagrama General

**Tabla 11:** Especificación Caso de Uso Análisis Descriptivo - General

<b>Nombre</b>	Análisis Descriptivo	
<b>Actores</b>	Cliente	
<b>Propósito</b>	Representar la actividad principal que se puede realizar en el caso de uso.	
<b>Resumen</b>		
El cliente puede realizar un análisis descriptivo de la relación entre las variables de la severidad, incidencia y los diferentes elementos del clima.		
<b>Casos de uso asociados</b>	<p>Este caso de uso tiene relación de <b>inclusión</b> con los siguientes casos de uso:</p> <ul style="list-style-type: none"> <li>• <b>Severidad:</b> Este caso de uso es importante debido a que es uno de los componentes fundamentales para realizar el análisis.</li> <li>• <b>Elementos del Clima:</b> El presente caso de uso es importante, debido a que si se selecciona algún elemento del clima se podrá realizar el respectivo análisis con la severidad y la incidencia.</li> </ul>	
<b>Flujo de Trabajo</b>		
<b>Actor</b>	<b>Sistema</b>	
Realizar un análisis descriptivo de las variables severidad, incidencia y elementos del clima.	<ol style="list-style-type: none"> <li>1. Ingresar al Sitio Web.</li> <li>2. Ingresar a la sección General.</li> <li>3. Establecer las configuraciones del gráfico.</li> <li>4. Generación del Gráfico.</li> <li>5. Visualización y Análisis de los gráficos resultado por parte del cliente.</li> </ol>	
<b>Prioridad</b>	Alta.	
<b>Cursos alternos</b>		

1. <b>Flujo Alternativo 1:</b> La página no se encuentra disponible. Esto sucedería cuando exista algún problema por parte del servidor o cliente al momento de tratar de acceder a la página.
2. <b>Flujo Alternativo 2:</b> Error al intentar cargar los datos para la creación del gráfico. Esto sucedería cuando la REST API del Back End no responda la solicitud hecha por el Front End, dando como resultado que no se visualice el gráfico.

**Tabla 12:** Especificación Caso de Uso Severidad - General

<b>Nombre</b>	Severidad
<b>Actores</b>	Cliente
<b>Propósito</b>	Establecer como una de las variables de análisis a la severidad.
<b>Resumen</b>	
El cliente puede realizar un análisis descriptivo de la severidad dentro de un tiempo determinado, conjuntamente se muestra el correspondiente de los valores de la incidencia y los elementos del clima.	
<b>Casos de uso asociados</b>	Este caso de uso tiene relación de <b>dependencia</b> con los siguientes casos de uso: <ul style="list-style-type: none"> <li>• <b>Análisis de la Relación:</b> Este caso de uso depende de Severidad, ya que es uno de los componentes fundamentales para el análisis.</li> <li>• <b>Incidencia:</b> Este caso de uso depende de <b>Severidad</b> debido a que es el componente fundamental del análisis.</li> </ul>
<b>Flujo de Trabajo</b>	
<b>Actor</b>	<b>Sistema</b>
Visualizar el comportamiento de la severidad.	<ol style="list-style-type: none"> <li>1. Ingresa al Sitio Web.</li> <li>2. Ingresa a la sección General.</li> </ol>

	<p>3. Establece las configuraciones del gráfico.</p> <p>4. Generación del Gráfico.</p> <p>5. Visualización y Análisis de los gráficos resultado por parte del cliente.</p>
<b>Prioridad</b>	Alta.
<b>Cursos alternos</b>	
<p>1. <b>Flujo Alterno 1:</b> La página no se encuentra disponible. Esto sucedería cuando exista algún problema por parte del servidor o cliente al momento de tratar de acceder a la página.</p> <p>2. <b>Flujo Alterno 2:</b> Error al intentar cargar los datos para la creación del gráfico. Esto sucedería cuando la REST API del Back End no responda la solicitud hecha por el Front End, dando como resultado que no se visualice el gráfico.</p>	

**Tabla 13:** Especificación Caso de Uso Incidencia - General

<b>Nombre</b>	Incidencia
<b>Actores</b>	Cliente
<b>Propósito</b>	Establecer como una de las variables de análisis a la incidencia.
<b>Resumen</b>	
<p>El cliente puede realizar un análisis descriptivo de la incidencia dentro de un tiempo determinado, conjuntamente se muestra el correspondiente de los valores de severidad y los elementos del clima.</p>	
<b>Casos de uso asociados</b>	<p>Este caso de uso tiene relación de <b>dependencia</b> con los siguientes casos de uso:</p> <ul style="list-style-type: none"> <li>• <b>Análisis de la Relación:</b> Este caso de uso depende de <b>Incidencia</b> debido a que es uno de los componentes fundamentales para realizar el análisis.</li> </ul>
<b>Flujo de Trabajo</b>	
<b>Actor</b>	<b>Sistema</b>

Visualizar el comportamiento de la incidencia.	<ol style="list-style-type: none"> <li>1. Ingresar al Sitio Web.</li> <li>2. Ingresar a la sección General.</li> <li>3. Establecer las configuraciones del gráfico.</li> <li>4. Generación del Gráfico.</li> <li>5. Visualización y Análisis de los gráficos resultado por parte del cliente.</li> </ol>
<b>Prioridad</b>	Alta.
<b>Cursos alternos</b>	
<ol style="list-style-type: none"> <li>1. <b>Flujo Alternativo 1:</b> La página no se encuentra disponible. Esto sucedería cuando exista algún problema por parte del servidor o cliente al momento de tratar de acceder a la página.</li> <li>2. <b>Flujo Alternativo 2:</b> Error al intentar cargar los datos para la creación del gráfico. Esto sucedería cuando la REST API del Back End no responda la solicitud hecha por el Front End, dando como resultado que no se visualice el gráfico.</li> </ol>	

**Tabla 14:** Especificación Caso de Uso Elementos del Clima - General

<b>Nombre</b>	Elementos del Clima
<b>Actores</b>	Cliente
<b>Propósito</b>	Establecer como una de las variables de análisis a los elementos del clima.
<b>Resumen</b>	
<p>El cliente puede realizar un análisis descriptivo de elementos del clima: temperatura, precipitación, humedad, punto de rocío, ráfaga de viento, velocidad de viento y dirección del viento, conjuntamente se muestra el correspondiente de los valores de severidad e incidencia.</p>	
<b>Casos de uso asociados</b>	<p>Este caso de uso tiene relación de <b>dependencia</b> con los siguientes casos de uso:</p> <ul style="list-style-type: none"> <li>• <b>Análisis de la relación: Este caso de uso es importante porque es uno de</b></li> </ul>

	<p><b>los componentes fundamentales para el análisis.</b></p> <p>Este caso de uso tiene relación de <b>extends</b> con los siguientes casos de uso:</p> <ul style="list-style-type: none"> <li>• <b>Temperatura, Humedad, Precipitación, Punto de Rocío, Velocidad de Viento, Ráfaga de Viento y Dirección del Viento;</b> Estos casos de uso son importantes debido a que son los componentes fundamentales para realizar el análisis en relación con el clima.</li> </ul>
<b>Flujo de Trabajo</b>	
<b>Actor</b>	<b>Sistema</b>
Visualizar el comportamiento del elemento del clima.	<ol style="list-style-type: none"> <li>1. Ingresar al Sitio Web.</li> <li>2. Ingresar a la sección General.</li> <li>3. Establece las configuraciones del gráfico.</li> <li>4. Generación del Gráfico.</li> <li>5. Visualización y Análisis de los gráficos resultado por parte del cliente.</li> </ol>
<b>Prioridad</b>	Alta.
<b>Cursos alternos</b>	
<ol style="list-style-type: none"> <li>1. <b>Flujo Alternativo 1:</b> La página no se encuentra disponible. Esto sucedería cuando exista algún problema por parte del servidor o cliente al momento de tratar de acceder a la página.</li> <li>2. <b>Flujo Alternativo 2:</b> Error al intentar cargar los datos para la creación del gráfico. Esto sucedería cuando la REST API del Back End no responda la solicitud hecha por el Front End, dando como resultado que no se visualice el gráfico.</li> </ol>	

**Tabla 15:** Especificación Caso de Uso Temperatura - General

<b>Nombre</b>	Temperatura
---------------	-------------

<b>Actores</b>	Cliente
<b>Propósito</b>	Establecer la capacidad que brinda el sistema al usuario de seleccionar que elemento del clima desea visualizar en el resultado.
<b>Resumen</b>	
El cliente puede visualizar en un gráfico estadístico el comportamiento de la temperatura, conjuntamente se muestra el correspondiente de los valores de la incidencia y la severidad.	
<b>Casos de uso asociados</b>	El presente caso de uso es un <b>extend</b> del caso de uso <b>Elementos del Clima</b> .
<b>Flujo de Trabajo</b>	
<b>Actor</b>	<b>Sistema</b>
Visualizar la relación de la severidad, incidencia y temperatura.	<ol style="list-style-type: none"> <li>1. Ingresar al Sitio Web.</li> <li>2. Ingresar a la sección General.</li> <li>3. Establecer las configuraciones del gráfico.</li> <li>4. Iniciar la configuración del gráfico con la selección de temperatura como variable del clima.</li> <li>5. Generación del Gráfico.</li> <li>6. Visualización y Análisis de los gráficos resultado por parte del cliente.</li> </ol>
<b>Prioridad</b>	Alta.
<b>Cursos alternos</b>	
<ol style="list-style-type: none"> <li>1. <b>Flujo Alternativo 1:</b> La página no se encuentra disponible. Esto sucedería cuando exista algún problema por parte del servidor o cliente al momento de tratar de acceder a la página.</li> <li>2. <b>Flujo Alternativo 2:</b> Error al intentar cargar los datos para la creación del gráfico. Esto sucedería cuando la REST API del Back End no responda la solicitud hecha por el Front End, dando como resultado que no se visualice el gráfico.</li> </ol>	

**Tabla 16:** Especificación Caso de Uso Precipitación - General

<b>Nombre</b>	Precipitación	
<b>Actores</b>	Cliente	
<b>Propósito</b>	Establecer la capacidad que brinda el sistema al usuario de seleccionar que elemento del clima desea visualizar en el resultado.	
<b>Resumen</b>		
El cliente puede visualizar en un gráfico estadístico el comportamiento de la precipitación, conjuntamente se muestra el correspondiente de los valores de la incidencia y la severidad.		
<b>Casos de uso asociados</b>	El presente caso de uso es un <b>extend</b> del caso de uso <b>Elementos del Clima</b> .	
<b>Flujo de Trabajo</b>		
<b>Actor</b>	<b>Sistema</b>	
Visualizar la relación de la severidad, incidencia y humedad.	<ol style="list-style-type: none"> <li>1. Ingresar al Sitio Web.</li> <li>2. Ingresar a la sección General.</li> <li>3. Establece las configuraciones del gráfico.</li> <li>4. Iniciar la configuración del gráfico con la selección de precipitación como variable del clima.</li> <li>5. Generación del Gráfico.</li> <li>6. Visualización y Análisis de los gráficos resultado por parte del cliente.</li> </ol>	
<b>Prioridad</b>	Alta.	
<b>Cursos alternos</b>		
<ol style="list-style-type: none"> <li>1. <b>Flujo Alternativo 1:</b> La página no se encuentra disponible. Esto sucedería cuando exista algún problema por parte del servidor o cliente al momento de tratar de acceder a la página.</li> <li>2. <b>Flujo Alternativo 2:</b> Error al intentar cargar los datos para la creación del gráfico. Esto sucedería cuando la REST API del Back End no responda</li> </ol>		

la solicitud hecha por el Front End, dando como resultado que no se visualice el gráfico.

**Tabla 17:** Especificación Caso de Uso Ráfaga de Viento - General

<b>Nombre</b>	Ráfaga de Viento
<b>Actores</b>	Cliente
<b>Propósito</b>	Establecer la capacidad que brinda el sistema al usuario de seleccionar que elemento del clima desea visualizar en el resultado.
<b>Resumen</b>	
El cliente puede visualizar en un gráfico estadístico el comportamiento de la ráfaga de viento, conjuntamente se muestra el correspondiente de los valores de la incidencia y la severidad.	
<b>Casos de uso asociados</b>	El presente caso de uso es un <b>extend</b> del caso de uso <b>Elementos del Clima</b> .
<b>Flujo de Trabajo</b>	
<b>Actor</b>	<b>Sistema</b>
Visualizar la relación de la severidad, incidencia y humedad.	<ol style="list-style-type: none"> <li>1. Ingresar al Sitio Web.</li> <li>2. Ingresar a la sección General.</li> <li>3. Establecer las configuraciones del gráfico.</li> <li>4. Iniciar la configuración del gráfico con la selección de ráfaga de viento como variable del clima.</li> <li>5. Generación del Gráfico.</li> <li>6. Visualización y Análisis de los gráficos resultado por parte del cliente.</li> </ol>
<b>Prioridad</b>	Alta.
<b>Cursos alternos</b>	
3. <b>Flujo Alterno 1:</b> La página no se encuentra disponible. Esto sucedería cuando exista algún problema por parte del servidor o cliente al momento de tratar de acceder a la página.	

4. **Flujo Alterno 2:** Error al intentar cargar los datos para la creación del gráfico. Esto sucedería cuando la REST API del Back End no responda la solicitud hecha por el Front End, dando como resultado que no se visualice el gráfico.

**Tabla 18:** Especificación Caso de Uso Punto de Rocío - General

<b>Nombre</b>	Punto de Rocío	
<b>Actores</b>	Cliente	
<b>Propósito</b>	Establecer la capacidad que brinda el sistema al usuario de seleccionar que elemento del clima desea visualizar en el resultado.	
<b>Resumen</b>	El cliente puede visualizar en un gráfico estadístico el comportamiento del punto de rocío, conjuntamente se muestra el correspondiente de los valores de la incidencia y la severidad.	
<b>Casos de uso asociados</b>	El presente caso de uso es un <b>extend</b> del caso de uso <b>Elementos del Clima</b> .	
<b>Flujo de Trabajo</b>		
<b>Actor</b>	<b>Sistema</b>	
Visualizar la relación de la severidad, incidencia y humedad.	<ol style="list-style-type: none"> <li>1. Ingresar al Sitio Web.</li> <li>2. Ingresar a la sección General.</li> <li>3. Establece las configuraciones del gráfico.</li> <li>4. Iniciar la configuración del gráfico con la selección de punto de rocío como variable del clima.</li> <li>5. Generación del Gráfico.</li> <li>6. Visualización y Análisis de los gráficos resultado por parte del cliente.</li> </ol>	
<b>Prioridad</b>	Alta.	
<b>Cursos alternos</b>		

5. <b>Flujo Alternativo 1:</b> La página no se encuentra disponible. Esto sucedería cuando exista algún problema por parte del servidor o cliente al momento de tratar de acceder a la página.
6. <b>Flujo Alternativo 2:</b> Error al intentar cargar los datos para la creación del gráfico. Esto sucedería cuando la REST API del Back End no responda la solicitud hecha por el Front End, dando como resultado que no se visualice el gráfico.

**Tabla 19:** Especificación Caso de Uso Velocidad de Viento - General

<b>Nombre</b>	Velocidad de Viento	
<b>Actores</b>	Cliente	
<b>Propósito</b>	Establecer la capacidad que brinda el sistema al usuario de seleccionar que elemento del clima desea visualizar en el resultado.	
<b>Resumen</b>		
El cliente puede visualizar en un gráfico estadístico el comportamiento de la velocidad del viento, conjuntamente se muestra el correspondiente de los valores de la incidencia y la severidad.		
<b>Casos de uso asociados</b>	El presente caso de uso es un <b>extend</b> del caso de uso <b>Elementos del Clima</b> .	
<b>Flujo de Trabajo</b>		
<b>Actor</b>	<b>Sistema</b>	
Visualizar la relación de la severidad, incidencia y humedad.	<ol style="list-style-type: none"> <li>1. Ingresa al Sitio Web.</li> <li>2. Ingresa a la sección General.</li> <li>3. Establece las configuraciones del gráfico.</li> <li>4. Iniciar la configuración del gráfico con la selección de velocidad del viento como variable del clima.</li> <li>5. Generación del Gráfico.</li> </ol>	

	6. Visualización y Análisis de los gráficos resultado por parte del cliente.
<b>Prioridad</b>	Alta.
<b>Cursos alternos</b>	
7. <b>Flujo Alterno 1:</b> La página no se encuentra disponible. Esto sucedería cuando exista algún problema por parte del servidor o cliente al momento de tratar de acceder a la página.	
8. <b>Flujo Alterno 2:</b> Error al intentar cargar los datos para la creación del gráfico. Esto sucedería cuando la REST API del Back End no responda la solicitud hecha por el Front End, dando como resultado que no se visualice el gráfico.	

**Tabla 20:** Especificación Caso de Uso Humedad Relativa - General

<b>Nombre</b>	Humedad Relativa
<b>Actores</b>	Cliente
<b>Propósito</b>	Establecer la capacidad que brinda el sistema al usuario de seleccionar que elemento del clima desea visualizar en el resultado.
<b>Resumen</b>	
El cliente puede visualizar en un gráfico estadístico el comportamiento de la humedad relativa, conjuntamente se muestra el correspondiente de los valores de la incidencia y la severidad.	
<b>Casos de uso asociados</b>	El presente caso de uso es un <b>extend</b> del caso de uso <b>Elementos del Clima</b> .
<b>Flujo de Trabajo</b>	
<b>Actor</b>	<b>Sistema</b>
Visualizar la relación de la severidad, incidencia y humedad relativa.	<ol style="list-style-type: none"> <li>1. Ingresa al Sitio Web.</li> <li>2. Ingresa a la sección General.</li> <li>3. Establece las configuraciones del gráfico.</li> </ol>

	<p>4. Iniciar la configuración del gráfico con la selección de humedad relativa como variable del clima.</p> <p>5. Generación del Gráfico.</p> <p>6. Visualización y Análisis de los gráficos resultado por parte del cliente.</p>
<b>Prioridad</b>	Alta.
<b>Cursos alternos</b>	
<p>1. <b>Flujo Alterno 1:</b> La página no se encuentra disponible. Esto sucedería cuando exista algún problema por parte del servidor o cliente al momento de tratar de acceder a la página.</p> <p>2. <b>Flujo Alterno 2:</b> Error al intentar cargar los datos para la creación del gráfico. Esto sucedería cuando la REST API del Back End no responda la solicitud hecha por el Front End, dando como resultado que no se visualice el gráfico.</p>	

**Tabla 21:** Especificación Caso de Uso Dirección del Viento - General

<b>Nombre</b>	Dirección del Viento	
<b>Actores</b>	Cliente	
<b>Propósito</b>	Establecer la capacidad que brinda el sistema al usuario de seleccionar que elemento del clima desea visualizar en el resultado.	
<b>Resumen</b>		
El cliente puede visualizar en un gráfico estadístico el comportamiento de la dirección del viento, conjuntamente se muestra el correspondiente de los valores de la incidencia y la severidad.		
<b>Casos de uso asociados</b>	El presente caso de uso es un <b>extend</b> del caso de uso <b>Elementos del Clima</b> .	
<b>Flujo de Trabajo</b>		
<b>Actor</b>	<b>Sistema</b>	

Visualizar la relación de la severidad, incidencia y la dirección del viento.	<ol style="list-style-type: none"> <li>1. Ingresa al Sitio Web.</li> <li>2. Ingresa a la sección General.</li> <li>3. Establece las configuraciones del gráfico.</li> <li>4. Iniciar la configuración del gráfico con la selección de dirección del viento como variable del clima.</li> <li>5. Generación del Gráfico.</li> <li>6. Visualización y Análisis de los gráficos resultado por parte del cliente.</li> </ol>
<b>Prioridad</b>	Alta.
<p><b>Cursos alternos</b></p> <ol style="list-style-type: none"> <li>1. <b>Flujo Alterno 1:</b> La página no se encuentra disponible. Esto sucedería cuando exista algún problema por parte del servidor o cliente al momento de tratar de acceder a la página.</li> <li>2. <b>Flujo Alterno 2:</b> Error al intentar cargar los datos para la creación del gráfico. Esto sucedería cuando la REST API del Back End no responda la solicitud hecha por el Front End, dando como resultado que no se visualice el gráfico.</li> </ol>	

**Tabla 22:** Especificación Caso de Uso Análisis de la Relación - General

<b>Nombre</b>	Análisis de la Relación
<b>Actores</b>	Cliente
<b>Propósito</b>	Representar la actividad principal que se puede realizar en el caso de uso.
<p><b>Resumen</b></p> <p>El cliente puede realizar un análisis de relación descriptivo de la relación entre las variables de la severidad, incidencia y los diferentes elementos del clima.</p>	
<b>Casos de uso asociados</b>	Este caso de uso tiene relación de <b>inclusión</b> con el siguiente caso de uso:

	<ul style="list-style-type: none"> <li>• <b>Periodo:</b> Este caso de uso es importante porque, si se selecciona un periodo específico, esta muestra junto con las variables de incidencia, severidad y los elementos del clima, información según el periodo seleccionado.</li> </ul>
<b>Flujo de Trabajo</b>	
<b>Actor</b>	<b>Sistema</b>
Realizar un análisis descriptivo de las variables severidad y elementos del clima dentro de un periodo.	<ol style="list-style-type: none"> <li>1. Ingresa al Sitio Web.</li> <li>2. Ingresa a la sección General.</li> <li>3. Establece las configuraciones del gráfico.</li> <li>4. Iniciar la configuración del gráfico con la selección de un elemento del clima.</li> <li>5. Iniciar la configuración del gráfico con la selección de un periodo de fecha permitidos (Desde – Hasta).</li> <li>6. Generación del Gráfico.</li> <li>7. Visualización y Análisis de los gráficos resultado por parte del cliente.</li> </ol>
<b>Prioridad</b>	Alta.
<b>Cursos alternos</b>	
<ol style="list-style-type: none"> <li>1. <b>Flujo Alterno 1:</b> La página no se encuentra disponible. Esto sucedería cuando exista algún problema por parte del servidor o cliente al momento de tratar de acceder a la página.</li> <li>2. <b>Flujo Alterno 2:</b> Error al intentar cargar los datos para la creación del gráfico. Esto sucedería cuando la REST API del Back End no responda la solicitud hecha por el Front End, dando como resultado que no se visualice el gráfico.</li> </ol>	

**Tabla 23:** Especificación Caso de Uso Periodo - General

<b>Nombre</b>	Periodo
<b>Actores</b>	Cliente
<b>Propósito</b>	Establecer las variables de análisis en función de un periodo.
<b>Resumen</b>	
El cliente puede visualizar en un gráfico estadístico del comportamiento dentro de un periodo específico la relación entre las variables incidencia, severidad y los diferentes elementos del clima.	
<b>Casos de uso asociados</b>	Este caso de uso tiene relación de <b>extends</b> con los siguientes casos de uso: <ul style="list-style-type: none"> <li>• <b>Mensual, Semanal y Diario;</b></li> </ul> Estos casos de uso son importantes debido a que son los componentes fundamentales para realizar el análisis de las variables de severidad, incidencia y los elementos del clima en función de periodo determinado.
<b>Flujo de Trabajo</b>	
<b>Actor</b>	<b>Sistema</b>
Realizar un análisis descriptivo de la variable severidad, incidencia y los elementos del clima en función de un periodo específico.	<ol style="list-style-type: none"> <li>1. Ingresa al Sitio Web.</li> <li>2. Ingresa a la sección General.</li> <li>3. Establece las configuraciones del gráfico.</li> <li>4. Iniciar la configuración del gráfico con la selección de un periodo de fecha permitidos (Desde – Hasta).</li> <li>5. Generación del Gráfico.</li> <li>6. Visualización y Análisis de los gráficos resultado por parte del cliente.</li> </ol>
<b>Prioridad</b>	Alta.
<b>Cursos alternos</b>	

1. **Flujo Alternativo 1:** La página no se encuentra disponible. Esto sucedería cuando exista algún problema por parte del servidor o cliente al momento de tratar de acceder a la página.
2. **Flujo Alternativo 2:** Error al intentar cargar los datos para la creación del gráfico. Esto sucedería cuando la REST API del Back End no responda la solicitud hecha por el Front End, dando como resultado que no se visualice el gráfico.

**Tabla 24:** Especificación Caso de Uso Mensual – General

<b>Nombre</b>	Mensual
<b>Actores</b>	Cliente
<b>Propósito</b>	Establecer la capacidad que brinda el sistema al usuario de seleccionar que periodo específico que desea visualizar en el resultado.
<b>Resumen</b>	
El cliente puede visualizar en un gráfico estadístico del comportamiento mensual de las variables incidencia, severidad y los diferentes elementos del clima.	
<b>Casos de uso asociados</b>	El presente caso de uso es un <b>extend</b> del caso de uso <b>Periodo</b> .
<b>Flujo de Trabajo</b>	
<b>Actor</b>	<b>Sistema</b>
Visualizar la relación de la severidad, incidencia y los elementos del clima en un periodo mensual.	<ol style="list-style-type: none"> <li>1. Ingresa al Sitio Web.</li> <li>2. Ingresa a la sección General.</li> <li>3. Establece las configuraciones del gráfico.</li> <li>4. Iniciar la configuración del gráfico con la selección de un elemento del clima.</li> <li>5. Iniciar la configuración del gráfico con la selección de un periodo de fecha permitidos (Desde – Hasta).</li> </ol>

	6. Generación del Gráfico Mensual. 7. Visualización y Análisis de los gráficos resultado por parte del cliente.
<b>Prioridad</b>	Media.
<b>Cursos alternos</b>	
1. <b>Flujo Alterno 1:</b> La página no se encuentra disponible. Esto sucedería cuando exista algún problema por parte del servidor o cliente al momento de tratar de acceder a la página.	
2. <b>Flujo Alterno 2:</b> Error al intentar cargar los datos para la creación del gráfico. Esto sucedería cuando la REST API del Back End no responda la solicitud hecha por el Front End, dando como resultado que no se visualice el gráfico.	

**Tabla 25:** Especificación Caso de Uso Semanal – General

<b>Nombre</b>	Semanal
<b>Actores</b>	Cliente
<b>Propósito</b>	Establecer la capacidad que brinda el sistema al usuario de seleccionar que periodo específico que desea visualizar en el resultado.
<b>Resumen</b>	
El cliente puede visualizar en un gráfico estadístico del comportamiento semanal de las variables incidencia, severidad y los diferentes elementos del clima.	
<b>Casos de uso asociados</b>	El presente caso de uso es un <b>extend</b> del caso de uso <b>Periodo</b> .
<b>Flujo de Trabajo</b>	
<b>Actor</b>	<b>Sistema</b>
Visualizar la relación de la severidad, incidencia y los elementos del clima en un periodo semanal.	1. Ingresa al Sitio Web. 2. Ingresa a la sección General. 3. Establece las configuraciones del gráfico.

	<p>4. Iniciar la configuración del gráfico con la selección de un elemento del clima.</p> <p>5. Iniciar la configuración del gráfico con la selección de un periodo de fecha permitidos (Desde – Hasta).</p> <p>6. Generación del Gráfico Semanal.</p> <p>7. Visualización y Análisis de los gráficos resultado por parte del cliente.</p>
<b>Prioridad</b>	Media.
<b>Cursos alternos</b>	
<p>1. <b>Flujo Alternativo 1:</b> La página no se encuentra disponible. Esto sucedería cuando exista algún problema por parte del servidor o cliente al momento de tratar de acceder a la página.</p> <p>2. <b>Flujo Alternativo 2:</b> Error al intentar cargar los datos para la creación del gráfico. Esto sucedería cuando la REST API del Back End no responda la solicitud hecha por el Front End, dando como resultado que no se visualice el gráfico.</p>	

**Tabla 26:** Especificación Caso de Uso Diario – General

<b>Nombre</b>	Diario
<b>Actores</b>	Cliente
<b>Propósito</b>	Establecer la capacidad que brinda el sistema al usuario de seleccionar que periodo específico que desea visualizar en el resultado.
<b>Resumen</b>	
El cliente puede visualizar en un gráfico estadístico del comportamiento diario de las variables incidencia, severidad y los diferentes elementos del clima.	
<b>Casos de uso asociados</b>	El presente caso de uso es un <b>extend</b> del caso de uso <b>Periodo</b> .

<b>Flujo de Trabajo</b>	
<b>Actor</b>	<b>Sistema</b>
Visualizar la relación de la severidad, incidencia y los elementos del clima en un periodo diario.	<ol style="list-style-type: none"> <li>1. Ingresa al Sitio Web.</li> <li>2. Ingresa a la sección General.</li> <li>3. Establece las configuraciones del gráfico.</li> <li>4. Iniciar la configuración del gráfico con la selección de un elemento del clima.</li> <li>5. Iniciar la configuración del gráfico con la selección de un periodo de fecha permitidos (Desde – Hasta).</li> <li>6. Generación del Gráfico Diario.</li> <li>7. Visualización y Análisis de los gráficos resultado por parte del cliente.</li> </ol>
<b>Prioridad</b>	Media.
<b>Cursos alternos</b>	
<ol style="list-style-type: none"> <li>1. <b>Flujo Alterno 1:</b> La página no se encuentra disponible. Esto sucedería cuando exista algún problema por parte del servidor o cliente al momento de tratar de acceder a la página.</li> <li>2. <b>Flujo Alterno 2:</b> Error al intentar cargar los datos para la creación del gráfico. Esto sucedería cuando la REST API del Back End no responda la solicitud hecha por el Front End, dando como resultado que no se visualice el gráfico.</li> </ol>	

### **Especificaciones de Caso de uso del Diagrama Estaciones**

**Tabla 27:** Especificación Caso de Uso Análisis Descriptivo - Estaciones

<b>Nombre</b>	Análisis Descriptivo
<b>Actores</b>	Cliente
<b>Propósito</b>	Representar la actividad principal que se puede realizar en el caso de uso.

<b>Resumen</b>	
El cliente puede realizar un análisis descriptivo de la relación entre las variables de la severidad y elementos del clima, observando la información en función de las tres estaciones existentes.	
<b>Casos de uso asociados</b>	El caso Análisis Descriptivo tiene una relación de dependencia con Estaciones, donde este depende del Análisis Descriptivo, ya que los gráficos generados muestran de forma descriptiva el comportamiento de las variables involucradas.
<b>Flujo de Trabajo</b>	
<b>Actor</b>	<b>Sistema</b>
Realizar un análisis descriptivo de las variables severidad y elementos del clima en función de las estaciones.	<ol style="list-style-type: none"> <li>1. Ingresa al Sitio Web.</li> <li>2. Ingresa a la sección General.</li> <li>3. Establece las configuraciones del gráfico.</li> <li>4. Generación del Gráfico.</li> <li>5. Visualización y Análisis de los gráficos resultado por parte del cliente.</li> </ol>
<b>Prioridad</b>	Alta.
<b>Cursos alternos</b>	
<ol style="list-style-type: none"> <li>1. <b>Flujo Alternativo 1:</b> La página no se encuentra disponible. Esto sucedería cuando exista algún problema por parte del servidor o cliente al momento de tratar de acceder a la página.</li> <li>2. <b>Flujo Alternativo 2:</b> No se encuentra la sección <b>General</b>. Esto sucedería cuando el usuario de clic en la opción <b>General</b> y el sitio web lance un error 404 indicando que no existe, esto representaría un problema en la configuración de las rutas del Sitio Web.</li> <li>3. <b>Flujo Alternativo 3:</b> Error al intentar cargar los datos para la creación del gráfico. Esto sucedería cuando la REST API del Back End no responda</li> </ol>	

la solicitud hecha por el Front End, dando como resultado que no se visualice el gráfico.

**Tabla 28:** Especificación Caso de Uso Estaciones - Estaciones

<b>Nombre</b>	Estaciones
<b>Actores</b>	Cliente
<b>Propósito</b>	Representar la forma de clasificación de los resultados que se pueden visualizar en el diagrama de caso de uso.
<b>Resumen</b>	
El cliente puede realizar un análisis descriptivo clasificando los resultados a partir de las estaciones presentes en el sistema, dicho análisis va enfocado a describir la relación de las variables severidad y elementos del clima.	
<b>Casos de uso asociados</b>	<ul style="list-style-type: none"> <li>• <b>Extends:</b> El caso de uso <b>Estaciones</b>, tiene a los casos de uso: <b>General</b>, <b>Estación 1</b>, <b>Estación 2</b> y <b>Estación 3</b> como extends, los cuales simbolizan se puede realizar el mismo análisis y obtener los mismos gráficos, para todas las estaciones o de manera individual para cada una de ellas.</li> <li>• <b>Includes:</b> El presente caso de uso tiene a los casos de uso: <b>Severidad</b> y <b>Elementos del Clima</b> como includes, debido a que, si se selecciona todas las estaciones o una de ellas, en cualquiera de los casos se podrá realizar el respectivo análisis de las variables severidad y elementos del clima, es decir, <b>Estaciones</b> incluye ambos casos de uso.</li> </ul>
<b>Flujo de Trabajo</b>	
<b>Actor</b>	<b>Sistema</b>

Clasificar los resultados en función de las estaciones, pudiendo ser de manera general o individual.	<ol style="list-style-type: none"> <li>1. Ingresa al Sitio Web.</li> <li>2. Ingresa a la Sección Estaciones.</li> <li>3. Selecciona la sección de todas las estaciones o una estación individual.</li> <li>4. Configura el gráfico.</li> <li>5. Generación del gráfico que muestra la severidad y el elemento del clima.</li> <li>6. Visualización y Análisis de los gráficos resultado por parte del cliente.</li> </ol>
<b>Prioridad</b>	Alta.
<p><b>Cursos alternos</b></p> <ol style="list-style-type: none"> <li>1. <b>Flujo Alterno 1:</b> La página no se encuentra disponible. Esto sucedería cuando exista algún problema por parte del servidor o cliente al momento de tratar de acceder a la página.</li> <li>2. <b>Flujo Alterno 2:</b> Error al intentar cargar los datos para la creación del gráfico. Esto sucedería cuando la REST API del Back End no responda la solicitud hecha por el Front End, dando como resultado que no se visualice el gráfico.</li> </ol>	

**Tabla 29:** Especificación Caso de Uso General - Estaciones

<b>Nombre</b>	General
<b>Actores</b>	Cliente
<b>Propósito</b>	Establecer la capacidad del sistema de mostrar los resultados para todas las estaciones.
<p><b>Resumen</b></p> <p>El cliente puede visualizar los resultados de los elementos del clima que hayan sido recolectados por todas las estaciones y los resultados correspondientes a la severidad.</p>	
<b>Casos de uso asociados</b>	Este caso de uso es un extend del caso de uso <b>Estaciones</b> , debido a que mantiene la

	misma funcionalidad con pequeños cambios.
<b>Flujo de Trabajo</b>	
<b>Actor</b>	<b>Sistema</b>
Visualizar el gráfico que contenga el comportamiento del elemento del clima a partir de los datos recolectados por todas las estaciones y resultados correspondientes de severidad.	<ol style="list-style-type: none"> <li>1. Ingresa al Sitio Web.</li> <li>2. Ingresa a la Sección Estaciones.</li> <li>3. Selecciona la opción de todas las Estaciones.</li> <li>4. Configuración del gráfico.</li> <li>5. Generación del gráfico.</li> <li>6. Visualización y análisis de los resultados por parte del cliente.</li> </ol>
<b>Prioridad</b>	Media.
<b>Cursos alternos</b>	
<ol style="list-style-type: none"> <li>1. <b>Flujo Alterno 1:</b> La página no se encuentra disponible. Esto sucedería cuando exista algún problema por parte del servidor o cliente al momento de tratar de acceder a la página.</li> <li>2. <b>Flujo Alterno 2:</b> No se encuentra la sección <b>Estaciones</b>. Esto sucedería cuando el usuario de clic en la opción <b>Estaciones</b> y el sitio web lance un error 404 indicando que no existe, esto representaría un problema en la configuración de las rutas del Sitio Web.</li> <li>3. <b>Flujo Alterno 3:</b> Error al intentar cargar los datos para la creación del gráfico. Esto sucedería cuando la REST API del Back End no responda la solicitud hecha por el Front End, dando como resultado que no se visualice el gráfico.</li> </ol>	

**Tabla 30:** Especificación Caso de Uso Severidad - Estaciones

<b>Nombre</b>	Severidad
<b>Actores</b>	Cliente
<b>Propósito</b>	Establecer como una de las variables de análisis a la severidad.
<b>Resumen</b>	

<p>El cliente puede hacer un análisis de la severidad, donde los datos representados en el gráfico estarán relacionados con el tiempo en que se realizó las mediciones de los elementos del clima para cada estación, es decir, solo se tomarán en cuenta los datos de la severidad que estén en las fechas donde se realizaron dichas mediciones.</p>	
<p><b>Casos de uso asociados</b></p>	<p>Este caso de uso tiene relación de dependencia con los siguientes casos de uso:</p> <ul style="list-style-type: none"> <li>• <b>Análisis de la Relación:</b> Este caso de uso depende de Severidad, ya que es uno de los componentes fundamentales para el análisis.</li> <li>• <b>Análisis Comportamiento:</b> Este caso de uso depende de <b>Severidad</b> debido a que es el componente fundamental del análisis.</li> <li>• <b>Análisis Comportamiento Individual:</b> Este caso es dependiente porque el único componente de análisis es la Severidad.</li> </ul>
<p><b>Flujo de Trabajo</b></p>	
<p><b>Actor</b></p>	<p><b>Sistema</b></p>
<p>Visualizar el comportamiento de la severidad en función de las estaciones.</p>	<ol style="list-style-type: none"> <li>1. Ingresa al Sitio Web.</li> <li>2. Ingresa a la Sección Estaciones.</li> <li>3. Selecciona la opción de todas las estaciones o individual.</li> <li>4. Configuración del Gráfico.</li> <li>5. Generación del Gráfico.</li> <li>6. Visualización de los resultados por parte del cliente.</li> </ol>
<p><b>Prioridad</b></p>	<p>Alta.</p>
<p><b>Cursos alternos</b></p>	

1. **Flujo Alternativo 1:** La página no se encuentra disponible. Esto sucedería cuando exista algún problema por parte del servidor o cliente al momento de tratar de acceder a la página.
2. **Flujo Alternativo 2:** Error al intentar cargar los datos para la creación del gráfico. Esto sucedería cuando la REST API del Back End no responda la solicitud hecha por el Front End, dando como resultado que no se visualice el gráfico.

**Tabla 31:** Especificación Caso de Uso Elementos del Clima - Estaciones

<b>Nombre</b>	Elementos del Clima
<b>Actores</b>	Cliente
<b>Propósito</b>	Establecer como una de las variables de análisis a los elementos del clima.
<b>Resumen</b>	
El cliente puede analizar elementos del clima: temperatura, humedad, humedad relativa y dirección del viento; de todas las estaciones en conjunto o individualmente, se muestra el correspondiente de los valores de severidad.	
<b>Casos de uso asociados</b>	<ul style="list-style-type: none"> <li>• <b>Dependencia:</b> Este caso de uso se relaciona con la dependencia de estos casos: Análisis de la Relación, Análisis Comportamiento y Análisis Comportamiento Individual; ya que resulta ser un componente fundamental para realizarlo.</li> <li>• <b>Extends:</b> Los extends del caso de uso son: <b>Temperatura, Humedad, Humedad Relativa y Dirección del Viento;</b> tienen el objetivo de representar la facultad del sistema de permitir la selección de una de</li> </ul>

	las variables del clima antes mencionada para mostrar los resultados en la gráfica.
<b>Flujo de Trabajo</b>	
<b>Actor</b>	<b>Sistema</b>
Visualizar el comportamiento del elemento del clima en función de las estaciones.	<ol style="list-style-type: none"> <li>1. Ingresa al Sitio Web.</li> <li>2. Ingresa a la Sección Estaciones.</li> <li>3. Selecciona la opción de todas las estaciones o individual.</li> <li>4. Inicia la configuración del gráfico con la selección de la variable del clima que desea visualizar.</li> <li>5. Termina la configuración del gráfico.</li> <li>6. Generación del Gráfico.</li> <li>7. Visualización y análisis de los resultados por parte del cliente.</li> </ol>
<b>Prioridad</b>	Alta.
<b>Cursos alternos</b>	
<ol style="list-style-type: none"> <li>1. <b>Flujo Alterno 1:</b> La página no se encuentra disponible. Esto sucedería cuando exista algún problema por parte del servidor o cliente al momento de tratar de acceder a la página.</li> <li>2. <b>Flujo Alterno 2:</b> Error al intentar cargar los datos para la creación del gráfico. Esto sucedería cuando la REST API del Back End no responda la solicitud hecha por el Front End, dando como resultado que no se visualice el gráfico.</li> </ol>	

**Tabla 32:** Especificación Caso de Uso Temperatura - Estaciones

<b>Nombre</b>	Temperatura
<b>Actores</b>	Cliente
<b>Propósito</b>	Establecer la capacidad que brinda el sistema al usuario de seleccionar que elemento del clima desea visualizar en el resultado.
<b>Resumen</b>	

El cliente puede visualizar en un gráfico estadístico el comportamiento de la temperatura y la severidad en función de las estaciones.	
<b>Casos de uso asociados</b>	El presente caso de uso es un extend del caso de uso <b>Elementos del Clima</b> .
<b>Flujo de Trabajo</b>	
<b>Actor</b>	<b>Sistema</b>
Visualizar la relación de la severidad y temperatura en función de las estaciones.	<ol style="list-style-type: none"> <li>1. Ingresar al Sitio Web.</li> <li>2. Ingresar a la Sección Estaciones.</li> <li>3. Seleccionar la opción todas las estaciones o individual.</li> <li>4. Iniciar la configuración del gráfico con la selección de temperatura como variable del clima.</li> <li>5. Finalizar la configuración del gráfico.</li> <li>6. Generación del Gráfico.</li> <li>7. Visualización y análisis por parte del cliente.</li> </ol>
<b>Prioridad</b>	Media.
<b>Cursos alternos</b>	
<ol style="list-style-type: none"> <li>1. <b>Flujo Alternativo 1:</b> La página no se encuentra disponible. Esto sucedería cuando exista algún problema por parte del servidor o cliente al momento de tratar de acceder a la página.</li> <li>2. <b>Flujo Alternativo 2:</b> Error al intentar cargar los datos para la creación del gráfico. Esto sucedería cuando la REST API del Back End no responda la solicitud hecha por el Front End, dando como resultado que no se visualice el gráfico.</li> </ol>	

**Tabla 33:** Especificación Caso de Uso Precipitación - Estaciones

<b>Nombre</b>	Precipitación
<b>Actores</b>	Cliente

<b>Propósito</b>	Establecer la capacidad que brinda el sistema al usuario de seleccionar que elemento del clima desea visualizar en el resultado.
<b>Resumen</b>	
El cliente puede visualizar en un gráfico estadístico el comportamiento de la precipitación y la severidad en función de las estaciones.	
<b>Casos de uso asociados</b>	El presente caso de uso es un extend del caso de uso <b>Elementos del Clima</b> .
<b>Flujo de Trabajo</b>	
<b>Actor</b>	<b>Sistema</b>
Visualizar la relación de la severidad y humedad en función de las estaciones.	<ol style="list-style-type: none"> <li>1. Ingresa al Sitio Web.</li> <li>2. Ingresar a la Sección Estaciones.</li> <li>3. Seleccionar la opción todas las estaciones o individual.</li> <li>4. Iniciar la configuración del gráfico con la selección de precipitación como variable del clima.</li> <li>5. Finalizar la configuración del gráfico.</li> <li>6. Generación del Gráfico.</li> <li>7. Visualización y análisis por parte del cliente.</li> </ol>
<b>Prioridad</b>	Media.
<b>Cursos alternos</b>	
<ol style="list-style-type: none"> <li>1. <b>Flujo Alternativo 1:</b> La página no se encuentra disponible. Esto sucedería cuando exista algún problema por parte del servidor o cliente al momento de tratar de acceder a la página.</li> <li>2. <b>Flujo Alternativo 2:</b> Error al intentar cargar los datos para la creación del gráfico. Esto sucedería cuando la REST API del Back End no responda la solicitud hecha por el Front End, dando como resultado que no se visualice el gráfico.</li> </ol>	

**Tabla 34:** Especificación Caso de Uso Ráfaga de Viento - Estaciones

<b>Nombre</b>	Ráfaga de Viento	
<b>Actores</b>	Cliente	
<b>Propósito</b>	Establecer la capacidad que brinda el sistema al usuario de seleccionar que elemento del clima desea visualizar en el resultado.	
<b>Resumen</b>		
El cliente puede visualizar en un gráfico estadístico el comportamiento de la ráfaga de viento, conjuntamente se muestra el correspondiente de los valores de la incidencia y la severidad.		
<b>Casos de uso asociados</b>	El presente caso de uso es un <b>extend</b> del caso de uso <b>Elementos del Clima</b> .	
<b>Flujo de Trabajo</b>		
<b>Actor</b>	<b>Sistema</b>	
Visualizar la relación de la severidad, incidencia y humedad.	<ol style="list-style-type: none"> <li>1. Ingresa al Sitio Web.</li> <li>2. Ingresa a la sección General.</li> <li>3. Establece las configuraciones del gráfico.</li> <li>4. Iniciar la configuración del gráfico con la selección de ráfaga de viento como variable del clima.</li> <li>5. Generación del Gráfico.</li> <li>6. Visualización y Análisis de los gráficos resultado por parte del cliente.</li> </ol>	
<b>Prioridad</b>	Media.	
<b>Cursos alternos</b>		
<p>9. <b>Flujo Alternativo 1:</b> La página no se encuentra disponible. Esto sucedería cuando exista algún problema por parte del servidor o cliente al momento de tratar de acceder a la página.</p> <p>10. <b>Flujo Alternativo 2:</b> Error al intentar cargar los datos para la creación del gráfico. Esto sucedería cuando la REST API del Back End no responda</p>		

la solicitud hecha por el Front End, dando como resultado que no se visualice el gráfico.

**Tabla 35:** Especificación Caso de Uso Punto de Rocío - Estaciones

<b>Nombre</b>	Punto de Rocío	
<b>Actores</b>	Cliente	
<b>Propósito</b>	Establecer la capacidad que brinda el sistema al usuario de seleccionar que elemento del clima desea visualizar en el resultado.	
<b>Resumen</b>		
El cliente puede visualizar en un gráfico estadístico el comportamiento del punto de rocío, conjuntamente se muestra el correspondiente de los valores de la incidencia y la severidad.		
<b>Casos de uso asociados</b>	El presente caso de uso es un <b>extend</b> del caso de uso <b>Elementos del Clima</b> .	
<b>Flujo de Trabajo</b>		
<b>Actor</b>	<b>Sistema</b>	
Visualizar la relación de la severidad, incidencia y humedad.	<ol style="list-style-type: none"> <li>1. Ingresa al Sitio Web.</li> <li>2. Ingresa a la sección General.</li> <li>3. Establece las configuraciones del gráfico.</li> <li>4. Iniciar la configuración del gráfico con la selección de punto de rocío como variable del clima.</li> <li>5. Generación del Gráfico.</li> <li>6. Visualización y Análisis de los gráficos resultado por parte del cliente.</li> </ol>	
<b>Prioridad</b>	Media.	
<b>Cursos alternos</b>		

1. **Flujo Alternativo 1:** La página no se encuentra disponible. Esto sucedería cuando exista algún problema por parte del servidor o cliente al momento de tratar de acceder a la página.
2. **Flujo Alternativo 2:** Error al intentar cargar los datos para la creación del gráfico. Esto sucedería cuando la REST API del Back End no responda la solicitud hecha por el Front End, dando como resultado que no se visualice el gráfico.

**Tabla 36:** Especificación Caso de Uso Velocidad de Viento - Estaciones

<b>Nombre</b>	Velocidad de Viento	
<b>Actores</b>	Cliente	
<b>Propósito</b>	Establecer la capacidad que brinda el sistema al usuario de seleccionar que elemento del clima desea visualizar en el resultado.	
<b>Resumen</b>		
El cliente puede visualizar en un gráfico estadístico el comportamiento de la velocidad del viento, conjuntamente se muestra el correspondiente de los valores de la incidencia y la severidad.		
<b>Casos de uso asociados</b>	El presente caso de uso es un <b>extend</b> del caso de uso <b>Elementos del Clima</b> .	
<b>Flujo de Trabajo</b>		
<b>Actor</b>	<b>Sistema</b>	
Visualizar la relación de la severidad, incidencia y humedad.	<ol style="list-style-type: none"> <li>1. Ingresa al Sitio Web.</li> <li>2. Ingresa a la sección General.</li> <li>3. Establece las configuraciones del gráfico.</li> <li>4. Iniciar la configuración del gráfico con la selección de velocidad del viento como variable del clima.</li> <li>5. Generación del Gráfico.</li> </ol>	

	6. Visualización y Análisis de los gráficos resultado por parte del cliente.
<b>Prioridad</b>	Media.
<b>Cursos alternos</b>	
1. <b>Flujo Alternativo 1:</b> La página no se encuentra disponible. Esto sucedería cuando exista algún problema por parte del servidor o cliente al momento de tratar de acceder a la página.	
2. <b>Flujo Alternativo 2:</b> Error al intentar cargar los datos para la creación del gráfico. Esto sucedería cuando la REST API del Back End no responda la solicitud hecha por el Front End, dando como resultado que no se visualice el gráfico.	

**Tabla 37:** Especificación Caso de Uso Humedad Relativa - Estaciones

<b>Nombre</b>	Humedad Relativa
<b>Actores</b>	Cliente
<b>Propósito</b>	Establecer la capacidad que brinda el sistema al usuario de seleccionar que elemento del clima desea visualizar en el resultado.
<b>Resumen</b>	
El cliente puede visualizar en un gráfico estadístico el comportamiento de la humedad relativa y la severidad en función de las estaciones.	
<b>Casos de uso asociados</b>	El presente caso de uso es un extend del caso de uso <b>Elementos del Clima</b> .
<b>Flujo de Trabajo</b>	
<b>Actor</b>	<b>Sistema</b>
Visualizar la relación de la severidad y humedad relativa en función de las estaciones.	<ol style="list-style-type: none"> <li>1. Ingresa al Sitio Web.</li> <li>2. Ingresar a la Sección Estaciones.</li> <li>3. Seleccionar la opción todas las estaciones o individual.</li> </ol>

	<p>4. Iniciar la configuración del gráfico con la selección de humedad relativa como variable del clima.</p> <p>5. Finalizar la configuración del gráfico.</p> <p>6. Generación del Gráfico.</p> <p>7. Visualización y análisis por parte del cliente.</p>
<b>Prioridad</b>	Media.
<b>Cursos alternos</b>	
<p>1. <b>Flujo Alterno 1:</b> La página no se encuentra disponible. Esto sucedería cuando exista algún problema por parte del servidor o cliente al momento de tratar de acceder a la página.</p> <p>2. <b>Flujo Alterno 2:</b> Error al intentar cargar los datos para la creación del gráfico. Esto sucedería cuando la REST API del Back End no responda la solicitud hecha por el Front End, dando como resultado que no se visualice el gráfico.</p>	

**Tabla 38:** Especificación Caso de Uso Dirección del Viento - Estaciones

<b>Nombre</b>	Dirección del Viento
<b>Actores</b>	Cliente
<b>Propósito</b>	Establecer la capacidad que brinda el sistema al usuario de seleccionar que elemento del clima desea visualizar en el resultado.
<b>Resumen</b>	
El cliente puede visualizar en un gráfico estadístico el comportamiento de la dirección del viento y la severidad en función de las estaciones.	
<b>Casos de uso asociados</b>	El presente caso de uso es un extend del caso de uso <b>Elementos del Clima</b> .
<b>Flujo de Trabajo</b>	
<b>Actor</b>	<b>Sistema</b>

Visualizar la relación de la severidad y dirección del viento en función de las estaciones.	<ol style="list-style-type: none"> <li>1. Ingresar al Sitio Web.</li> <li>2. Ingresar a la Sección Estaciones.</li> <li>3. Seleccionar la opción todas las estaciones o individual.</li> <li>4. Iniciar la configuración del gráfico con la selección de dirección del viento como variable del clima.</li> <li>5. Finalizar la configuración del gráfico.</li> <li>6. Generación del Gráfico.</li> <li>7. Visualización y análisis por parte del cliente.</li> </ol>
<b>Prioridad</b>	Media.
<b>Cursos alternos</b>	
<ol style="list-style-type: none"> <li>1. <b>Flujo Alternativo 1:</b> La página no se encuentra disponible. Esto sucedería cuando exista algún problema por parte del servidor o cliente al momento de tratar de acceder a la página.</li> <li>2. <b>Flujo Alternativo 2:</b> Error al intentar cargar los datos para la creación del gráfico. Esto sucedería cuando la REST API del Back End no responda la solicitud hecha por el Front End, dando como resultado que no se visualice el gráfico.</li> </ol>	

**Tabla 39:** Especificación Caso de Uso Análisis de Comportamiento (Severidad) - Estaciones

<b>Nombre</b>	Análisis de Comportamiento
<b>Actores</b>	Cliente
<b>Propósito</b>	Establecer la capacidad del sistema de mostrar los monitoreos presentes en quince días.
<b>Resumen</b>	
El cliente puede visualizar el comportamiento de la severidad según los monitoreos presentes en los quince días.	
<b>Casos de uso asociados</b>	El caso de uso <b>Monitoreos</b> , es un extend del presente caso de uso, ya que especifica

	que el mismo análisis se realiza en función de los monitoreos.
<b>Flujo de Trabajo</b>	
<b>Actor</b>	<b>Sistema</b>
Visualizar el comportamiento de la severidad en función de los monitoreos presentes en el periodo quincenal.	<ol style="list-style-type: none"> <li>1. Ingresar al Sitio Web.</li> <li>2. Ingresar a la sección Estaciones.</li> <li>3. Seleccionar la opción todos los resultados o individual.</li> <li>4. Configuración del gráfico.</li> <li>5. Generación del gráfico.</li> <li>6. Visualización y análisis por parte del cliente.</li> </ol>
<b>Prioridad</b>	Alta.
<b>Cursos alternos</b>	
<ol style="list-style-type: none"> <li>1. <b>Flujo Alternativo 1:</b> La página no se encuentra disponible. Esto sucedería cuando exista algún problema por parte del servidor o cliente al momento de tratar de acceder a la página.</li> <li>2. <b>Flujo Alternativo 2:</b> Error al intentar cargar los datos para la creación del gráfico. Esto sucedería cuando la REST API del Back End no responda la solicitud hecha por el Front End, dando como resultado que no se visualice el gráfico.</li> <li>3. <b>Flujo Alternativo 3:</b> Error al desplegar el gráfico resultado. Esto puede ocurrir cuando el usuario intente acceder al detalle de un mes en específico, donde no se logre visualizar ningún resultado y los motivos pueden estar relacionados al manejo de JAVASCRIPT, HTML o CSS.</li> </ol>	

**Tabla 40:** Especificación Caso de Uso Monitoreos - Estaciones

<b>Nombre</b>	Monitoreos
<b>Actores</b>	Cliente
<b>Propósito</b>	Establecer la capacidad del sistema de clasificar los resultados en función de los monitoreos.
<b>Resumen</b>	

<p>El cliente puede visualizar un gráfico estadístico con el comportamiento de la severidad, clasificando los resultados en función de los monitoreos presentes en el periodo quincenal.</p>	
<p><b>Casos de uso asociados</b></p>	<p>Es presente caso de uso extend del caso de uso <b>Análisis Comportamiento</b>.</p>
<p><b>Flujo de Trabajo</b></p>	
<p><b>Actor</b></p>	<p><b>Sistema</b></p>
<p>Visualizar el comportamiento de la severidad de los diferentes monitoreos realizados en un periodo quincenal.</p>	<ol style="list-style-type: none"> <li>1. Ingresar al Sitio Web.</li> <li>2. Ingresar a la Sección de Estaciones.</li> <li>3. Seleccionar la opción de todas las estaciones o individual.</li> <li>4. Configuración del Gráfico.</li> <li>5. Generación del gráfico con los resultados clasificados en periodos de quince días.</li> <li>6. Selección de uno de los resultados.</li> <li>7. Generación del gráfico con los resultados clasificados en función de los monitoreos.</li> <li>8. Visualización y análisis por parte del cliente.</li> </ol>
<p><b>Prioridad</b></p>	<p>Media.</p>
<p><b>Cursos alternos</b></p> <ol style="list-style-type: none"> <li>1. <b>Flujo Alternativo 1:</b> La página no se encuentra disponible. Esto sucedería cuando exista algún problema por parte del servidor o cliente al momento de tratar de acceder a la página.</li> <li>2. <b>Flujo Alternativo 2:</b> Error al intentar cargar los datos para la creación del gráfico. Esto sucedería cuando la REST API del Back End no responda la solicitud hecha por el Front End, dando como resultado que no se visualice el gráfico.</li> </ol>	

**Tabla 41:** Especificación Caso de Uso Análisis de Comportamiento (Elementos del Clima) - Estaciones

<b>Nombre</b>	Análisis de Comportamiento	
<b>Actores</b>	Cliente	
<b>Propósito</b>	Establecer la capacidad del sistema de mostrar detalladamente las semanas presentes en el periodo mensual.	
<b>Resumen</b>		
El cliente puede visualizar el comportamiento de la severidad en función de las semanas presentes en el periodo mensual.		
<b>Casos de uso asociados</b>	El caso de uso <b>Periodo Semanal</b> , es un extend del presente caso de uso, ya que especifica que el mismo análisis se realiza en función de un periodo semanal.	
<b>Flujo de Trabajo</b>		
<b>Actor</b>	<b>Sistema</b>	
Visualizar el comportamiento de la severidad en función de las semanas presentes en el periodo mensual.	<ol style="list-style-type: none"> <li>1. Ingresar al Sitio Web.</li> <li>2. Ingresar a la sección Estaciones.</li> <li>3. Seleccionar la opción todos los resultados o individual.</li> <li>4. Configurar del gráfico.</li> <li>5. Generar el gráfico.</li> <li>6. Visualizar y analizar por parte del cliente.</li> </ol>	
<b>Prioridad</b>	Alta.	
<b>Cursos alternos</b>		
<ol style="list-style-type: none"> <li>1. <b>Flujo Alterno 1:</b> La página no se encuentra disponible. Esto sucedería cuando exista algún problema por parte del servidor o cliente al momento de tratar de acceder a la página.</li> <li>2. <b>Flujo Alterno 2:</b> Error al intentar cargar los datos para la creación del gráfico. Esto sucedería cuando la REST API del Back End no responda la solicitud hecha por el Front End, dando como resultado que no se visualice el gráfico.</li> </ol>		

3. **Flujo Alternativo 3:** Error al desplegar el gráfico resultado. Esto puede ocurrir cuando el usuario intente acceder al detalle de un mes en específico, donde no se logre visualizar ningún resultado y los motivos pueden estar relacionados al manejo de JAVASCRIPT, HTML o CSS.

### Especificaciones de Caso de uso del Diagrama Severidad

**Tabla 42:** Especificación Caso de Uso Análisis Descriptivo Severidad - Severidad

<b>Nombre</b>	Análisis Descriptivo Severidad	
<b>Actores</b>	Cliente	
<b>Propósito</b>	Establecer la variable de análisis severidad.	
<b>Resumen</b>		
El cliente puede analizar la severidad en un tiempo concreto.		
<b>Casos de uso asociados</b>	Este caso de uso tiene relación de <b>inclusión</b> con el siguiente caso de uso: <ul style="list-style-type: none"> <li>• <b>Tiempo:</b> Este caso de uso es importante debido a que, si se selecciona un tiempo específico, esta muestra la información de la variable de severidad en función del periodo seleccionado.</li> </ul>	
<b>Flujo de Trabajo</b>		
<b>Actor</b>	<b>Sistema</b>	
Visualizar el comportamiento de la severidad.	<ol style="list-style-type: none"> <li>1. Ingresar al Sitio Web.</li> <li>2. Ingresar a la sección Severidad con opción General.</li> <li>3. Establecer las configuraciones del gráfico.</li> <li>4. Generar del Gráfico.</li> <li>5. Visualizar y analizar de los gráficos resultado por parte del cliente.</li> </ol>	
<b>Prioridad</b>	Alta.	
<b>Cursos alternos</b>		

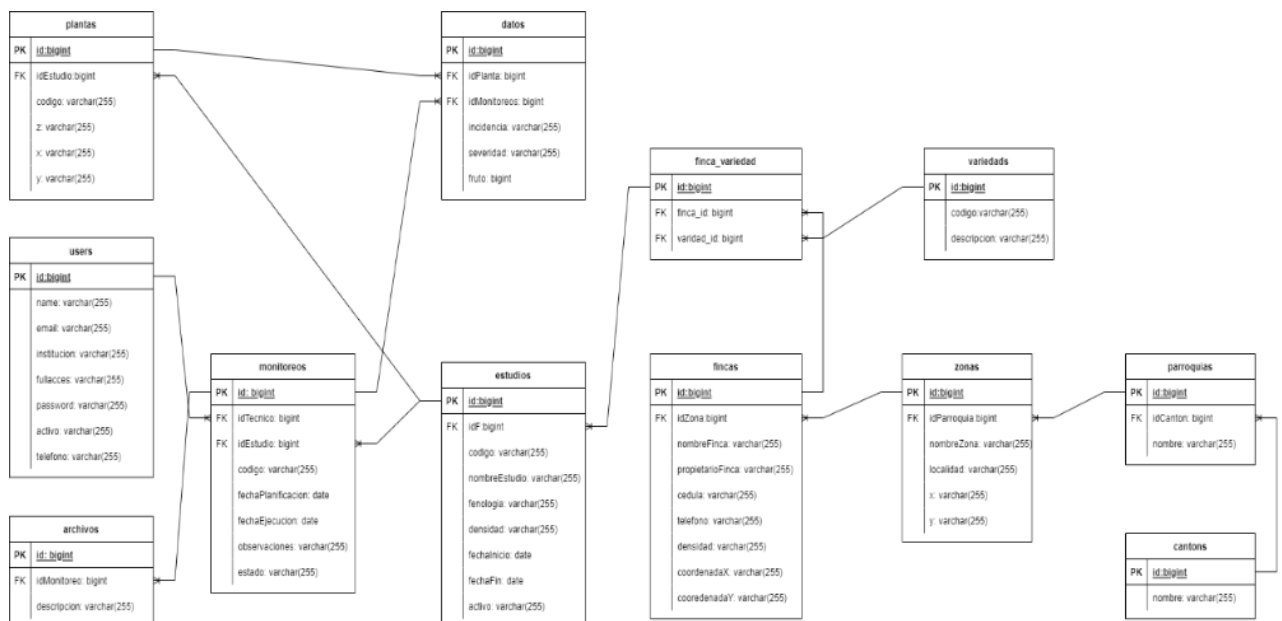
3. **Flujo Alternativo 1:** La página no se encuentra disponible. Esto sucedería cuando exista algún problema por parte del servidor o cliente al momento de tratar de acceder a la página.
4. **Flujo Alternativo 2:** Error al intentar cargar los datos para la creación del gráfico. Esto sucedería cuando la REST API del Back End no responda la solicitud hecha por el Front End, dando como resultado que no se visualice el gráfico.

### 3.2 Análisis

#### Esquema Conceptual de las Fuentes de Datos

#### Base de Datos Relacional Estudios

La primera fuente de datos considerada para el desarrollo del Data Warehouse corresponde a un sistema transaccional que tiene el objetivo de almacenar datos relacionados a la severidad e incidencia. Lo que almacena la base de datos relacional son datos sobre las diferentes ubicaciones de los cultivos de cacao, los estudios y monitoreos que se realizan a dichos cultivos, datos sobre las plantas y sus ubicaciones, juntamente con los datos correspondientes de la severidad e incidencia. La base de datos posee el siguiente modelo relacional:



**Figura 11: Modelo Relacional - Estudios**

Con base en el modelo anterior, se pudo realizar el siguiente conceptual para detallar la estructura de los datos a considerar.

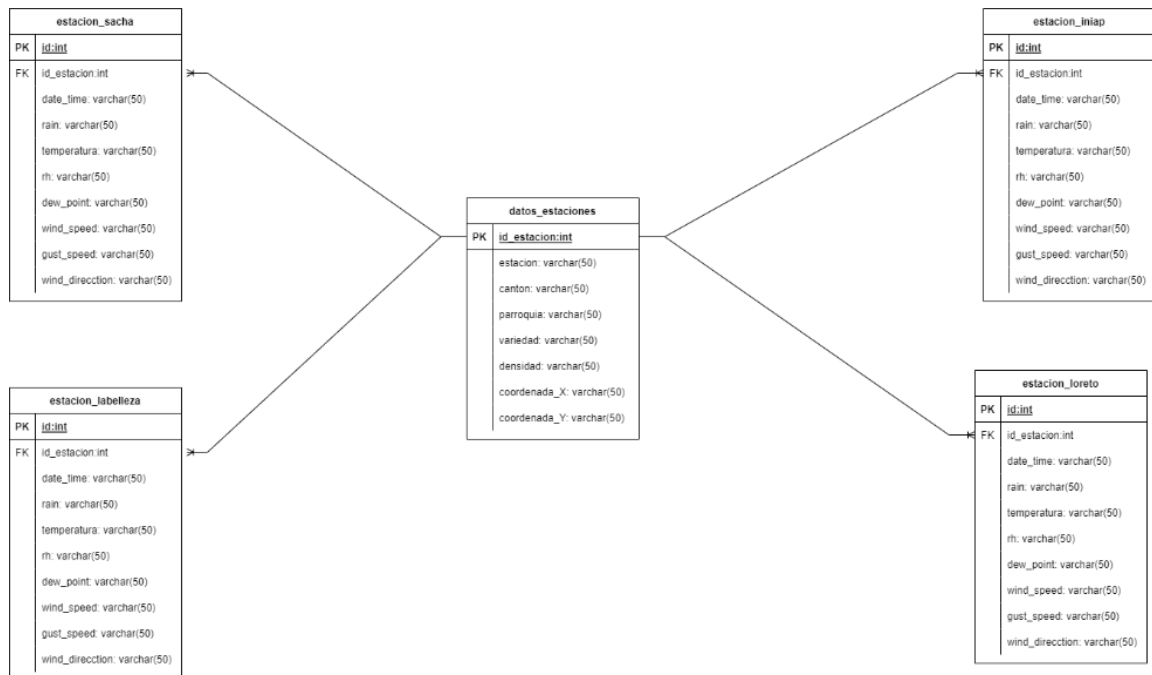
Las tablas consideradas de la base de datos relacional son: datos, plantas, monitoreos, estudios, fincas y variedades; resultan ser tablas de suma importancia que proporcionan datos descriptivos o medidas relacionadas a la severidad e incidencia.



**Figura 12:** Esquema conceptual de la fuente de datos – Base De Datos Relacional  
Estudios

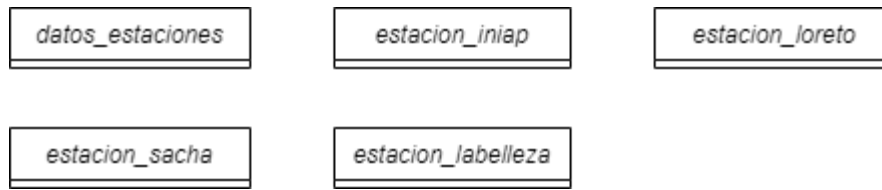
### Base de Datos Relacional Estaciones

La segunda fuente de datos a considera para el desarrollo del Data Warehouse corresponde a una base de datos relaciona que tiene el objetivo de almacenar datos climatológicos de las diferentes estaciones, juntamente con datos descriptivos para cada una de las estaciones.



**Figura 13:** Modelo Relacional - Estaciones

Con base en el modelo anterior, se pudo realizar el siguiente conceptual para detallar la estructura de los datos a considerar.



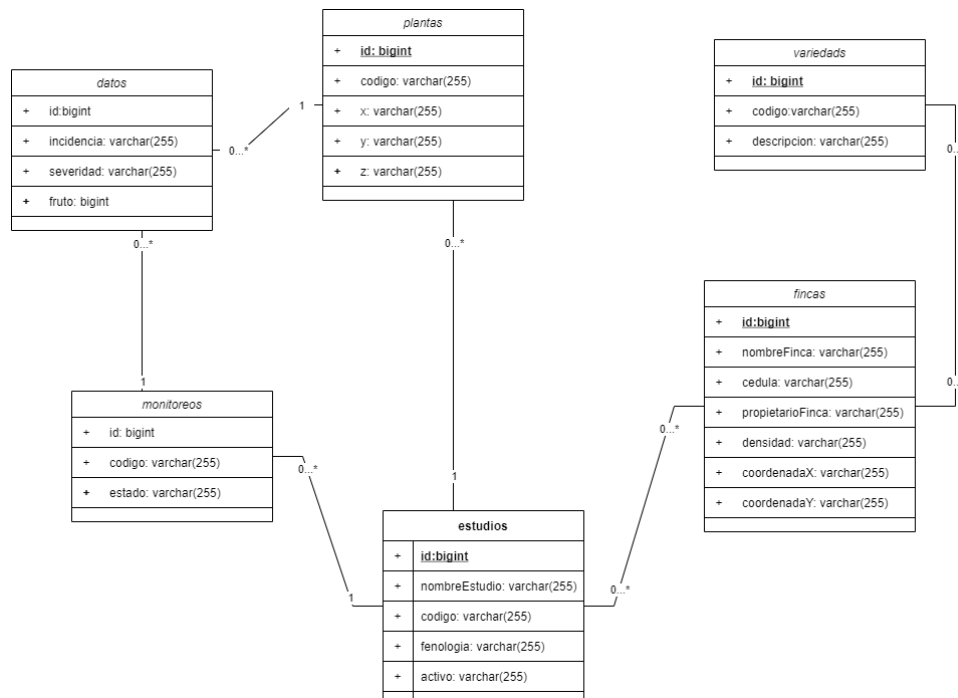
**Figura 14:** Esquema conceptual de la fuente de datos – Base de Datos Relacional

Estaciones

**Esquema Lógico de las Fuentes de Datos**

**Base de Datos Relacional Estudios**

Con base en el análisis del esquema conceptual del modelo relacional, se obtuvo el siguiente diagrama lógico representado por un diagrama de clases. Este diagrama tiene la finalidad de mostrar las tablas que serán consideradas para alimentar el Data Warehouse.



**Figura 15:** Esquema lógico de la fuente de datos – Base de Datos Relacional Estaciones

**Diccionario de Datos**

**Tabla 43:** Descripción de la tabla fincas

<b>Tabla</b>	fincas	
<b>Descripción</b>	Esta tabla guarda información sobre las diferentes fincas que cuentan con cultivos de cacao.	
<b>Campo</b>	<b>Tipo de dato</b>	<b>Descripción</b>

id	bigint	Identificador único autogenerado por la Base de Datos.
nombreFinca	varchar(255)	Corresponde al nombre con el cual se identifica a la finca.
Cedula	varchar(255)	Este campo almacena el número de identificación del propietario de la finca.
propietarioFinca	varchar(255)	Nombre del propietario de la finca.
Densidad	varchar(255)	Población del terreno agrícola (cantidad de plantas por área de terreno).
coordenadaX	varchar(255)	Ubicación de la finca en la coordenada de longitud.
coordenadaY	varchar(255)	Ubicación de la finca en la coordenada de latitud.

**Tabla 44:** Descripción de la tabla variedades

<b>Tabla</b>	variedades	
<b>Descripción</b>	Contiene información sobre las distintas variedades o tipos de cultivos que se pueden realizar. Tiene el objetivo de establecer la o las variedades utilizadas en una finca en específico.	
<b>Campo</b>	<b>Tipo de dato</b>	<b>Descripción</b>
id	bigint	Identificador único autogenerado por la Base de Datos.
codigo	varchar(255)	Código único establecido manualmente para cada variedad.
descripcion	varchar(255)	Nombre de la variedad de cultivo.

**Tabla 45:** Descripción de la tabla estudios

<b>Tabla</b>	estudios
<b>Descripción</b>	Almacena datos sobre los diferentes estudios realizados en las plantaciones.

<b>Campo</b>	<b>Tipo de dato</b>	<b>Descripción</b>
id	bigint	Identificador único autogenerado por la Base de Datos.
nombreEstudio	varchar(255)	Nombre asignado al estudio.
codigo	varchar(255)	Código único, asignado manualmente, al estudio.
fenologia	varchar(255)	Almacena datos relacionados al ciclo en que se encuentran las diferentes plantas.
activo	varchar(255)	Establece si el estudio se encuentra activo o finalizado.

**Tabla 46:** Descripción de la tabla monitoreos

<b>Tabla</b>	monitoreos	
<b>Descripción</b>	Almacena datos sobre los monitoreos realizados en los diferentes estudios.	
<b>Campo</b>	<b>Tipo de dato</b>	<b>Descripción</b>
id	bigint	Identificador único autogenerado por la Base de Datos.
codigo	varchar(255)	Código único establecido manualmente para cada monitoreo.
estado	varchar(255)	Establece si el monitoreo se encuentra activo o finalizado.
observaciones	varchar(255)	Establece una descripción del monitoreo realizado.

**Tabla 47:** Descripción de la tabla plantas

<b>Tabla</b>	plantas	
<b>Descripción</b>	Almacena datos descriptivos sobre las diferentes plantas presentes en un cultivo.	

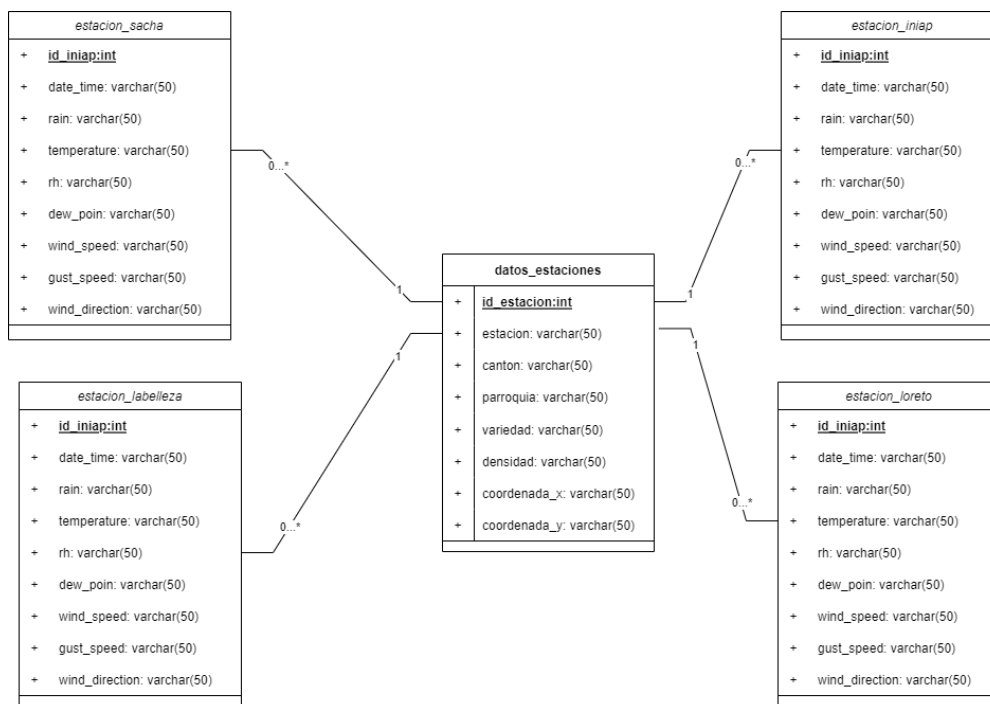
<b>Campo</b>	<b>Tipo de dato</b>	<b>Descripción</b>
id	bigint	Identificador único autogenerado por la Base de Datos.
codigo	varchar(255)	Código único establecido manualmente para cada planta.
x	varchar(255)	Ubicación de la planta en la coordenada X.
y	varchar(255)	Ubicación de la planta en la coordenada Y.
z	varchar(255)	Ubicación de la planta en la coordenada Z.

**Tabla 48:** Descripción de la tabla datos

<b>Tabla</b>	datos	
<b>Descripción</b>	Almacena los datos relacionados al porcentaje de severidad e incidencia de las diferentes plantas.	
<b>Campo</b>	<b>Tipo de dato</b>	<b>Descripción</b>
id	bigint	Identificador único autogenerado por la Base de Datos.
incidencia	varchar(255)	Porcentaje de presencia de la enfermedad.
severidad	varchar(255)	Porcentaje de avance de la enfermedad en el fruto.
fruto	bigint	Representas un fruto de la planta.

### **Base de Datos Relacional Estaciones**

El modelo lógico de la Base de Datos Relacional fue representado a través del siguiente diagrama de clases.



**Figura 16:** Esquema lógico de la fuente de datos – Base de Datos Relacional Estaciones.

## Diccionario de Datos

**Tabla 49:** Descripción de la tabla estacion\_iniap

Tabla	estacion_iniap	
Descripción	Almacena datos meteorológicos tomados por la estación.	
Campo	Tipo de dato	Descripción
date_time	varchar(50)	Fecha y hora en la que se toma la medida meteorológica de cada uno de los campos.
rain	varchar(50)	Precipitación de la atmosfera tomada por el sensor.
temperature	varchar(50)	Temperatura de un sensor de la estación.
rh	varchar(50)	Humedad relativa tomada por el sensor.
Dew_point	varchar(50)	Punto de Rocío tomado por el sensor.
wind_speed	varchar(50)	Velocidad del viento tomada por el sensor.
gust_speed	varchar(50)	Ráfaga de viento tomada por el sensor.
wind_direction	varchar(50)	Dirección en la que se movía el viento tomado por el sensor de la estación.

**Tabla 50:** Descripción de la tabla estacion\_loreto

<b>Tabla</b>	estacion_loreto	
<b>Descripción</b>	Almacena datos meteorológicos tomados por la estación.	
<b>Campo</b>	<b>Tipo de dato</b>	<b>Descripción</b>
date_time	varchar(50)	Fecha y hora en la que se toma la medida meteorológica de cada uno de los campos.
rain	varchar(50)	Precipitación de la atmosfera tomada por el sensor.
temperature	varchar(50)	Temperatura de un sensor de la estación.
rh	varchar(50)	Humedad relativa tomada por el sensor.
Dew_point	varchar(50)	Punto de Rocío tomado por el sensor.
wind_speed	varchar(50)	Velocidad del viento tomada por el sensor.
gust_speed	varchar(50)	Ráfaga de viento tomada por el sensor.
wind_direction	varchar(50)	Dirección en la que se movía el viento tomado por el sensor de la estación.

**Tabla 51:** Descripción de la tabla estacion\_sacha

<b>Tabla</b>	estacion_sacha	
<b>Descripción</b>	Almacena datos meteorológicos tomados por la estación.	
<b>Campo</b>	<b>Tipo de dato</b>	<b>Descripción</b>
date_time	varchar(50)	Fecha y hora en la que se toma la medida meteorológica de cada uno de los campos.
rain	varchar(50)	Precipitación de la atmosfera tomada por el sensor.
temperature	varchar(50)	Temperatura de un sensor de la estación.
rh	varchar(50)	Humedad relativa tomada por el sensor.
Dew_point	varchar(50)	Punto de Rocío tomado por el sensor.
wind_speed	varchar(50)	Velocidad del viento tomada por el sensor.
gust_speed	varchar(50)	Ráfaga de viento tomada por el sensor.

wind_direction	varchar(50)	Dirección en la que se movía el viento tomado por el sensor de la estación.
----------------	-------------	---

**Tabla 52:** Descripción de la tabla estacion\_labelleza

<b>Tabla</b>	estacion_labelleza	
<b>Descripción</b>	Almacena datos meteorológicos tomados por la estación.	
<b>Campo</b>	<b>Tipo de dato</b>	<b>Descripción</b>
date_time	varchar(50)	Fecha y hora en la que se toma la medida meteorológica de cada uno de los campos.
rain	varchar(50)	Precipitación de la atmosfera tomada por el sensor.
temperature	varchar(50)	Temperatura de un sensor de la estación.
rh	varchar(50)	Humedad relativa tomada por el sensor.
Dew_point	varchar(50)	Punto de Rocío tomado por el sensor.
wind_speed	varchar(50)	Velocidad del viento tomada por el sensor.
gust_speed	varchar(50)	Ráfaga de viento tomada por el sensor.
wind_direction	varchar(50)	Dirección en la que se movía el viento tomado por el sensor de la estación.

**Tabla 53:** Descripción de la tabla datos\_estaciones

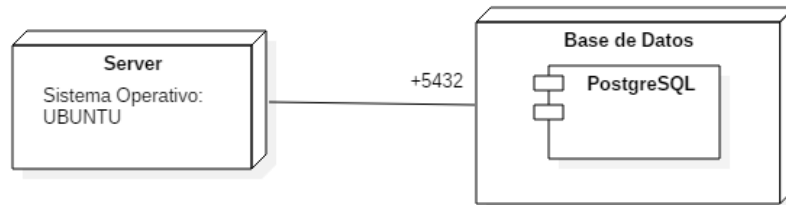
<b>Tabla</b>	datos_estaciones	
<b>Descripción</b>	Almacena datos descriptivos de las estaciones consideradas.	
<b>Campo</b>	<b>Tipo de dato</b>	<b>Descripción</b>
id	int	Identificador numérico de las estaciones.
estacion	varchar(100)	Nombre de la estación.
canton	varchar(100)	Cantón al que pertenece la estación.
parroquia	varchar(100)	Parroquia al que pertenece la estación.
variedad	varchar(100)	Variedad de cultivo presente en la zona donde se encuentra la estación.

densidad	varchar(100)	Población del cultivo presente en la zona donde se encuentra la estación.
coordenada_x	varchar(100)	Ubicación de longitud de la estación.
coordenada_y	varchar(100)	Ubicación de latitud de la estación.

## Esquema Físico de las Fuentes de Datos

### Base de Datos Relacional Estudios

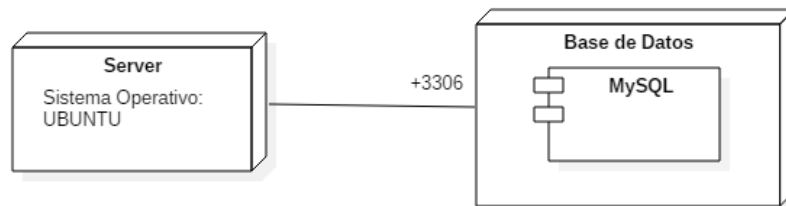
El diagrama físico de base de datos muestra cómo se encuentra desplegada en los servidores del cliente. En este caso la fuente de datos se encuentra desplegada en PostgreSQL dentro de un servidor Ubuntu.



**Figura 17:** Esquema físico de la fuente de datos – Base de Datos Relacional Estaciones

### Base de Datos Relacional Estaciones

El diagrama físico de base de datos muestra cómo se encuentra desplegada en los servidores del cliente. En este caso la fuente de datos se encuentra desplegada en MySQL dentro de un servidor Ubuntu.



**Figura 18:** Esquema físico de la fuente de datos – Base de Datos Relacional Estaciones.

## Reglas de Negocio

Para definir las Reglas de Negocio se realizó un análisis manual de la calidad de los datos, donde se evaluaron las siguientes métricas:

- **Compleitud:** Esta métrica va enfocada a validar que no existan datos vacíos o nulos.
- **Validez:** Esta métrica va enfocada a validar que los datos mantengan un mismo formato o características entre ellos.

- **Unicidad:** Esta métrica valida que los campos correspondientes sean únicos, o sea, no existan datos repetidos según la naturaleza del campo.
- **Coherencia:** Esta métrica va enfocada a validar que los datos vayan acorde al objetivo que tiene el campo.
- **Representación:** Esta métrica va enfocada a validar que los datos cumplan con un formato en específico.
- **Precisión:** Esta métrica va enfocada a validar que los datos representen adecuadamente la realidad, es decir, que en su contexto sean reales.

**Tabla 54:** Análisis de la calidad de los datos

Métrica	Problemas Encontrados					Descripción	Solución
	Tabla	Columna	Cantidad Errores	Cantidad Total Filas	Error (%)		
Compleitud	monitoreos	observaciones	190	214	88,79	Datos nulos	Reemplazar el valor por: "Sin observaciones"
	Plantas	z	70	70	100	Datos nulos	Reemplazar el valor por: "0", basado en los valores por defecto de los campos "x" e "y"
Validez	Datos	incidencia	1294	19116	6,77	El campo de tipo numérico es llenado con una cadena de texto: "No existe frutos"	Para el análisis únicamente resultan relevante la existencia de un fruto, es por ello que para solucionar el problema se eliminaron los datos
		severidad	1294	19116	6,77	Este campo representa un porcentaje y se utiliza el valor de -1, para simbolizar la inexistencia de fruto, lo	Para el análisis únicamente resultan relevante los frutos con severidad, es por ello que

						que provoca que no se acorde al resto del conjunto de datos.	los valores negativos son irrelevantes, por lo que para solucionar el problema se eliminaron los datos
	Estudios	fenologia	1	7	14,29	La mayoría de los datos del campo contiene 3 caracteres, de la siguiente manera: "000", pero únicamente un dato contiene solo un "0". Este campo representa el ciclo de vida de una planta.	Únicamente se deja enunciado el problema
	Fincas	telefono	1	6	16,67	No cumple con el formato de 10 dígitos que mantienen los demás datos	Únicamente se deja enunciado el problema
<b>Unicidad</b>	No se encontraron problemas relacionados con esta métrica						
<b>Coherencia</b>	Datos	incidencia	1294	19116	6,77	El campo tiene como objetivo almacenar los datos relacionados a la presencia o no de la enfermedad contiene datos numéricos y cadenas de caracteres	Eliminar los campos con cadenas de caracteres

<b>Representación</b>	Datos	incidencia	1294	19116	6,77	El campo no cumple con el formato numérico	Eliminar los campos que no cumplan con el formato
	Fincas	telefono	1	6	16,67	El teléfono debe ser de 10 dígitos, el dato en cuestión cuenta con solo 9 dígitos	Únicamente se deja enunciado el problema
		nombrePropietario	1	6	16,67	El nombre del propietario contiene un carácter especial, por lo que no cumple con el formato establecido para el dato. El carácter especial que contiene es el "-".	El problema no es tan complejo por lo que únicamente se deja enunciado el problema
<b>Precisión</b>	Datos	severidad	1294	19116	6,77	Se coloca como valor "-1" para simbolizar que no existe el fruto, pero este campo está orientado a representar el porcentaje de contagio de la enfermedad, por lo que no puede existir un porcentaje negativo	Eliminar los campos con valores negativos
	Fincas	telefono	1	6	16,67	El dato del teléfono no representa un valor real, ya que solo cuenta con 9 dígitos.	Únicamente se deja enunciado el problema

### 3.3 Fase de Elaboración

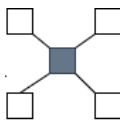
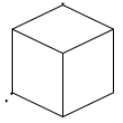
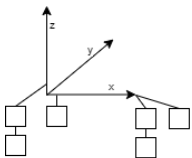
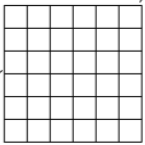
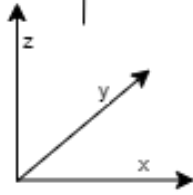

#### Diseño

#### Esquema Conceptual del Data Warehouse

#### Estereotipos multidimensionales

En la siguiente tabla se detallan la simbología UML establecida por la metodología DWEP para el desarrollo del esquema conceptual del Data Warehouse. Dicha simbología es conocida dentro de la metodología como conceptos o estereotipos multidimensionales.

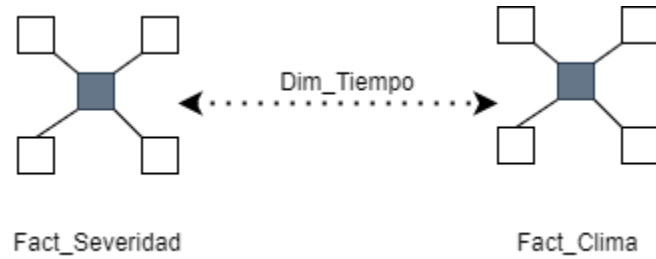
**Tabla 55:** Representación UML de los estereotipos de la metodología DWEP

Nivel	Estereotipo	Descripción	Símbolo
1	StartPackage	Representa de manera simple un esquema de estrella.	
2	FactPackage	Representa de manera simple los hechos de un esquema.	
2	DimensionPackage	Representa de manera simple las dimensiones de un esquema.	
3	Fact	Representa los hechos con sus atributos y relaciones.	
3	Dimension	Representa dimensiones con sus atributos y relaciones.	
3	Base	Representa la jerarquía presente en una dimensión.	

**Fuente:** (Luján & Trujillo, 2003)

### Nivel 1

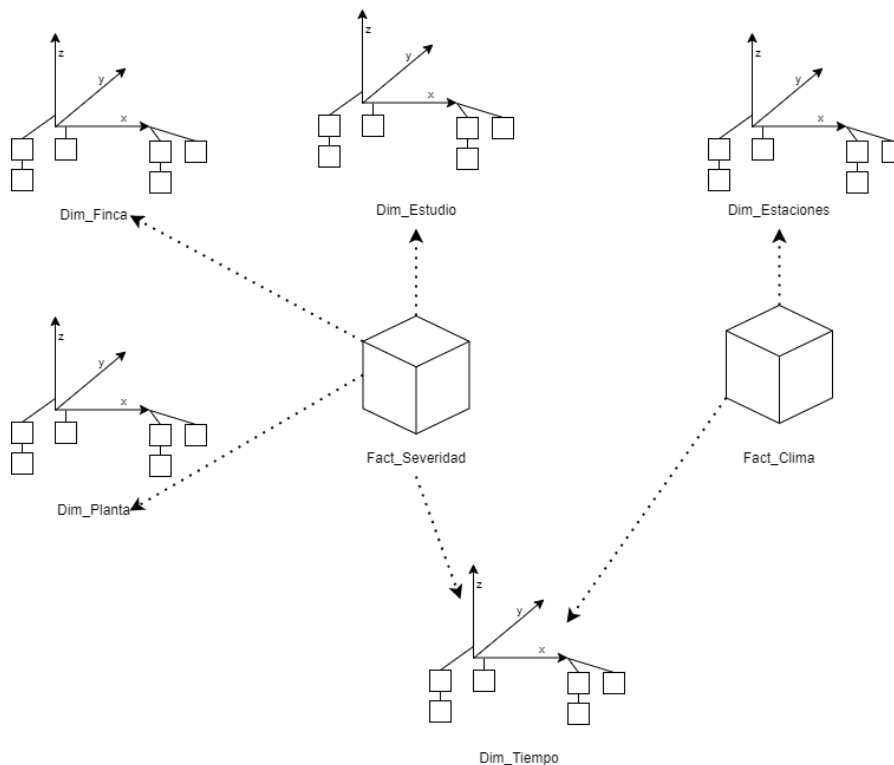
En este nivel se definen los esquemas estrella que forman parte del modelo multidimensional, en la siguiente figura se puede observar una relación entre los hechos: Fact\_Severidad y Fact\_Clima, la cual indica que la Dimensión Tiempo (Dim\_Tiempo), esta compartida entre ambos hechos.



**Figura 19:** Esquema conceptual – Nivel 1

### Nivel 2

En este nivel se describe a detalle los esquemas estrella planteados en el nivel anterior, mediante la representación de los diferentes hechos y dimensiones, juntamente con sus relaciones, presentes en los esquemas del nivel 1.



**Figura 20:** Esquema conceptual – Nivel 2

### Nivel 3

En este nivel se describen los hechos y dimensiones del esquema anterior; principalmente se describen las dimensiones a detalle, para definir los datos que estarán presentes en cada dimensión, junto con su jerarquía.

**Tabla 56:** Simbología de los atributos de los hechos y dimensiones

<b>Símbolo</b>	<b>Descripción</b>
FA	FactAttribute: Representa una medida de los hechos presentes en el esquema.
DA	DimensionAttribute: Representa un atributo de la dimensión.
OID	Representa un atributo de identificación que es generado automáticamente.
D	Descriptor: Son atributos que tienen la finalidad de facilitar el proceso de análisis dimensional; siendo los identificadores únicos utilizados en las fuentes de origen.

**Fuente:** (Mora-Luján, Trujillo, & Song, 2006)

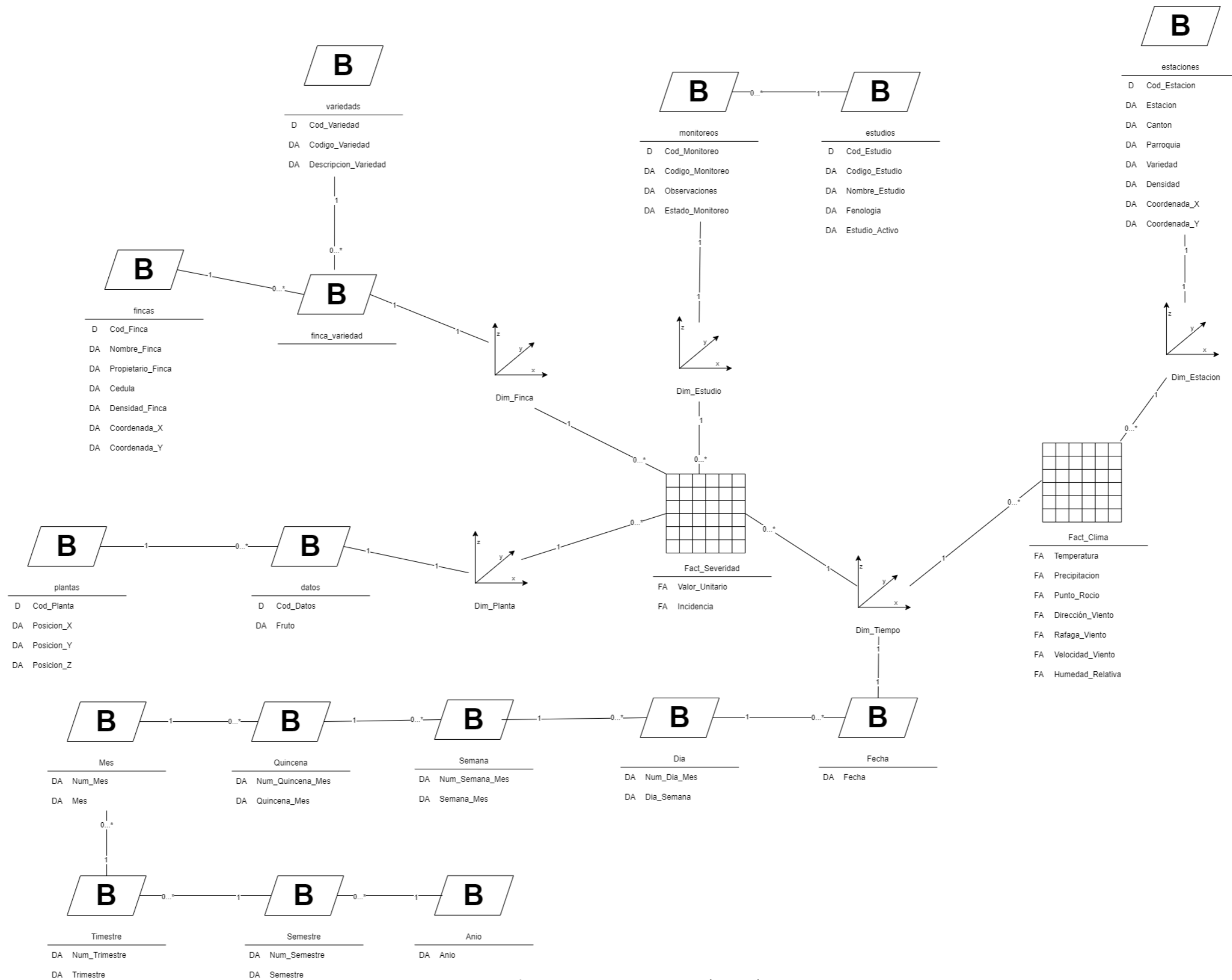


Figura 21: Esquema conceptual – Nivel 3

## Dimensiones

### Dim\_Estudio

Almacena datos sobre los estudios y monitoreos realizados en los diferentes cultivos de cacao considerados para el proyecto.

**Tabla 57:** Jerarquía de la dimensión estudio

<b>Atributo</b>	<b>Nivel</b>	<b>Tipo de Dato</b>	<b>Descripción</b>
ID_Dim_Estudio	1	int	Identificador único para las filas generado automáticamente.
Cod_Estudio	2	int	Identificador empresarial único para las filas de la tabla estudios del modelo ER, generado automáticamente.
Cod_Monitoreo	1	int	Identificador empresarial único para las filas de la tabla monitoreos del modelo ER, generado automáticamente.
Codigo_Estudio	2	varchar(100)	Código asignado manualmente al estudio.
Nombre_Estudio	2	varchar(100)	Nombre asignado al estudio.
Estudio_Activo	2	boolean	Establece si el estudio se encuentra activo o finalizado.
Fenologia	2	varchar(100)	Almacena datos relacionados al ciclo de desarrollo en que se encuentran las diferentes plantas.
Codigo_Monitoreo	1	varchar(100)	Código único establecido manualmente para cada monitoreo.
Observaciones	1	text	Establece una descripción del monitoreo realizado.
Estado_Monitoreo	1	boolean	Establece si el monitoreo se encuentra activo o finalizado.

### Dim\_Finca

Almacena datos sobre las fincas y las variedades de cultivos de cacao que realizan.

**Tabla 58:** Jerarquía de la dimensión finca

<b>Atributo</b>	<b>Nivel</b>	<b>Tipo de Dato</b>	<b>Descripción</b>
ID_Dim_Finca	1	int	Identificador único para las filas generado automáticamente.
Cod_Finca	2	int	Identificador empresarial único para las filas de la tabla fincas del modelo ER, generado automáticamente.
Cod_Varietas	2	int	Identificador empresarial único para las filas de la tabla variedades del modelo ER, generado automáticamente.
Nombre_Finca	2	varchar(100)	Corresponde al nombre con el cual se identifica a la finca.
Cedula	2	varchar(100)	Este campo almacena el número de identificación del propietario de la finca.
Propietario_Finca	2	varchar(100)	Nombre del propietario de la finca.
Densidad_Finca	2	varchar(100)	Población del terreno agrícola (cantidad de plantas por área de terreno).
Coordenada_X	2	varchar(100)	Ubicación de la finca en la coordenada de longitud.
Coordenada_Y	2	varchar(100)	Ubicación de la finca en la coordenada de latitud.
Codigo_Varietas	2	varchar(100)	Código único establecido manualmente para cada variedad.
Descripcion_Varietas	2	varchar(100)	Nombre de la variedad de cultivo.

## Dim\_Plantas

Almacena datos sobre las plantas y sus frutos de los diferentes cultivos de cacao.

**Tabla 59:** Jerarquía de la dimensión planta

<b>Atributo</b>	<b>Nivel</b>	<b>Tipo de Dato</b>	<b>Descripción</b>
ID_Dim_Plantas	1	int	Identificador único para las filas generado automáticamente.
Cod_Planta	2	int	Identificador empresarial único para las filas de la tabla plantas del modelo ER, generado automáticamente.
Cod_Datos	1	int	Identificador empresarial único para las filas de la tabla datos del modelo ER, generado automáticamente.
Fruto	1	int	Representa un fruto de la planta.
Posicion_X	2	varchar(100)	Ubicación de la planta en la coordenada X.
Posicion_Y	2	varchar(100)	Ubicación de la planta en la coordenada Y.
Posicion_Z	2	varchar(100)	Ubicación de la planta en la coordenada Z.

## Dim\_Tiempo

Establece el tiempo para los diferentes datos presentes en los hechos, tales como: año, mes, día... etc.

**Tabla 60:** Jerarquía de la dimensión tiempo

<b>Atributo</b>	<b>Nivel</b>	<b>Tipo de Dato</b>	<b>Descripción</b>
ID_Dim_Tempo	1	int	Identificador único para las filas generado a partir del campo fecha.
Fecha	1	date	Valor de tiempo para establecer un día único.

Anio	8	int	Año obtenido del campo fecha.
Num_Mes	5	int	Número del mes obtenido del campo fecha.
Mes	5	varchar(100)	Nombre del mes obtenido del campo fecha.
Num_Dia_Mes	2	int	Número del día del mes obtenido del campo fecha.
Dia_Semana	2	varchar(100)	Nombre del día de la semana obtenido del campo fecha.
Num_Semestre	7	int	Número del semestre del año obtenido del campo fecha.
Semestre	7	varchar(100)	Nombre del semestre del año obtenido del campo fecha.
Num_Trimestre	6	int	Número del trimestre del año obtenido del campo fecha.
Trimestre	6	varchar(100)	Nombre del trimestre del año obtenido del campo fecha.
Num_Quincena_Mes	4	int	Número de la quincena del mes obtenido del campo fecha.
Quincena_Mes	4	varchar(100)	Nombre de la quincena del mes obtenido del campo fecha.
Num_Semana_Mes	3	Int	Número de la semana del mes obtenido del campo fecha.
Semana_Mes	3	varchar(100)	Nombre de la semana del mes obtenido del campo fecha.

### **Dim\_Estacion**

Almacena datos descriptivos de las diferentes estaciones meteorológicas. Los datos provienen de un origen estático.

**Tabla 61:** Jerarquía de la dimensión estaciones

<b>Atributo</b>	<b>Nivel</b>	<b>Tipo de Dato</b>	<b>Descripción</b>
-----------------	--------------	---------------------	--------------------

ID_Dim_Estacion	1	int	Identificador único para las filas generado a partir del campo fecha.
Cod_Estacion	1	int	Identificador numérico de las estaciones.
Estacion	1	varchar(100)	Nombre de la estación
Canton	3	varchar(100)	Cantón al que pertenece la estación.
Parroquia	2	varchar(100)	Parroquia al que pertenece la estación.
Variedad	1	varchar(100)	Variedad de cultivo presente en la zona donde se encuentra la estación.
Densidad	1	varchar(100)	Población del cultivo presente en la zona donde se encuentra la estación.
Coordenada_X	1	varchar(100)	Ubicación de longitud de la estación.
Coordenada_Y	1	varchar(100)	Ubicación de latitud de la estación.

## Hechos

### Fact\_Severidad

Almacena datos sobre el comportamiento de la severidad e incidencia.

**Tabla 62:** Granularidad del hecho severidad

Nombre de la Dimensión	Atributo	Descripción	Llave Primaria
Dim_Tiempo	ID_Dim_Tiempo	Momento en el tiempo al cual pertenece el registro.	Sí
Dim_Estudio	ID_Dim_Estudio	Estudio en el que se realizaron las mediciones de severidad e incidencia.	Sí

Dim_Planta	ID_Dim_Planta	Planta a la que se le realizaron las mediciones de severidad e incidencia.	Sí
Dim_Finca	ID_Dim_Finca	Finca en la que se realizaron las mediciones de severidad e incidencia.	Sí

**Tabla 63:** Medidas del hecho severidad

Atributo	Tipo de Dato	Descripción
Valor_Unitario	double precision	Porcentaje de severidad presente en un fruto de una planta de cacao.
Incidencia	double precision	Porcentaje de presencia de la enfermedad en un fruto de una planta de cacao.

### Fact\_Clima

Almacena datos sobre el comportamiento de los elementos del clima.

**Tabla 64:** Granularidad del hecho clima

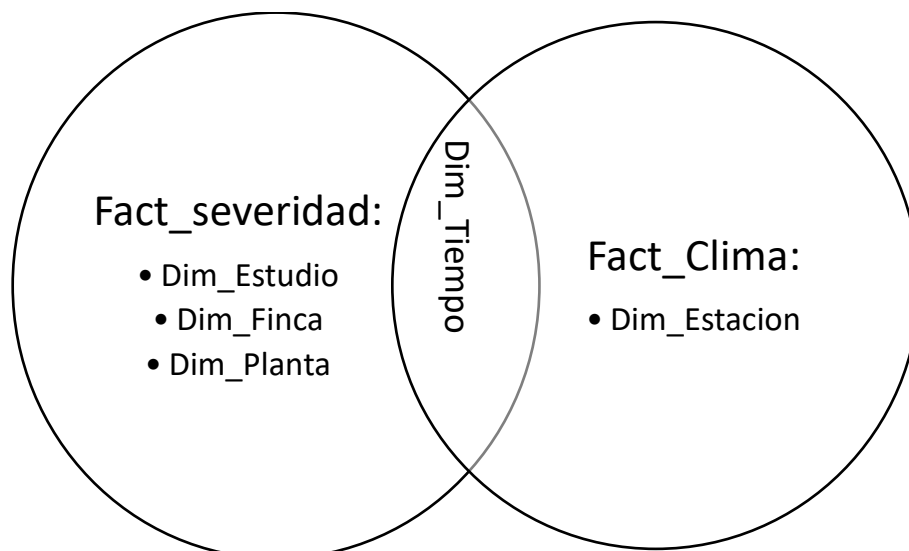
Nombre de la Dimensión	Atributo	Descripción	Llave Primaria
Dim_Tiempo	ID_Dim_Tiempo	Momento en el tiempo al cual pertenece el registro.	Sí
Dim_Estacion	ID_Dim_Estacion	Datos Descriptivos de la estación que realizo la recolección de datos.	Sí

**Tabla 65:** Medidas del hecho clima

Atributo	Tipo de Dato	Descripción
----------	--------------	-------------

Temperatura	double precision	Temperatura de un sensor de la estación.
Precipitacion	double precision	Precipitación de la atmosfera tomada por el sensor.
Direccion_Viento	double precision	Dirección en la que se movía el viento tomado por el sensor de la estación.
Punto_Rocio	double precision	Punto de Roció o Temperatura de Roció tomada por el sensor de la estación.
Rafaga_Viento	double precision	Aumento de velocidad o ráfaga de viento tomada por el sensor.
Velocidad_Viento	double precision	Velocidad del Viento tomada por el sensor.
Humedad_Relativa	double precision	Medida tomada por el sensor de la estación.

### Hechos y Dimensiones



**Figura 22:** Dimensiones vs. Hechos

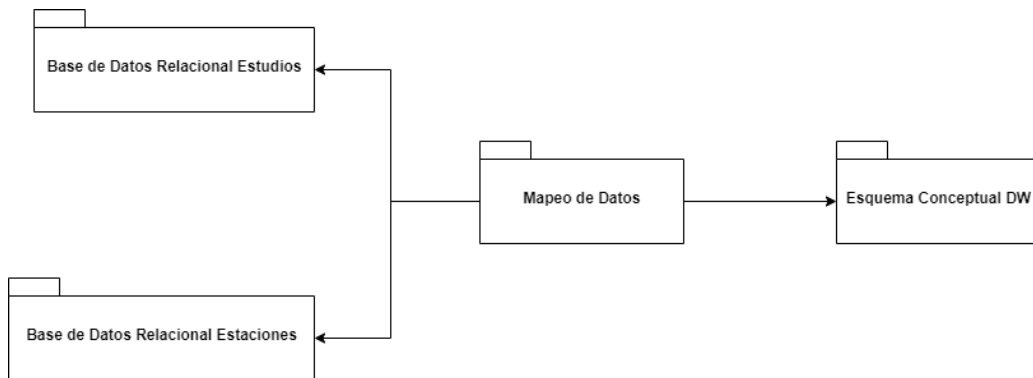
### Diagrama de Mapeo de Datos

Según Duran (2017), el diagrama tiene la finalidad de representar el emparejamiento entre los datos del origen y los datos del destino, es por ello que cuenta con tres elementos:

- **Proveedor:** Representa el origen de los datos.
- **Emparejamiento:** Forma de comunicación entre el origen y el destino.
- **Consumidor:** Representa el destino de los datos, recibe los datos proporcionados por el proveedor mediante emparejamiento.

### Nivel 0

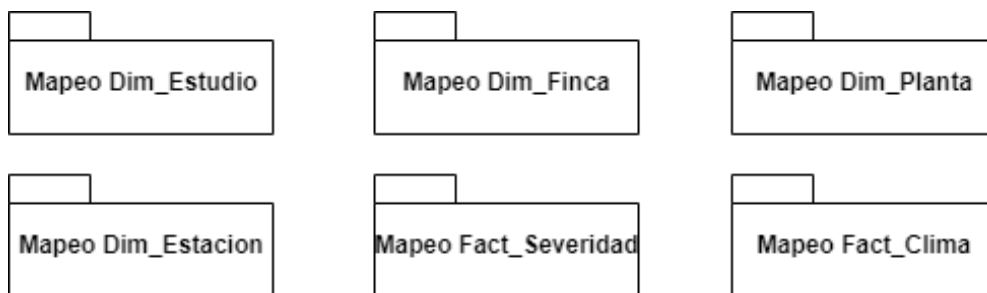
En este nivel se muestra de manera abstracta la relación entre las fuentes de datos y la base de datos del Data Warehouse.



**Figura 23:** Mapeo nivel de base de datos – Nivel 0

### Nivel 1

En este nivel se detalla el paquete de mapeo de datos presente en el nivel 0. En este caso, hay cuatro dimensiones y dos hechos, por lo que existen seis paquetes de mapeo de datos para cada uno.

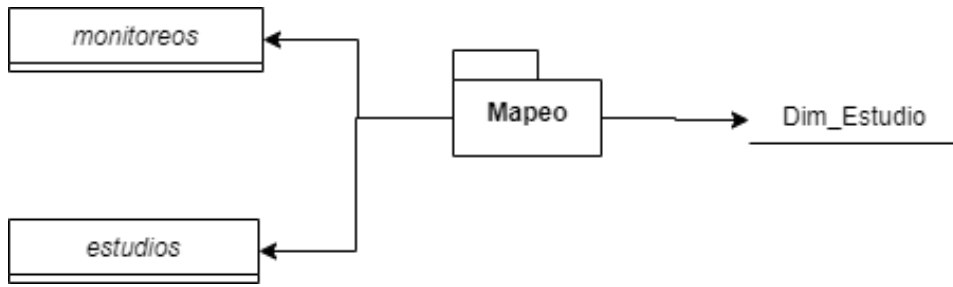


**Figura 24:** Mapeo nivel de flujo de datos – Nivel 1

### Nivel 2

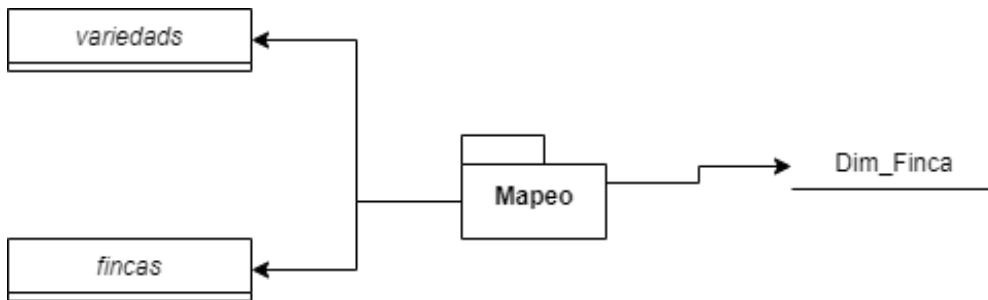
En este nivel se describe brevemente de manera individual todos los procesos de mapeo identificados en el nivel anterior.

#### Dim\_Estudio



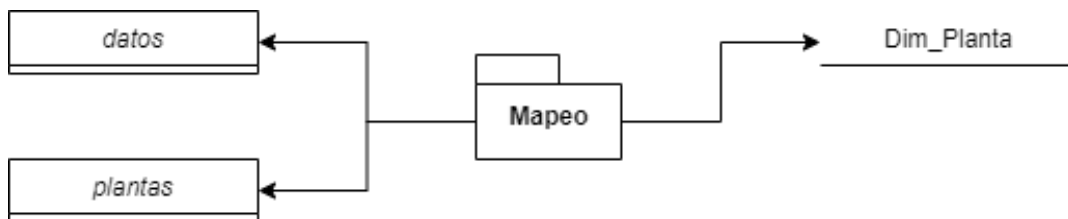
**Figura 25:** Mapeo dimensión estudio – Nivel 2

**Dim\_Finca**



**Figura 26:** Mapeo dimensión finca – Nivel 2

**Dim\_Planta**



**Figura 27:** Mapeo dimensión planta – Nivel 2

**Dim\_Estacion**



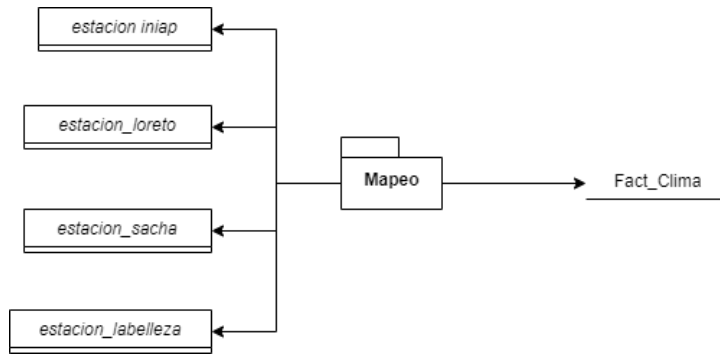
**Figura 28:** Mapeo dimensión estación – Nivel 2

**Fact\_Severidad**



**Figura 29:** Mapeo hecho severidad – Nivel 2

**Fact\_Clima**

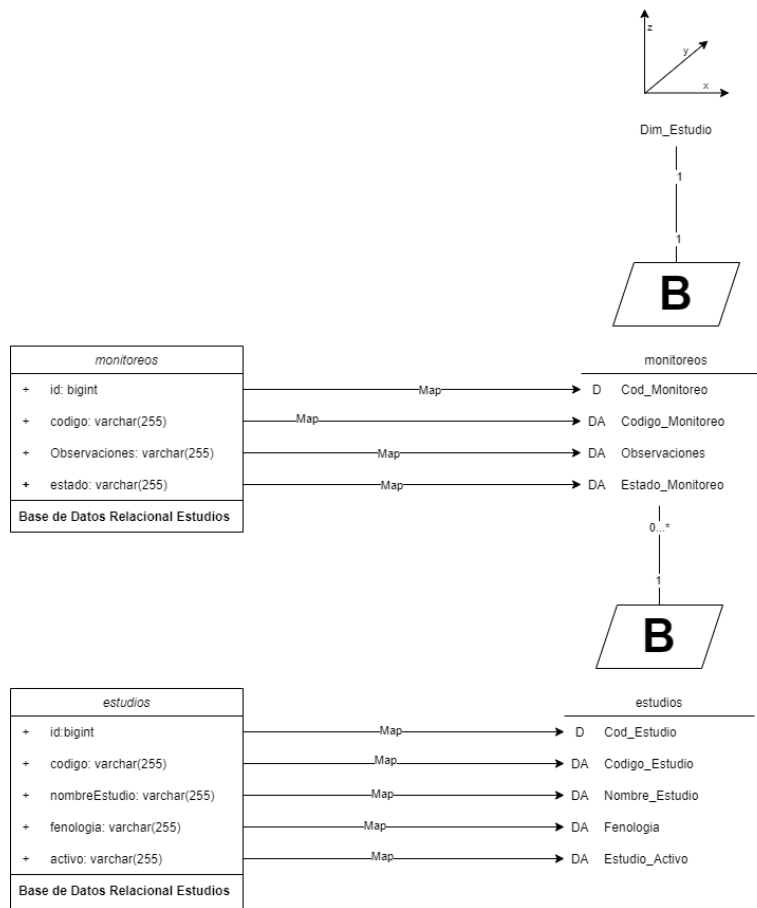


**Figura 30:** Mapeo hecho clima – Nivel 2

**Nivel 3**

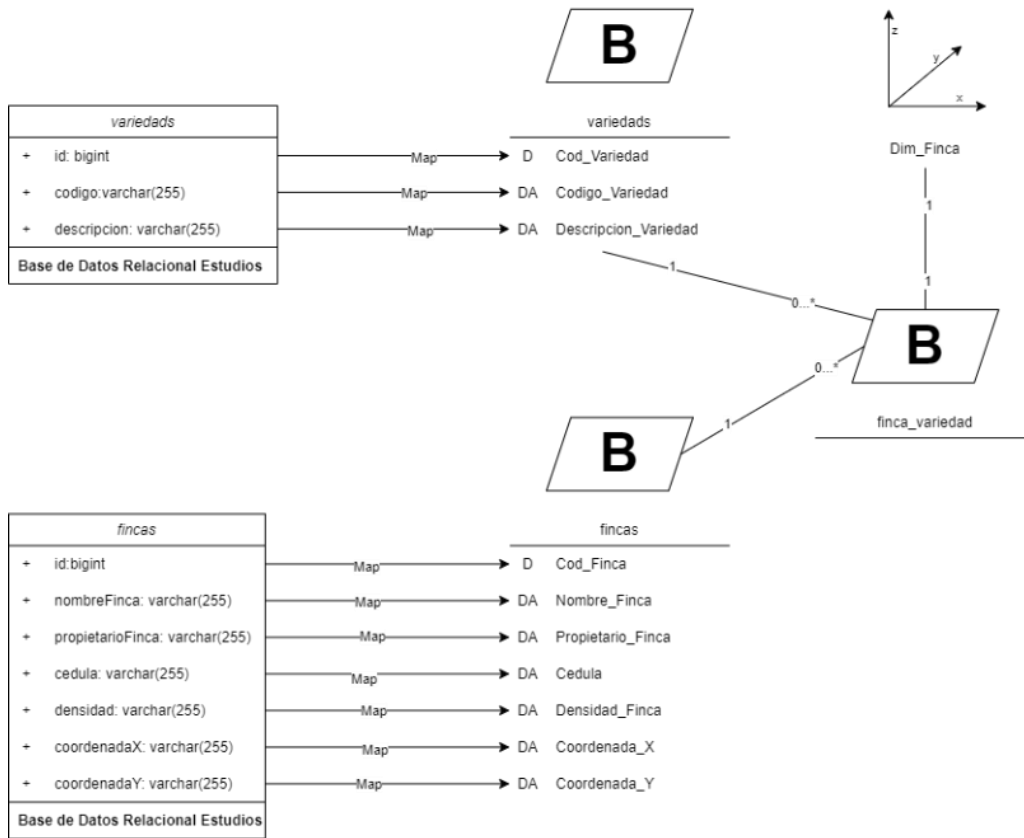
En este nivel se detalla el mapeo de los datos a nivel de atributos.

**Dim\_Estudio**



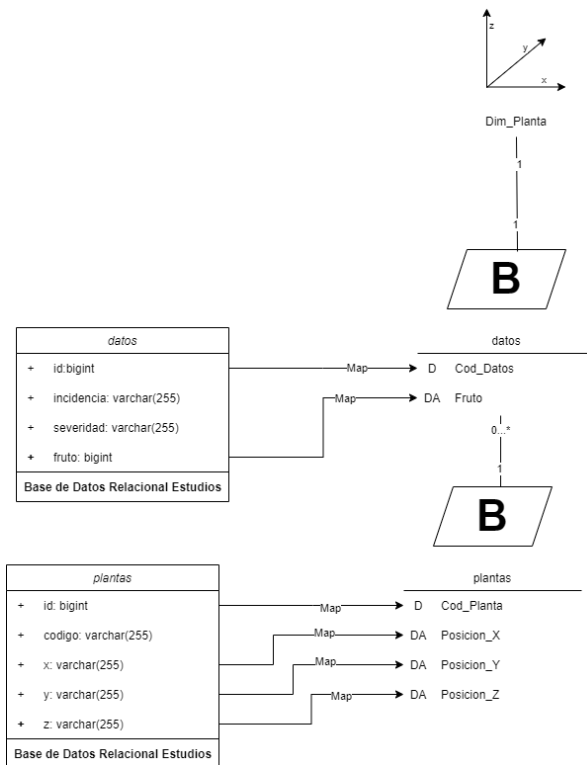
**Figura 31:** Mapeo dimensión estudio – Nivel 3

**Dim\_Finca**



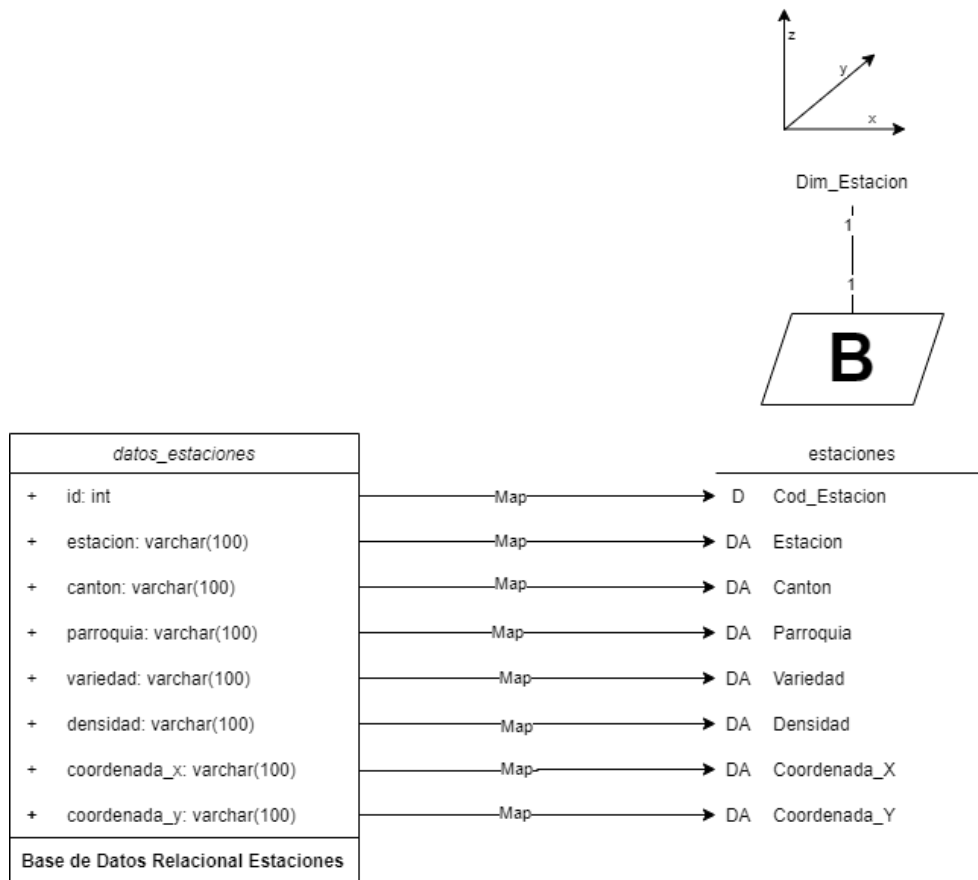
**Figura 32:** Mapeo dimensión finca – Nivel 3

**Dim\_Planta**



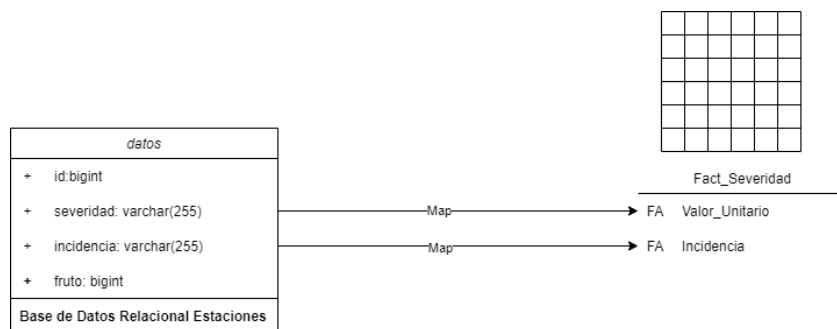
**Figura 33:** Mapeo dimensión planta – Nivel 3

## Dim\_Estacion



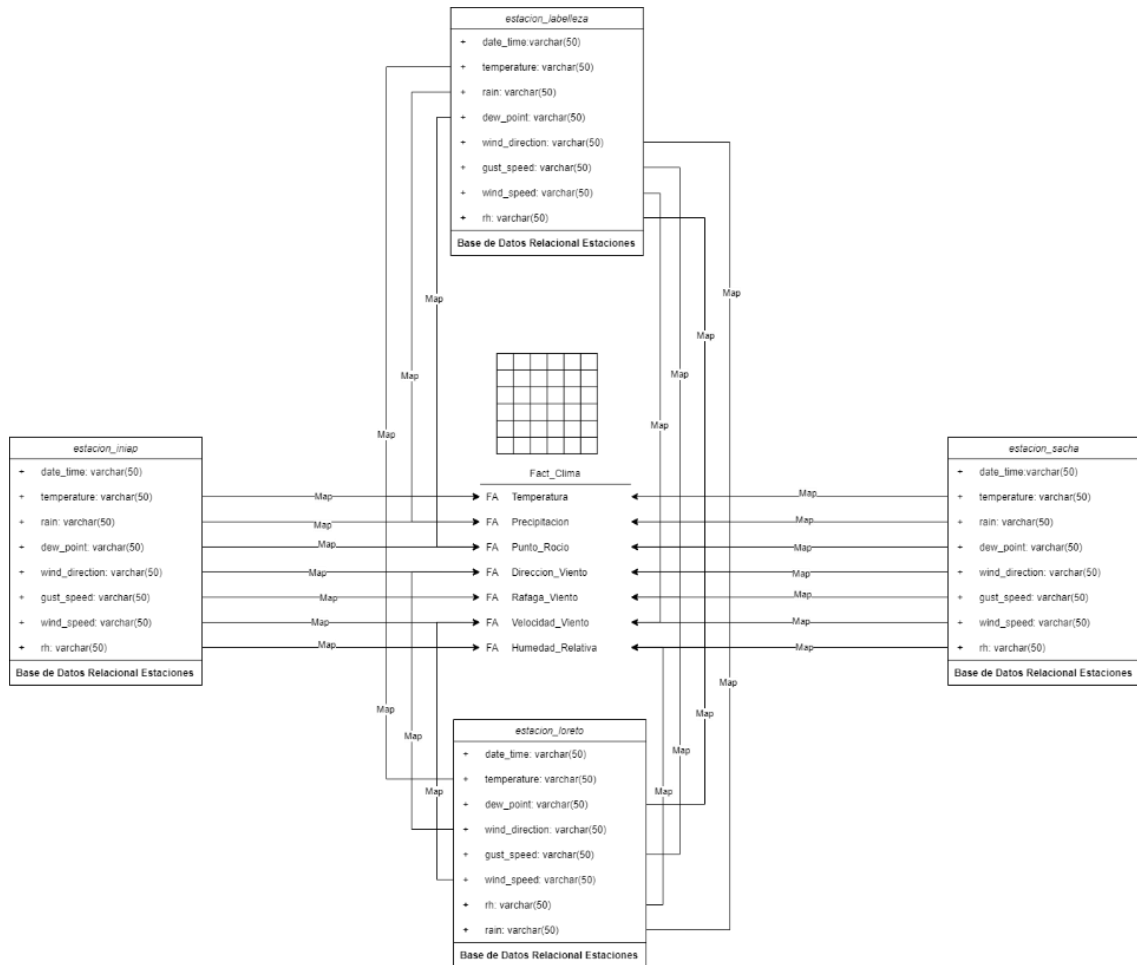
**Figura 34:** Mapeo dimensión estación – Nivel 3

## Fact\_Severidad



**Figura 35:** Mapeo hecho severidad – Nivel 3

## Fact\_Clima

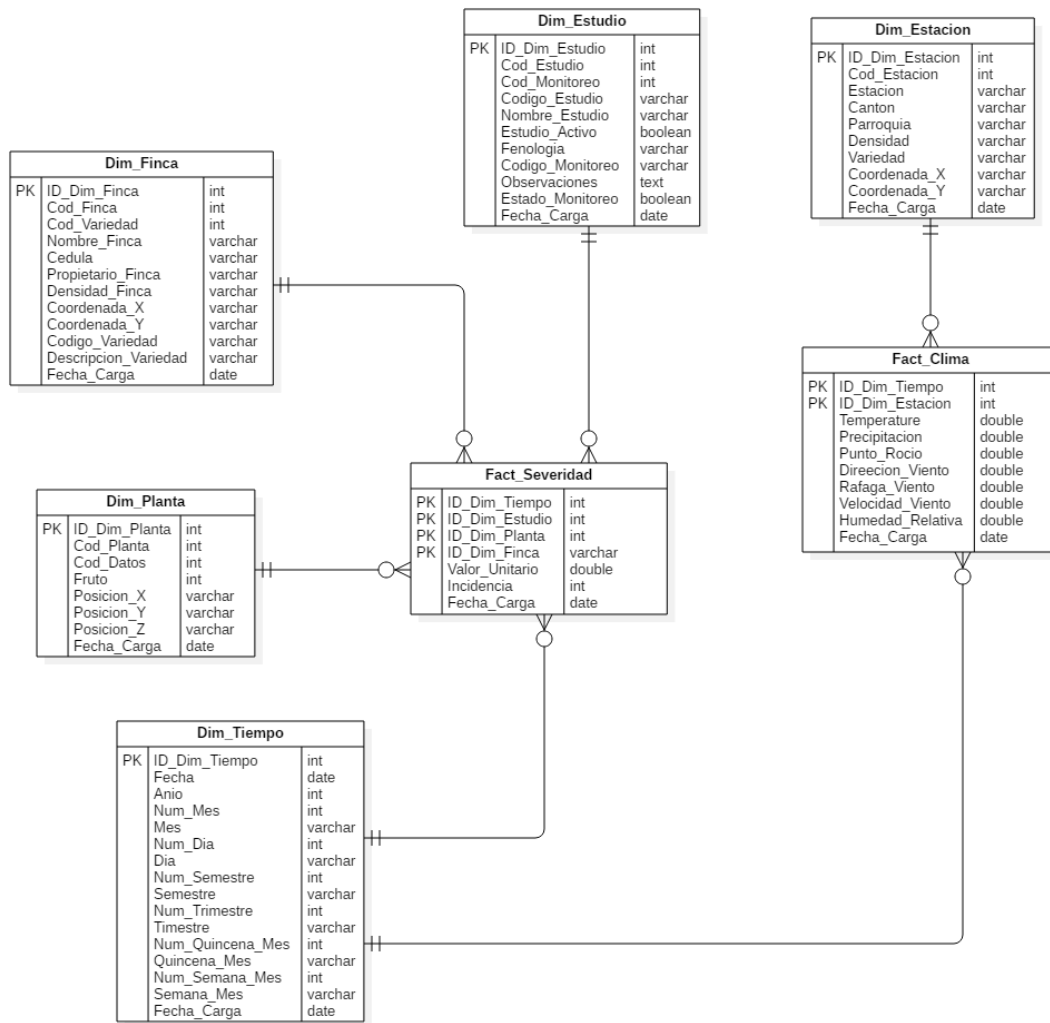


**Figura 36:** Mapeo hecho clima – Nivel 3

### 3.4 Implementación

#### Esquema lógico del Data Warehouse

Este esquema pretende representar las tablas de hechos y dimensiones, que conforman los modelos estrella del esquema conceptual.



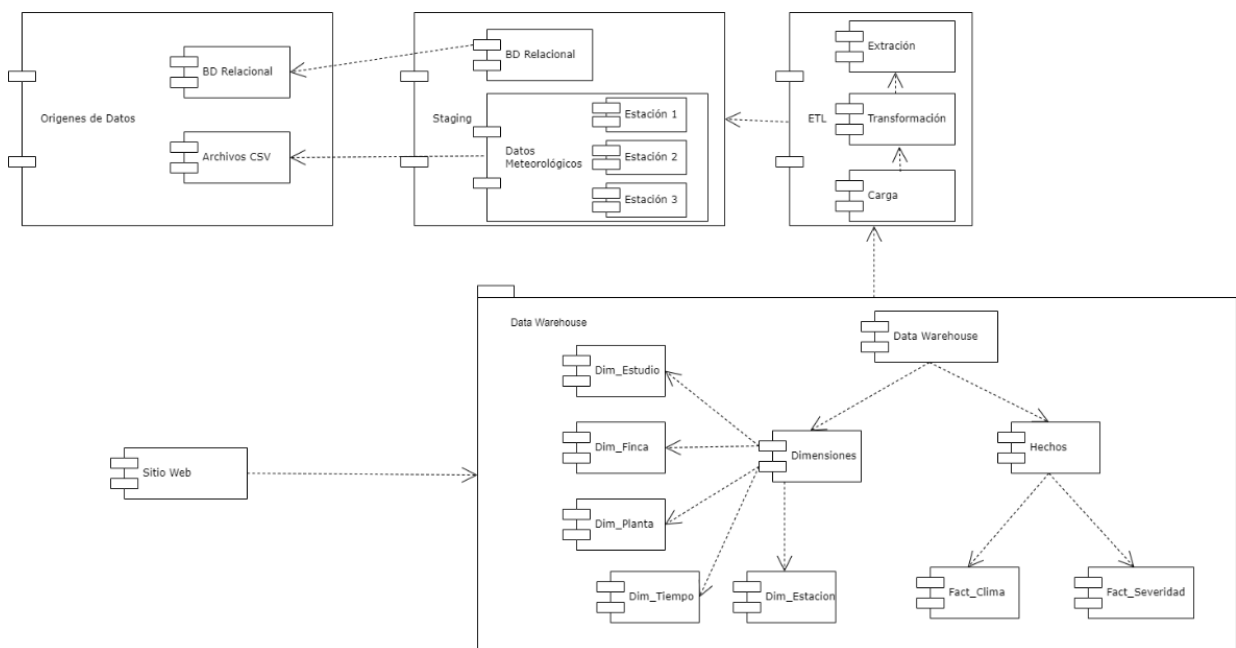
**Figura 37:** Esquema lógico Data Warehouse

## Esquema Físico del Data Warehouse

### Diagrama de componentes

El esquema que se detalla a continuación tiene el objetivo de representar los componentes del Data Warehouse. El componente del Data Warehouse está en

dependencia de los componentes **dimensiones** y **hechos**, mientras que estos últimos dependen de las tablas que están en su entorno.



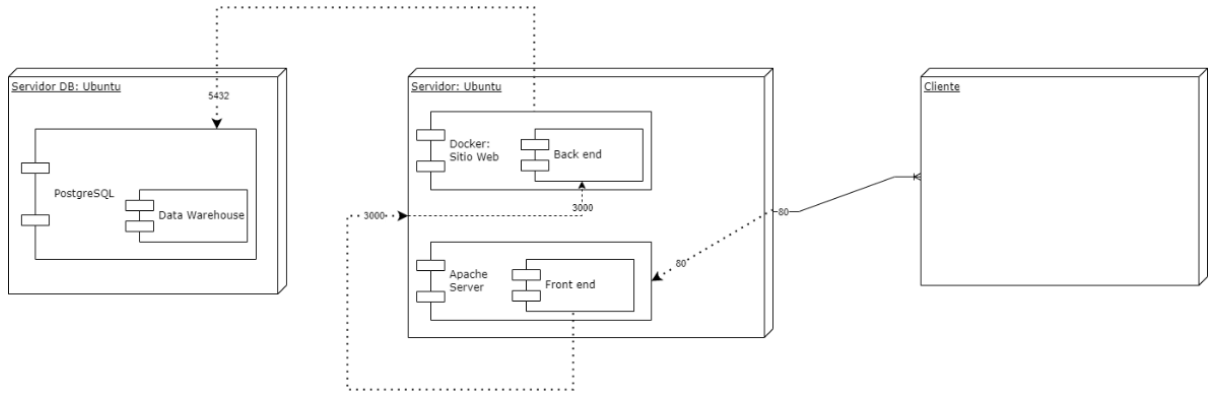
**Figura 38:** Esquema físico - componentes del Data Warehouse

### Diagrama de despliegue

El esquema que se detalla a continuación tiene el objetivo de representar la forma en la que está desplegado el proyecto. Para la implementación del proyecto se cuenta con dos servidores Ubuntu; el primero será el encargado de contener las Bases de datos, por lo que se instalará **PostgreSQL**, mientras que el segundo será el servidor de aplicaciones, donde se instalará **Docker**. PostgreSQL tiene el objetivo de almacenar el Data Warehouse mediante una configuración **ROLAP** y Docker tiene el objetivo de desplegar el Back End del sitio web:

- **Back end.** Permite la consulta de los datos al front end mediante una REST API. El Front End de la aplicación será desplegado en el servidor apache instalado en el servidor de aplicaciones:
- **Front end.** Despliega la interfaz gráfica con la que interactuara el usuario.

El componente del back end se comunica con la base de datos instalada directamente en el host y vincula en el servidor de aplicaciones el puerto 3000 del contenedor con el puerto 3000 del host. El front end se comunica con el back end a través del puerto 3000, consumiendo la REST API y haciendo consultas de los datos.

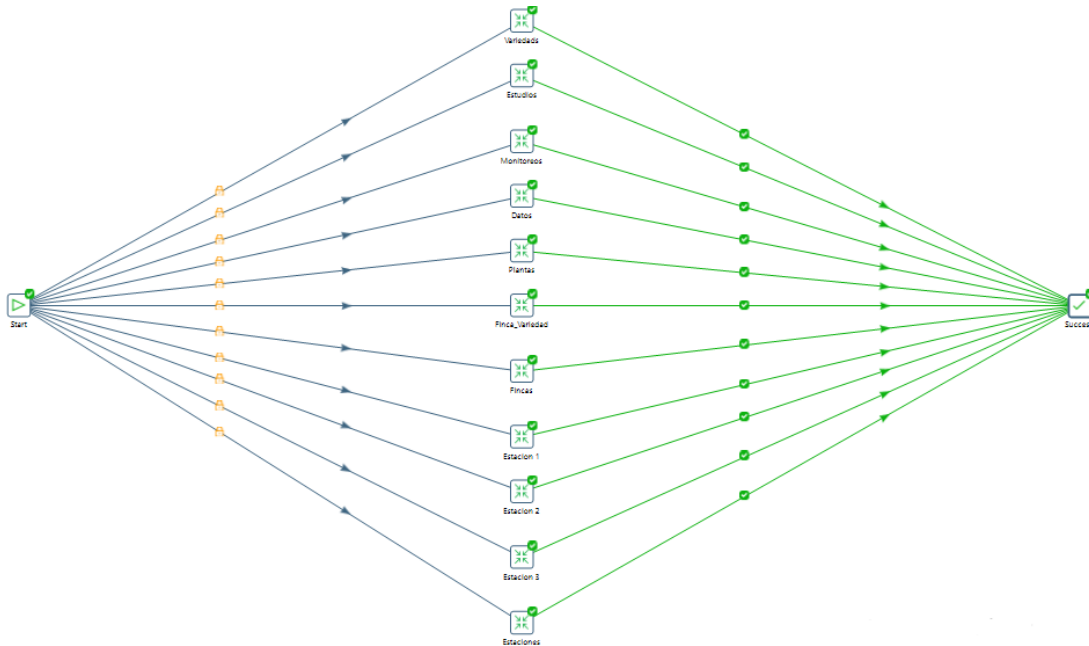


**Figura 39:** Esquema físico - despliegue del Data Warehouse

## Proceso ETL

### Staging

Esta etapa se encuentra entre el origen y el destino de datos, cuyo objetivo es almacenar de manera temporal la información de las distintas fuentes de datos estudios.



**Figura 40:** Staging area.

### Tmp\_Estudios

Almacena datos sobre los estudios realizados en los diferentes cultivos de cacao considerados para el proyecto. Esta información es extraída de la fuente de datos relacional estudios.



**Figura 41:** Tmp\_Estudios – Staging area.

**Tabla 66:** Resumen tmp\_estudios

Área	Descripción
estudios	Se extrae información de esta tabla
Select value	Se selecciona los atributos a considerar (mapping).
tmp_estudios	Se almacena la nueva información en esta tabla.

### Tmp\_Fincas

Almacena información sobre las diferentes fincas que cuentan con cultivos de cacao considerados para el proyecto. Esta información es extraída de la fuente de datos relacional estudios.



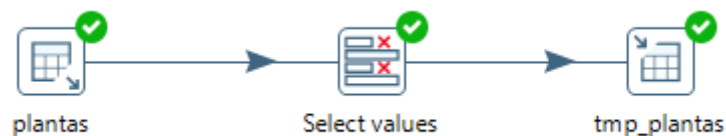
**Figura 42:** Tmp\_Fincas – Staging area.

**Tabla 67:** Resumen tmp\_fincas

Área	Descripción
fincas	Se extrae información de esta tabla
Select value	Se selecciona los atributos a considerar (mapping).
tmp_fincas	Se almacena la nueva información en esta tabla.

### Tmp\_Plantas

Almacena datos descriptivos sobre las diferentes plantas presentes en un cultivo. Esta información es extraída de la fuente de datos relacional estudios.



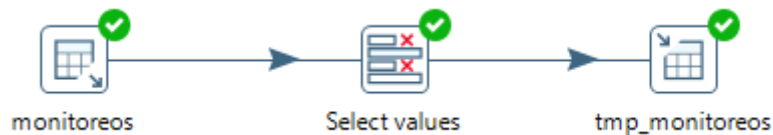
**Figura 43:** Tmp\_Plantas – Staging area.

**Tabla 68:** Resumen tmp\_plantas

Área	Descripción
plantas	Se extrae información de esta tabla
Select value	Se selecciona los atributos a considerar (mapping).
tmp_plantas	Se almacena la nueva información en esta tabla.

### Tmp\_Monitoreos

Almacena información sobre los monitoreos realizados en los diferentes estudios. Esta información es extraída de la fuente de datos relacional estudios.



**Figura 44:** Tmp\_Monitoreos – Staging area.

**Tabla 69:** Resumen tmp\_monitoreos

Área	Descripción
monitoreos	Se extrae información de esta tabla
Select value	Se selecciona los atributos a considerar (mapping).
tmp_monitoreos	Se almacena la nueva información en esta tabla.

### Tmp\_Variedades

Almacena información sobre las diferentes variedades de cultivos. Esta información es extraída de la fuente de datos relacional estudios.



**Figura 45:** Tmp\_Variedades – Staging area.

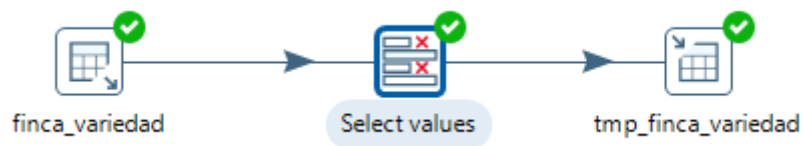
**Tabla 70:** Resumen tmp\_variedades

Área	Descripción
------	-------------

variedades	Se extrae información de esta tabla
Select value	Se selecciona los atributos a considerar (mapping).
tmp_variedades	Se almacena la nueva información en esta tabla.

### Tmp\_Finca\_VariEDAD

Almacena información sobre la relación entre finca y las diferentes variedades de cultivos. Esta información es extraída de la fuente de datos relacional estudios.



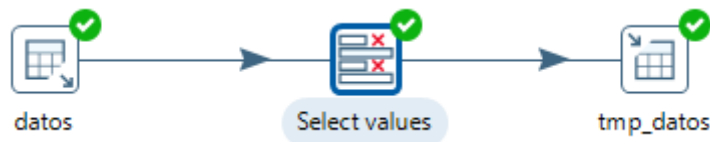
**Figura 46:** Tmp\_Finca\_VariEDAD – Staging area.

**Tabla 71:** Resumen tmp\_finca\_variedad

Área	Descripción
finca_variedad	Se extrae información de esta tabla
Select value	Se selecciona los atributos a considerar (mapping).
tmp_finca_variedad	Se almacena la nueva información en esta tabla.

### Tmp\_Datos

Almacena información sobre los datos realizados en los diferentes monitoreos. Esta información es extraída de la fuente de datos relacional estudios.



**Figura 47:** Tmp\_Datos – Staging area.

**Tabla 72:** Resumen tmp\_datos

Área	Descripción
datos	Se extrae información de esta tabla

Select value	Se selecciona los atributos a considerar (mapping).
tmp_datos	Se almacena la nueva información en esta tabla.

### Tmp\_Estaciones

Almacena información descriptiva de las estaciones. Esta información es extraída de la fuente Base de Datos Relacional Estaciones.



**Figura 48:** Tmp\_Estaciones – Staging area.

**Tabla 73:** Resumen tmp\_estaciones

Área	Descripción
Estaciones	Se extrae información.
Select value	Se selecciona los atributos a considerar (mapping).
tmp_estaciones	Se almacena la nueva información en esta tabla.

### Tmp\_Estacion\_1

Almacena información de datos climatológicos de la estación 1. Esta información es extraída de la fuente de datos Base de Datos Relacional Estaciones.



**Figura 49:** Tmp\_Estacion\_1 – Staging area.

**Tabla 74:** Resumen tmp\_estacion\_1

Área	Descripción
Estación 1	Se extrae información.
Select value	Se selecciona los atributos a considerar (mapping).

tmp_estacion_1	Se almacena la nueva información en esta tabla.
----------------	---

### Tmp\_Estacion\_2

Almacena información de datos climatológicos de la estación 2. Esta información es extraída de la fuente de datos Base de Datos Relacional Estaciones.



**Figura 50:** Tmp\_Estacion\_2 – Staging area.

**Tabla 75:** Resumen tmp\_estacion\_2

Área	Descripción
Estación 2	Se extrae información.
Select value	Se selecciona los atributos a considerar (mapping).
tmp_estacion_2	Se almacena la nueva información en esta tabla.

### Tmp\_Estacion\_3

Almacena información de datos climatológicos de la estación 3. Esta información es extraída de la fuente de datos Base de Datos Relacional Estaciones.



**Figura 51:** Tmp\_Estacion\_3 – Staging area.

**Tabla 76:** Resumen tmp\_estacion\_3

Área	Descripción
Estación 3	Se extrae información.
Select value	Se selecciona los atributos a considerar (mapping).
tmp_estacion_3	Se almacena la nueva información en esta tabla.

### Tmp\_Estacion\_4

Almacena información de datos climatológicos de la estación 4. Esta información es extraída de la fuente de datos Base de Datos Relacional Estaciones.



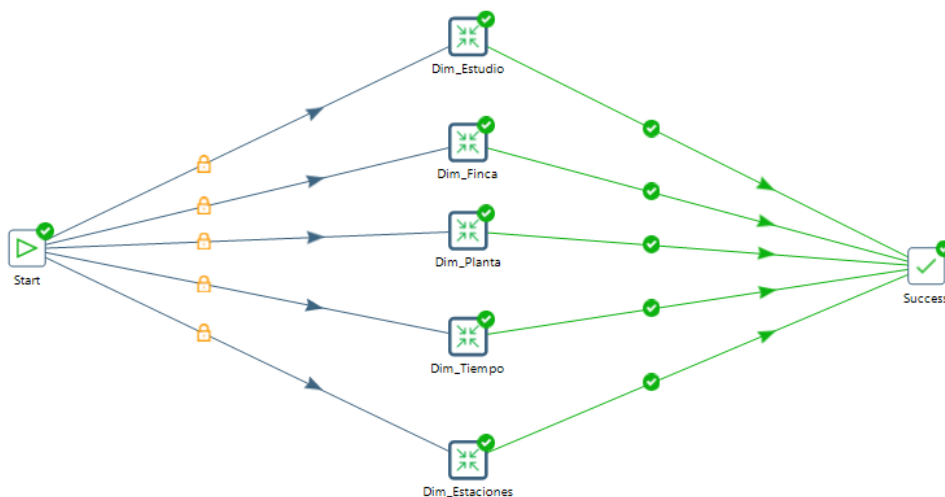
**Figura 52:** Tmp\_Estacion\_4 – Staging area.

**Tabla 77:** Resumen tmp\_estacion\_3

Área	Descripción
Estación 4	Se extrae información.
Select value	Se selecciona los atributos a considerar (mapping).
tmp_estacion_3	Se almacena la nueva información en esta tabla.

### Proceso ETL Dimensiones

Esta etapa se encarga de almacenar información de un área de negocio específica, que mediante sus atributos de dimensión poseen información característica de las tablas de hechos.



**Figura 53:** Dimensiones.

### Dim\_Estudios

Para el proceso ETL de esta dimensión, en primera instancia depende de las siguientes tablas: monitoreos y estudios, estas tablas provienen de la base de datos relacional, que mediante un proceso ETL serán cargadas en la **Dim\_Estudio**.



**Figura 54:** Dim\_Estudios – ETL Dimensiones.

**Tabla 78:** Resumen ETL dim\_estudios

Área	Descripción
Desnormalizado	Se desnormaliza la información desde las siguientes tablas: <b>tmp_estudios</b> y <b>tmp_monitoreos</b> .
Select value	Se selecciona los atributos a considerar (mapping).
Insert / Update	Se actualiza nueva información en la dimensión.

### Dim\_Finca

Para el proceso ETL de esta dimensión, en primera instancia depende de las siguientes tablas: variedades y fincas, estas tablas provienen de la base de datos relacional, que mediante un proceso ETL serán cargadas en la **Dim\_Finca**.



**Figura 55:** Dim\_Finca – ETL Dimensiones.

**Tabla 79:** Resumen ETL dim\_finca

Área	Descripción
Desnormalizado	Se desnormaliza la información desde las siguientes tablas: <b>tmp_fincas</b> y <b>tmp_variedades</b> .
Select value	Se selecciona los atributos a considerar (mapping).

Insert / Update	Se actualiza nueva información en la dimensión.
-----------------	---

### Dim\_Planta

Para el proceso ETL de esta dimensión, en primera instancia depende de las siguientes tablas: datos y plantas, estas tablas provienen de la base de datos relacional, que mediante un proceso ETL serán cargadas en la **Dim\_Finca**.



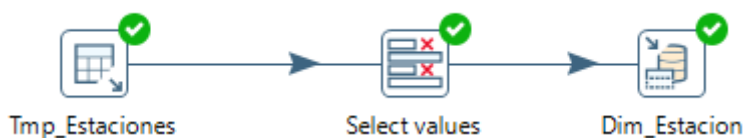
**Figura 56:** Dim\_Planta – ETL Dimensiones.

**Tabla 80:** Resumen ETL dim\_planta

Área	Descripción
Desnormalizado	Se desnormaliza la información desde las siguientes tablas: <b>tmp_plantas</b> y <b>tmp_datos</b> .
Select value	Se selecciona los atributos a considerar (mapping).
Insert / Update	Se actualiza nueva información en la dimensión.

### Dim\_Estacion

Para el proceso ETL de esta dimensión, en primera instancia depende de la información estática, que mediante un proceso ETL serán cargadas en la **Dim\_Estacion**.



**Figura 57:** Dim\_Estacion – ETL Dimensiones.

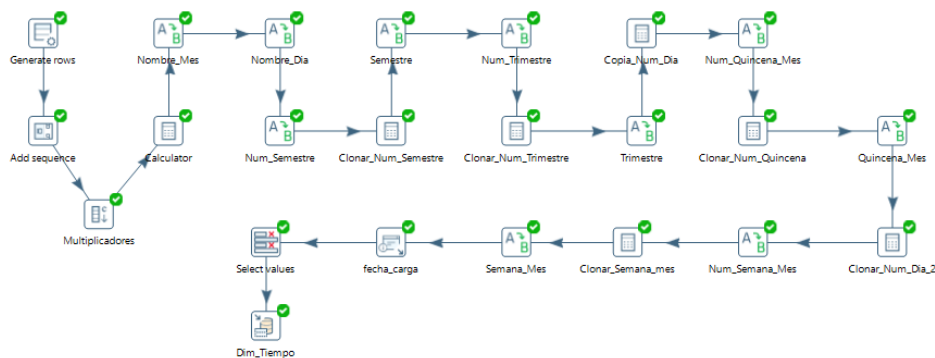
**Tabla 81:** Resumen ETL dim\_estacion

Área	Descripción
Tmp_Estacion	Se obtiene la información desde el staging área, específicamente de Tmp_Estaciones

Select value	Se selecciona los atributos a considerar (mapping).
Dim_Estacion	Se actualiza nueva información en la dimensión.

### Dim\_Tiempo

A diferencia de las demás dimensiones los registros no son traídos de un origen de datos, sino se crean automáticamente a partir de una fecha base. Se generan 1000 fechas más a partir de la fecha base establecida.



**Figura 58:** Dim\_Tiempo – ETL Dimensiones.

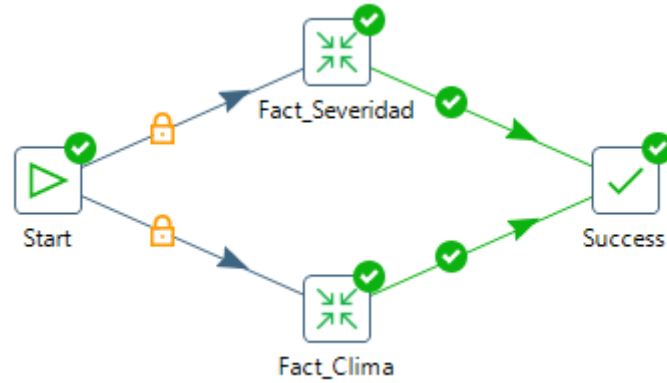
**Tabla 82:** Resumen ETL dim\_tiempo

Área	Descripción
Generate Rows	Genera una cantidad de filas establecidas con una cierta configuración. En este punto se establece la fecha inicial.
Add Sequence	Agrega una columna, generando un nuevo valor para cada registro existente, que sigue una secuencia establecida, en este caso la secuencia va incremento de 1.
Multiplicador	Agrega una o varias columnas de constantes a todas las filas presentes en la tabla.
Calculator	Permite realizar operaciones con cualquiera de las columnas de la tabla, en este caso se

	<p>emplea para obtener valores específicos de la dimensión.</p>
<p>Nombre_Mes</p>	<p>Es un <b>Value Mapper</b>, permite convertir los datos en función de una coincidencia, es decir, busque el dato objetivo y lo transforma a un valor establecido. Esta acción se repite en los siguientes componentes:</p> <p>Nombre_Dia,  Nombre_Semestre, Semestre,  Num_Trimestre, Trimestre,  Num_Quincena_Mes, Quincena_Mes,  Num_Semana_Mes, Semana_Mes.</p>
<p>Clonar_Num_Semestre</p>	<p>Tiene el objetivo de duplicar una columna en específico. Esta acción se repite en los siguientes componentes: Num_Trimestre, Copia_Num_Dia, Clonar_Num_Quincena, Clonar_Num_Dia_2, Clonar_Semana_mes.</p>
<p>fecha_carga</p>	<p>Es un componente denominado <b>Get system info</b>, tiene el objetivo de obtener valores específicos del sistema, en este caso se está utilizando para cargar la fecha actual del sistema.</p>
<p>Select values</p>	<p>Se selecciona los atributos a considerar (mapping).</p>
<p>Dim_Tiempo</p>	<p>Se inserta o actualiza la información en la dimensión.</p>

## Proceso ETL Hechos

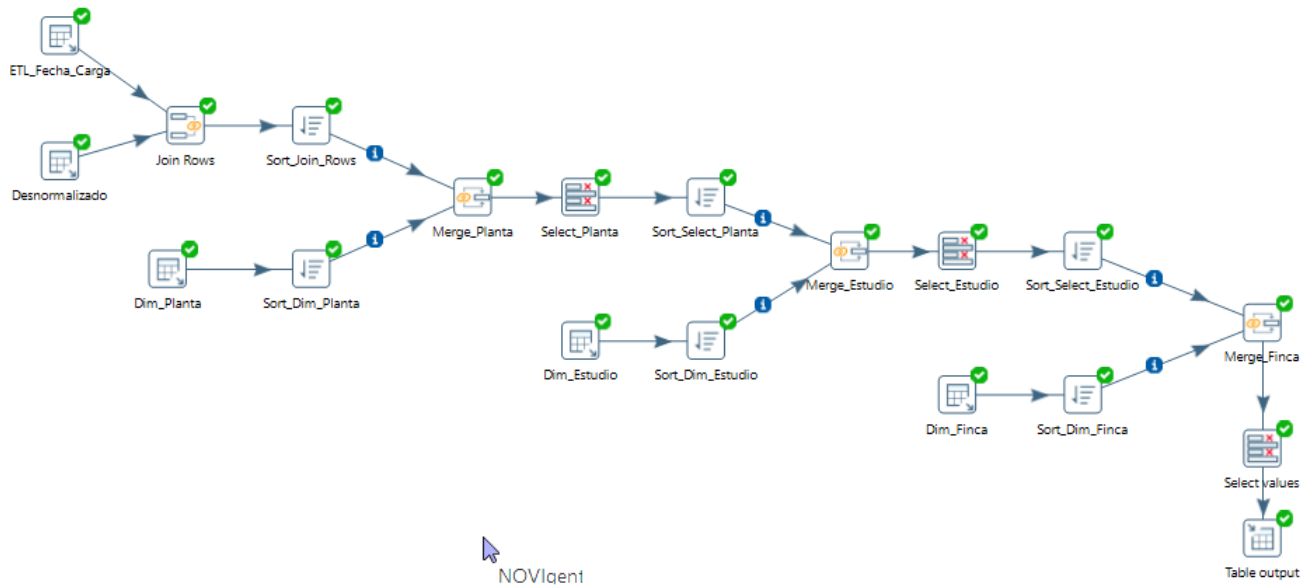
Esta etapa contiene datos de estudio (severidad, incidencia y elementos del clima), métricas que se analizarán en este proyecto.



**Figura 59:** Hechos.

## Fact\_Severidad

Este hecho utiliza las IDs de las siguientes dimensiones: Dim\_estudio, Dim\_finca, Dim\_planta y Dim\_tiempo, para mediante un proceso ETL relacionar los registros de la tabla con los respectivos registros de las dimensiones.



**Figura 60:** Proceso ETL – Fact\_Severidad

**Tabla 83:** Resumen ETL Fact\_Severidad

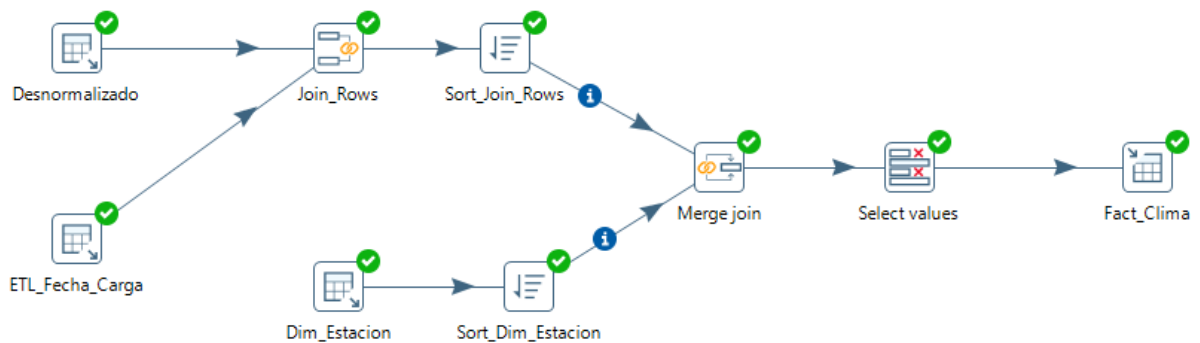
Área	Descripción
------	-------------

ETL_Fecha_Carga	Carga los datos de la tabla ETL_Fecha_Carga, presente en la base de datos.
Desnormalizado	Estructura los registros, cargados desde el Staging, acorde al modelo de la tabla establecido.
Join Rows	Filtra las filas en función del rango de fechas establecidas en la tabla ETL_Fecha_Carga, es decir, solo considera los registros que se encuentren dentro de ese rango, conjuntamente considera el campo <b>Forzar</b> .
Sort_Join_Rows	Ordena de manera descendente o ascendente los registros, en función de una o varias de sus columnas. Esta acción se repite en los siguientes componentes: Sort_Dim_Planta, Sort_Select_Planta, Sort_Dim_Estudio, Sort_Select_Estudio, Sort_Dim_Finca.
Dim_Planta	Carga los datos de la dimensión planta.
Merge_Planta	Este componente tiene el objetivo de realizar un <b>inner join</b> de los datos de la dimensión con los datos del hecho, con el fin de relacionar la ID de la dimensión con el respectivo registro en el hecho. Esta acción se repite en los siguientes componentes: Merge_Estudio, Merge_Finca.

Select_Planta	Se selecciona los atributos a considerar (mapping). Esta acción se repite para los siguientes componentes: Select_Estudio, Select values.
Table output	Se encarga de insertar los registros en la tabla Fact_Severidad de la Base de Datos.

### Fact\_Clima

Este hecho utiliza la ID de la dimensión estación, para mediante un proceso ETL relacionar los registros de la tabla con los respectivos registros de la dimensión.



**Figura 61:** Proceso ETL – Fact\_Severidad

**Tabla 84:** Resumen ETL Fact\_Clima

Área	Descripción
ETL_Fecha_Carga	Carga los datos de la tabla ETL_Fecha_Carga, presente en la base de datos.
Desnormalizado	Estructura los registros, cargados desde el Staging, acorde al modelo de la tabla establecido.
Join Rows	Filtra las filas en función del rango de fechas establecidas en la tabla ETL_Fecha_Carga, es decir, solo considera

	los registros que se encuentren dentro de ese rango, conjuntamente considera el campo <b>Forzar</b> .
Sort_Join_Rows	Ordena de manera descendente o ascendente los registros, en función de una o varias de sus columnas. Esta acción se repite para el componente Dim_Estacion.
Dim_Estacion	Carga los datos de la dimensión estación.
Merge_Join	Este componente tiene el objetivo de realizar un <b>inner join</b> de los datos de la dimensión con los datos del hecho, con el fin de relacionar la ID de la dimensión con el respectivo registro en el hecho.
Select values	Se selecciona los atributos a considerar (mapping).
Fact_Clima	Se encarga de insertar los registros en la tabla Fact_Clima de la Base de Datos.

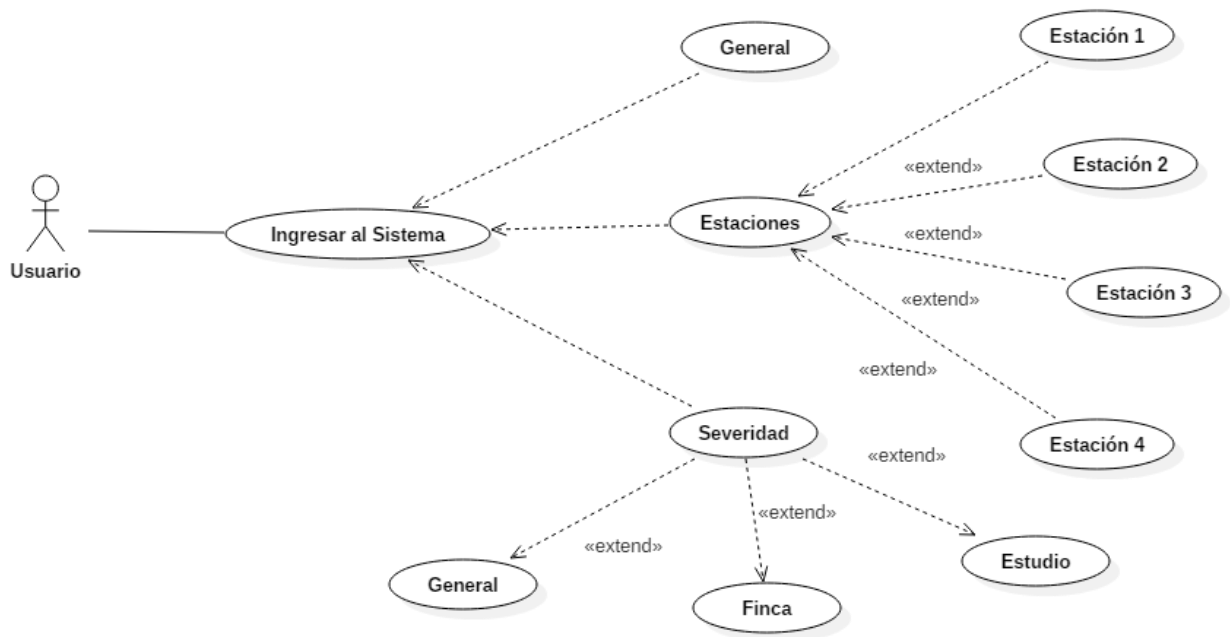
### 3.5 Fase de Construcción

#### Implementación

En el presente flujo de trabajo se estableció como actividad principal para la implementación de los informes, debido a esto dentro de la presente sección se detallará el informe técnico del Sitio Web.

#### Diagrama de Casos de Uso

El diagrama que se detalla a continuación únicamente represente los casos de uso posibles para el Sitio Web, el caso de uso para el Data Warehouse se encuentra desarrollado en una sección anterior.



**Figura 62:** Caso de uso Sitio Web

### Especificación de los Casos de Uso

**Tabla 85:** Especificación Caso de Uso Ingresar al Sistema – Sitio Web

<b>Nombre</b>	Ingresar al Sistema
<b>Actores</b>	Usuario
<b>Propósito</b>	Representa el ingreso general al sistema por parte del usuario.
<b>Resumen</b>	El sistema solo cuenta con el nivel usuario, el cual puede ingresar al sistema para visualizar todos los contenidos y acceder a todas las funciones.
<b>Casos de uso asociados</b>	Este caso uso tiene una relación de dependencia con <b>General, Estaciones y Severidad</b> , ya que estos casos de uso dependen de que el usuario ingrese al sistema.
<b>Flujo de Trabajo</b>	
<b>Actor</b>	<b>Sistema</b>
Ingresar al sistema para visualizar los diferentes	<ol style="list-style-type: none"> <li>1. Ingresar al Navegador.</li> <li>2. Digitar la URL en la barra de búsqueda.</li> </ol>

gráficos y resultados del Data Warehouse.	3. Ingresar al Sitio Web.
<b>Prioridad</b>	Alta.
<b>Cursos alternos</b>	
1. <b>Flujo Alterno 1:</b> La página no se encuentra disponible. Esto sucedería cuando exista algún problema por parte del servidor o cliente al momento de tratar de acceder a la página.	

**Tabla 86:** Especificación Caso de Uso Estaciones – Sitio Web

<b>Nombre</b>	Estaciones	
<b>Actores</b>	Usuario	
<b>Propósito</b>	Representar la capacidad del sistema de permitir el acceso al usuario hacia el módulo estaciones.	
<b>Resumen</b>		
El usuario puede ingresar al módulo estaciones para visualizar los diferentes dashboards disponibles, para todas las estaciones o de manera individual.		
<b>Casos de uso asociados</b>	Los casos de uso: <b>Estacion 1, Estacion 2 y Estacion 3</b> ; son extends del presente caso de uso, debido a que representan la misma funcionalidad con características diferentes.	
<b>Flujo de Trabajo</b>		
<b>Actor</b>	<b>Sistema</b>	
Ingresar al módulo estaciones y visualizar los diferentes gráficos.	<ol style="list-style-type: none"> <li>1. Ingresar al Navegador.</li> <li>2. Digitar la URL en la barra de búsqueda.</li> <li>3. Ingresar al Sitio Web.</li> <li>4. Ingresar al módulo estaciones.</li> </ol>	
<b>Prioridad</b>	Media.	
<b>Cursos alternos</b>		

1. **Flujo Alterno 1:** La página no se encuentra disponible. Esto sucedería cuando exista algún problema por parte del servidor o cliente al momento de tratar de acceder a la página.
2. **Flujo Alterno 2:** Error 404. Esto sucedería cuando la página esté disponible, pero no se encuentre la ruta a la que se intenta acceder, siendo un error dentro del enrutador del Front End.

**Tabla 87:** Especificación Caso de Uso Estación 1 – Sitio Web

<b>Nombre</b>	Estación 1	
<b>Actores</b>	Usuario	
<b>Propósito</b>	Representar las distintas funcionalidades disponibles para el módulo estaciones.	
<b>Resumen</b>		
El usuario puede visualizar los gráficos disponibles para la estación 1.		
<b>Casos de uso asociados</b>	Es un extend del caso de uso <b>Estaciones</b> .	
<b>Flujo de Trabajo</b>		
<b>Actor</b>	<b>Sistema</b>	
Ingresar al módulo estaciones y visualizar los gráficos de la Estación 1.	<ol style="list-style-type: none"> <li>1. Ingresar al Navegador.</li> <li>2. Digitar la URL en la barra de búsqueda.</li> <li>3. Ingresar al Sitio Web.</li> <li>4. Ingresar al módulo estaciones.</li> <li>5. Ingresar a la sección de la Estación 1.</li> <li>6. Visualización de los gráficos.</li> </ol>	
<b>Prioridad</b>	Baja.	
<b>Cursos alternos</b>		
<ol style="list-style-type: none"> <li>1. <b>Flujo Alterno 1:</b> La página no se encuentra disponible. Esto sucedería cuando exista algún problema por parte del servidor o cliente al momento de tratar de acceder a la página.</li> <li>2. <b>Flujo Alterno 2:</b> Error 404. Esto sucedería cuando la página esté disponible, pero no se encuentre la ruta a la que se intenta acceder, siendo un error dentro del enrutador del Front End.</li> </ol>		

3. **Flujo Alterno 3:** Error al intentar cargar los datos para la creación del gráfico. Esto sucedería cuando la REST API del Back End no responda la solicitud hecha por el Front End, dando como resultado que no se visualice el gráfico.

**Tabla 88:** Especificación Caso de Uso Modulo General – Sitio Web

<b>Nombre</b>	General	
<b>Actores</b>	Usuario	
<b>Propósito</b>	Representar la actividad principal que se puede realizar en el caso de uso.	
<b>Resumen</b>		
El sistema solo cuenta con el nivel usuario, el cual puede ingresar al sistema para visualizar todos los contenidos y acceder a todas las funciones.		
<b>Casos de uso asociados</b>	Este caso de uso tiene relación de <b>dependencia</b> del caso de uso: <ul style="list-style-type: none"> <li>• <b>Ingresar al sistema:</b> este caso de uso depende de que el usuario ingrese al sistema.</li> </ul>	
<b>Flujo de Trabajo</b>		
<b>Actor</b>	<b>Sistema</b>	
Ingresar al módulo general y visualizar los gráficos.	<ol style="list-style-type: none"> <li>1. Ingresar al Navegador.</li> <li>2. Digitar la URL en la barra de búsqueda.</li> <li>3. Ingresar al Sitio Web.</li> <li>4. Ingresa a la sección Severidad.</li> <li>5. Establece las configuraciones del gráfico.</li> <li>6. Generación del Gráfico.</li> <li>7. Visualización los reportes.</li> </ol>	
<b>Prioridad</b>	Alta.	
<b>Cursos alternos</b>		

1. <b>Flujo Alternativo 1:</b> La página no se encuentra disponible. Esto sucedería cuando exista algún problema por parte del servidor o cliente al momento de tratar de acceder a la página.
2. <b>Flujo Alternativo 2:</b> Error al intentar cargar los datos para la creación del gráfico. Esto sucedería cuando la REST API del Back End no responda la solicitud hecha por el Front End, dando como resultado que no se visualice el gráfico.

**Tabla 89:** Especificación Caso de Uso Severidad – Sitio Web

<b>Nombre</b>	Severidad	
<b>Actores</b>	Usuario	
<b>Propósito</b>	Representar la actividad principal que se puede realizar en el caso de uso.	
<b>Resumen</b>		
El sistema solo cuenta con el nivel usuario, el cual puede ingresar al sistema para visualizar todos los contenidos y acceder a todas las funciones.		
<b>Casos de uso asociados</b>	Este caso de uso tiene relación de <b>extend</b> con el siguiente caso de uso: <ul style="list-style-type: none"> <li>• <b>Tiempo:</b> Este caso de uso es importante debido a que, si se selecciona un tiempo específico, esta muestra la información de la variable de severidad en función del periodo seleccionado.</li> </ul>	
<b>Flujo de Trabajo</b>		
<b>Actor</b>	<b>Sistema</b>	
Ingresar al módulo de severidad y visualizar los gráficos.	<ol style="list-style-type: none"> <li>1. Ingresar al Navegador.</li> <li>2. Digitar la URL en la barra de búsqueda.</li> <li>3. Ingresar al Sitio Web.</li> <li>4. Ingresa a la sección Severidad.</li> <li>5. Establece las configuraciones del gráfico.</li> </ol>	

	6. Generación del Gráfico. 7. Visualización los reportes.
<b>Prioridad</b>	Alta.
<b>Cursos alternos</b>	
1. <b>Flujo Alterno 1:</b> La página no se encuentra disponible. Esto sucedería cuando exista algún problema por parte del servidor o cliente al momento de tratar de acceder a la página.	
2. <b>Flujo Alterno 2:</b> Error al intentar cargar los datos para la creación del gráfico. Esto sucedería cuando la REST API del Back End no responda la solicitud hecha por el Front End, dando como resultado que no se visualice el gráfico.	

**Tabla 90:** Especificación Caso de Uso General – Sitio Web

<b>Nombre</b>	General
<b>Actores</b>	Usuario
<b>Propósito</b>	Representar las distintas funcionalidades disponibles para el componente general del módulo Severidad.
<b>Resumen</b>	
El usuario puede visualizar los gráficos disponibles para la severidad en general.	
<b>Casos de uso asociados</b>	Es un extend del caso de uso <b>Severidad</b> .
<b>Flujo de Trabajo</b>	
<b>Actor</b>	<b>Sistema</b>
Ingresar al módulo severidad y visualizar los gráficos de la severidad en general.	<ol style="list-style-type: none"> <li>1. Ingresar al Navegador.</li> <li>2. Digitar la URL en la barra de búsqueda.</li> <li>3. Ingresar al Sitio Web.</li> <li>4. Ingresa a la sección Severidad con opción en general.</li> <li>5. Establece las configuraciones del gráfico.</li> <li>6. Generación del Gráfico.</li> </ol>

	7. Visualización los reportes.
<b>Prioridad</b>	Baja.
<b>Cursos alternos</b>	
1. <b>Flujo Alterno 1:</b> La página no se encuentra disponible. Esto sucedería cuando exista algún problema por parte del servidor o cliente al momento de tratar de acceder a la página.	
2. <b>Flujo Alterno 2:</b> Error 404. Esto sucedería cuando la página esté disponible, pero no se encuentre la ruta a la que se intenta acceder, siendo un error dentro del enrutador del Front End.	
3. <b>Flujo Alterno 3:</b> Error al intentar cargar los datos para la creación del gráfico. Esto sucedería cuando la REST API del Back End no responda la solicitud hecha por el Front End, dando como resultado que no se visualice el gráfico.	

**Tabla 91:** Especificación Caso de Uso Finca – Sitio Web

<b>Nombre</b>	Finca
<b>Actores</b>	Usuario
<b>Propósito</b>	Representar las distintas funcionalidades disponibles para el componente finca del módulo severidad.
<b>Resumen</b>	
El usuario puede visualizar los gráficos disponibles para finca.	
<b>Casos de uso asociados</b>	Es un extend del caso de uso <b>Severidad</b> .
<b>Flujo de Trabajo</b>	
<b>Actor</b>	<b>Sistema</b>
Ingresar al módulo severidad y visualizar los gráficos de las fincas.	<ol style="list-style-type: none"> <li>1. Ingresar al Navegador.</li> <li>2. Digitar la URL en la barra de búsqueda.</li> <li>3. Ingresar al Sitio Web.</li> <li>4. Ingresa a la sección Severidad con opción en fincas.</li> <li>5. Establece las configuraciones del gráfico.</li> </ol>

	6. Generación del Gráfico. 7. Visualización los reportes.
<b>Prioridad</b>	Baja.
<b>Cursos alternos</b>	
1. <b>Flujo Alterno 1:</b> La página no se encuentra disponible. Esto sucedería cuando exista algún problema por parte del servidor o cliente al momento de tratar de acceder a la página.	
2. <b>Flujo Alterno 2:</b> Error 404. Esto sucedería cuando la página esté disponible, pero no se encuentre la ruta a la que se intenta acceder, siendo un error dentro del enrutador del Front End.	
3. <b>Flujo Alterno 3:</b> Error al intentar cargar los datos para la creación del gráfico. Esto sucedería cuando la REST API del Back End no responda la solicitud hecha por el Front End, dando como resultado que no se visualice el gráfico.	

**Tabla 92:** Especificación Caso de Uso Estudio – Sitio Web

<b>Nombre</b>	Estudio
<b>Actores</b>	Usuario
<b>Propósito</b>	Representar las distintas funcionalidades disponibles para el componente estudio del módulo severidad.
<b>Resumen</b>	
El usuario puede visualizar los gráficos disponibles para estudio.	
<b>Casos de uso asociados</b>	Es un extend del caso de uso <b>Severidad</b> .
<b>Flujo de Trabajo</b>	
<b>Actor</b>	<b>Sistema</b>
Ingresar al módulo severidad y visualizar los gráficos de los estudios realizados.	<ol style="list-style-type: none"> <li>1. Ingresar al Navegador.</li> <li>2. Digitar la URL en la barra de búsqueda.</li> <li>3. Ingresar al Sitio Web.</li> <li>4. Ingresa a la sección Severidad con opción en estudios.</li> </ol>

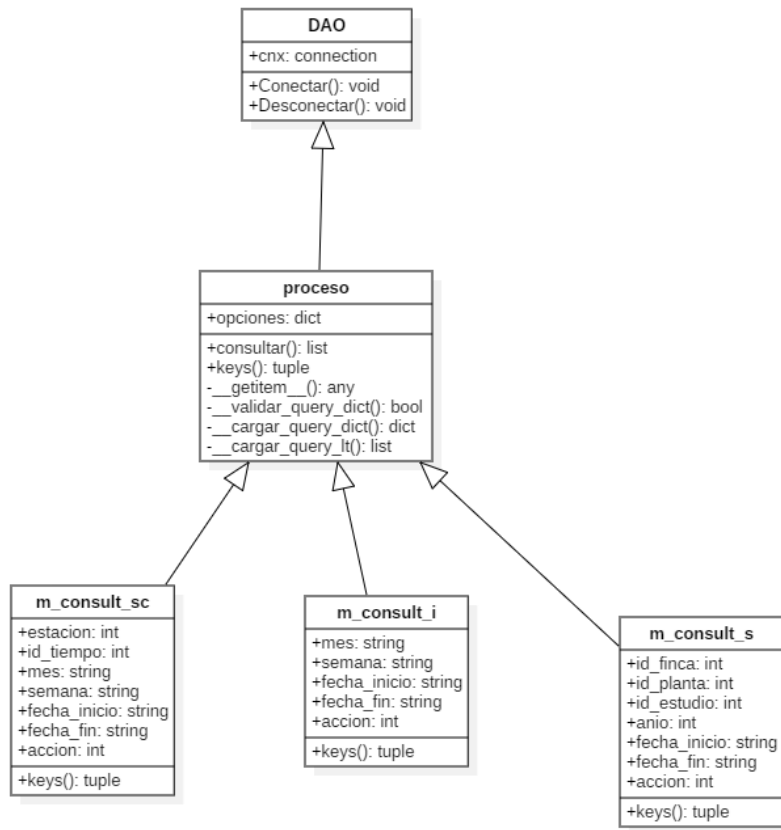
	5. Establece las configuraciones del gráfico. 6. Generación del Gráfico. 7. Visualización los reportes.
<b>Prioridad</b>	Baja.
<b>Cursos alternos</b> 1. <b>Flujo Alternativo 1:</b> La página no se encuentra disponible. Esto sucedería cuando exista algún problema por parte del servidor o cliente al momento de tratar de acceder a la página. 2. <b>Flujo Alternativo 2:</b> Error 404. Esto sucedería cuando la página esté disponible, pero no se encuentre la ruta a la que se intenta acceder, siendo un error dentro del enrutador del Front End. 3. <b>Flujo Alternativo 3:</b> Error al intentar cargar los datos para la creación del gráfico. Esto sucedería cuando la REST API del Back End no responda la solicitud hecha por el Front End, dando como resultado que no se visualice el gráfico.	

## Diagrama de Clases

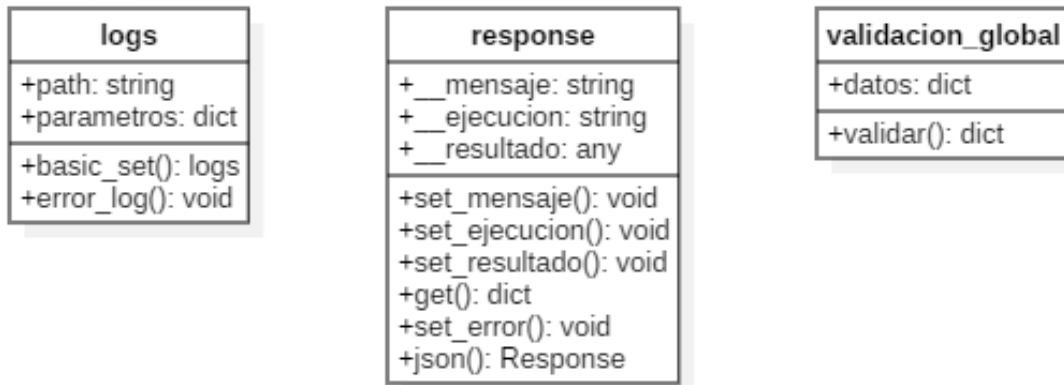
### Front End

El esquema de trabajo del Front End no está orientado a objetos, sino a componentes.

### Back End



**Figura 63:** Conexión a la Base de Datos



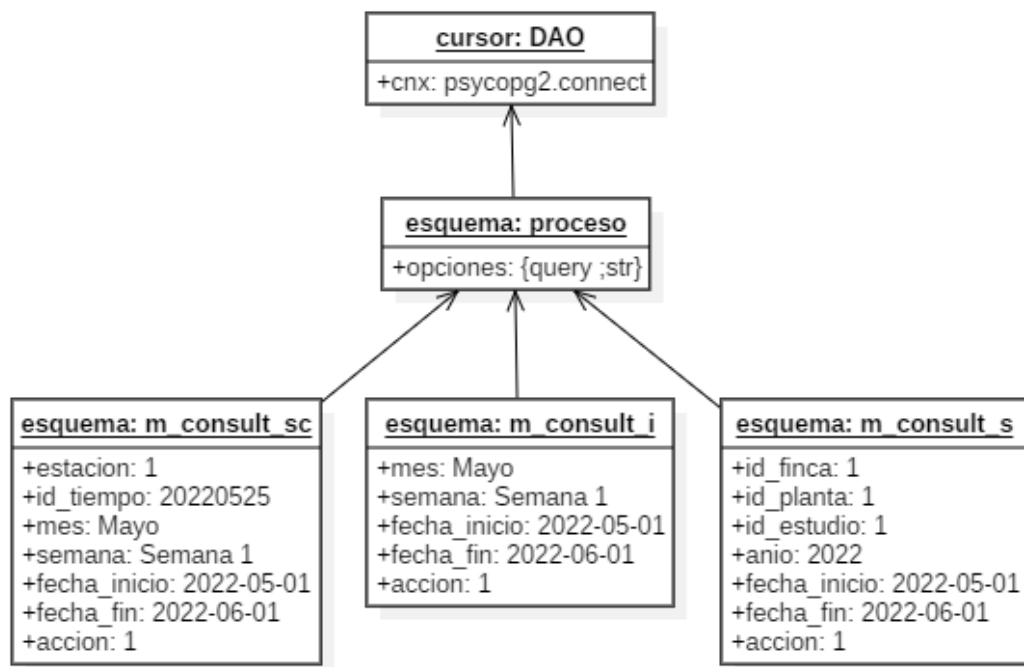
**Figura 64:** Clases Aisladas

## Diagrama de Objetos

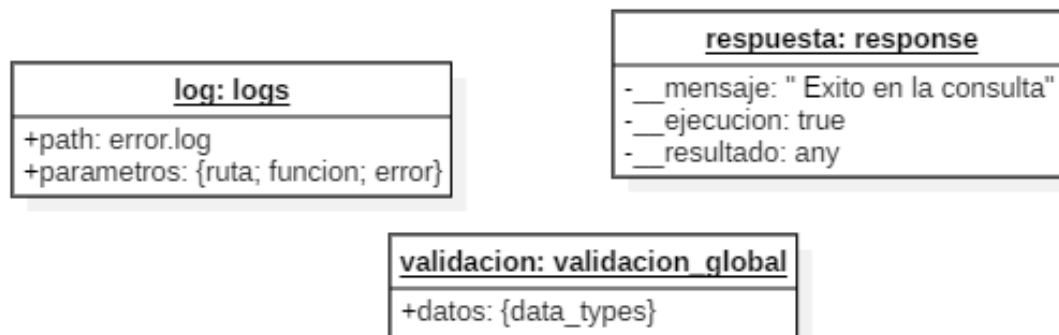
### Front End

El esquema de trabajo del Front End no está orientado a objetos, sino a componentes.

### Back End

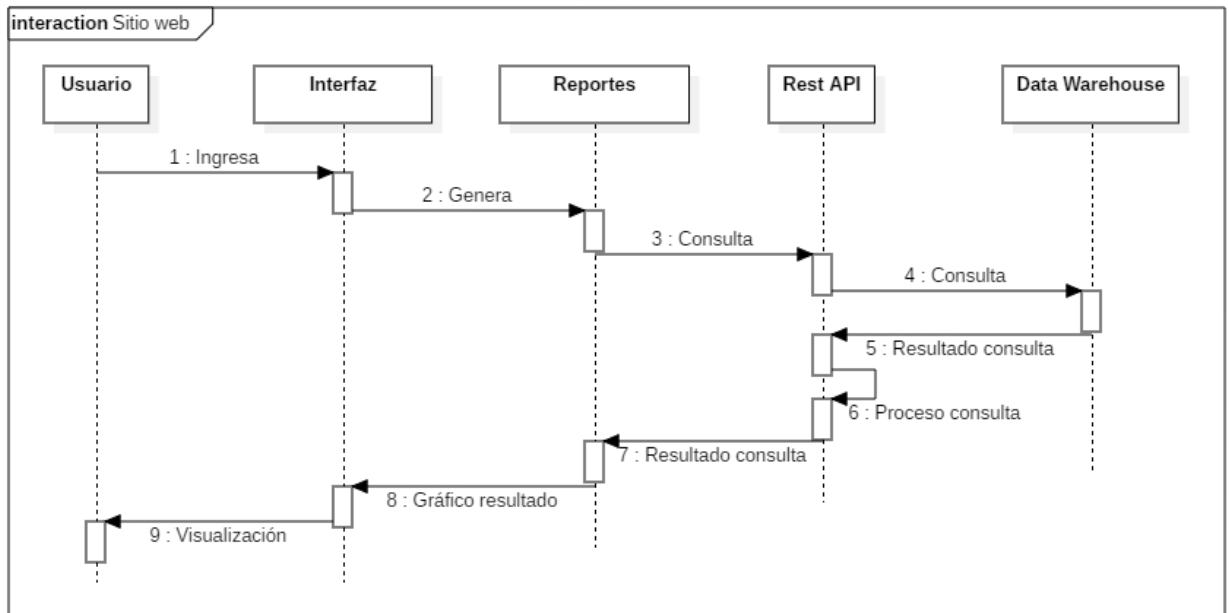


**Figura 65:** Conexión a Base de Datos



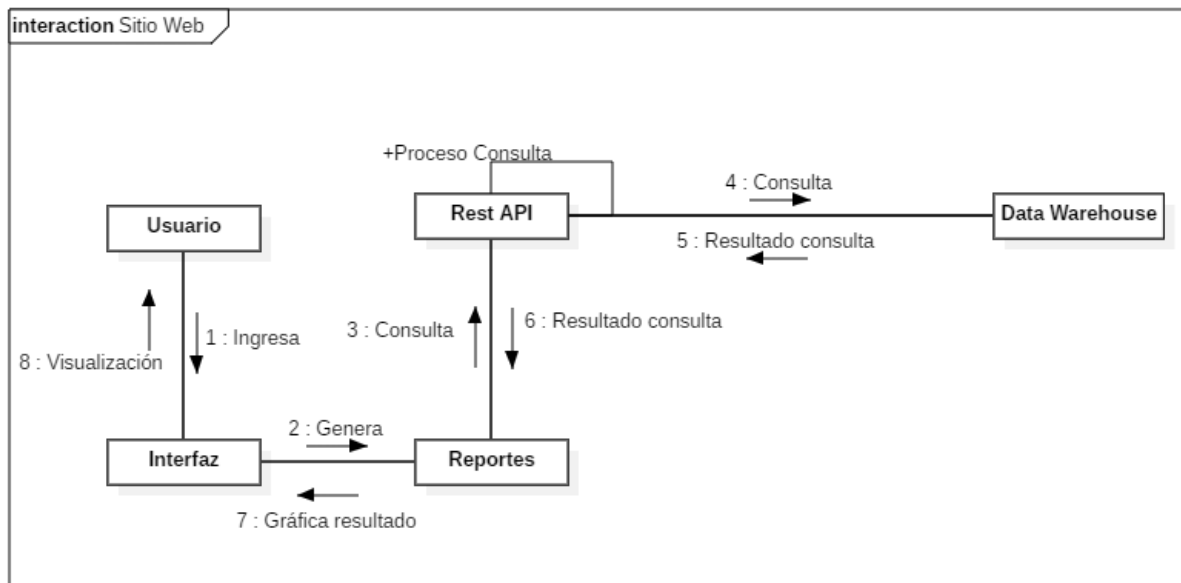
**Figura 66:** Clases Aisladas

## Diagrama de Secuencia



**Figura 67:** Diagrama de Secuencia

## Diagrama de Colaboración



**Figura 68:**Diagrama de Colaboración

## Diagrama de Actividades

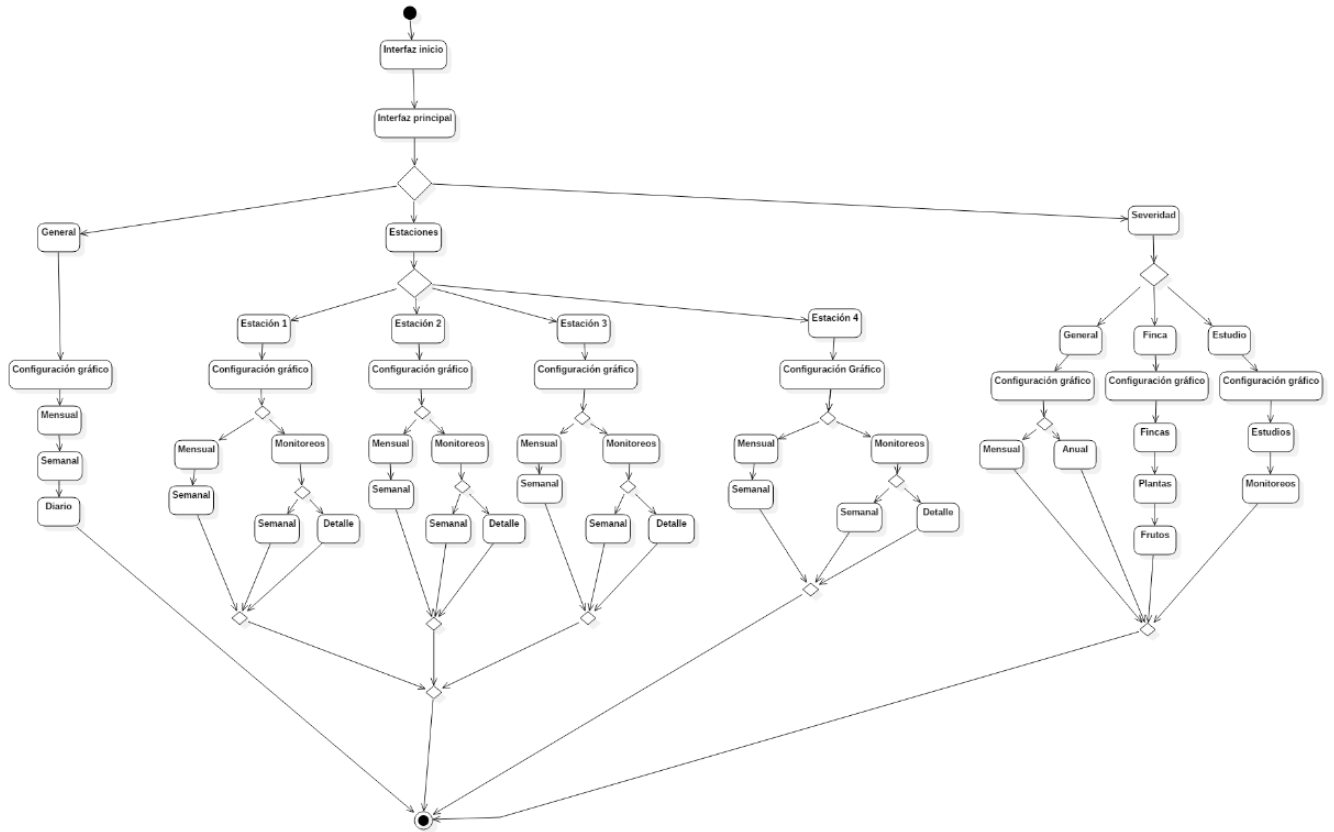
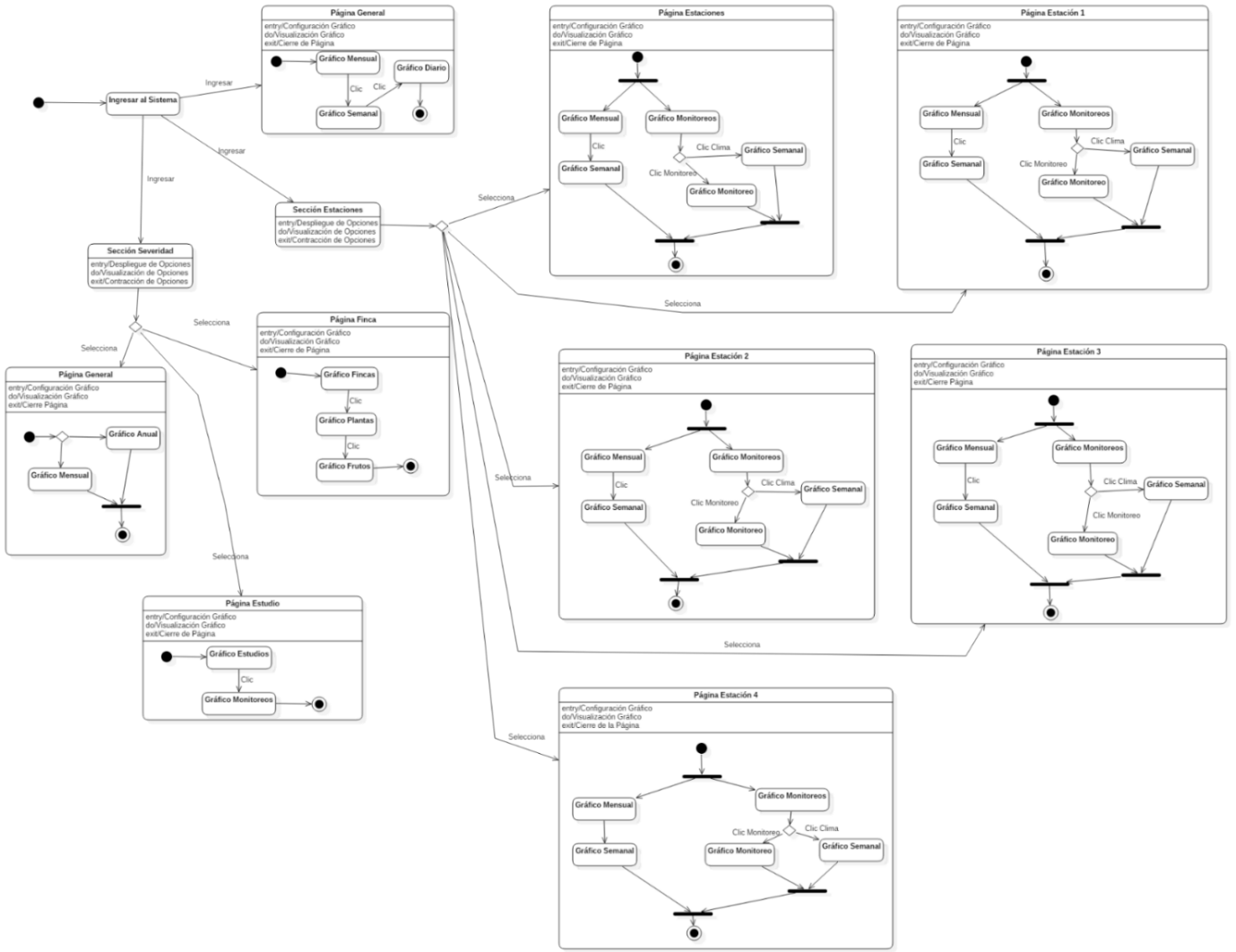


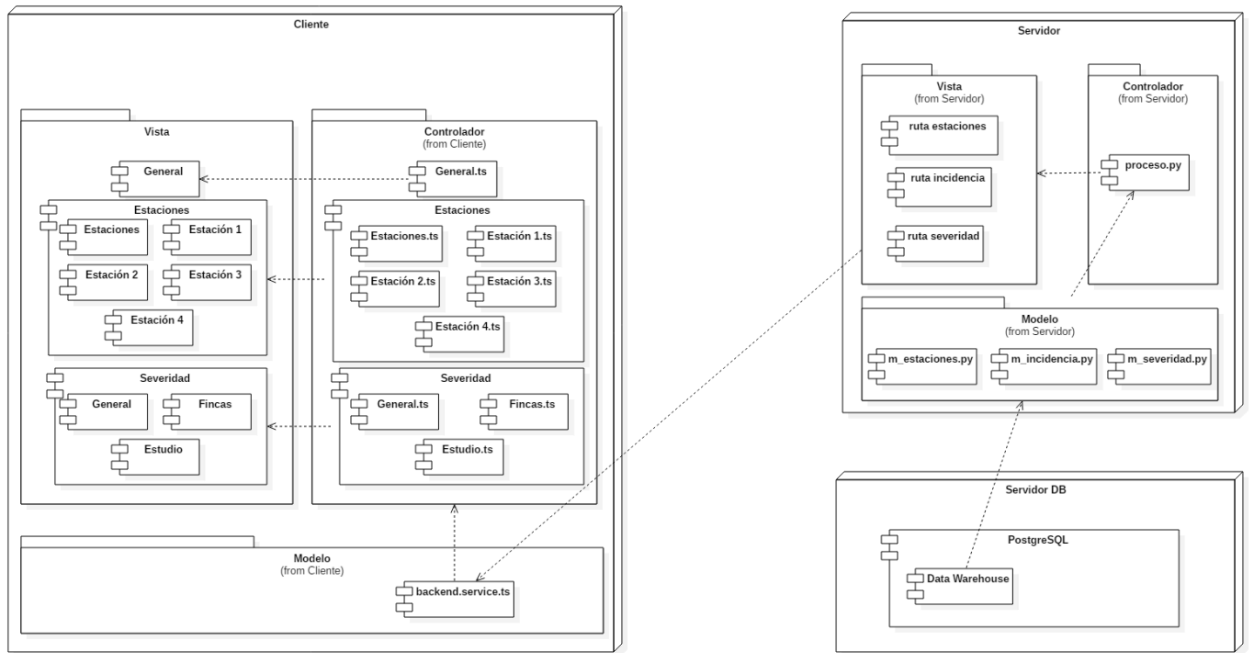
Figura 69: Diagrama de Actividades

# Diagrama de Estados



**Figura 70:** Diagrama de Estados

## Diagrama de Componentes



**Figura 71:** Diagrama de Componentes

## 3.6 Pruebas

### Planeación

A continuación, se detalla la planificación del diseño e implementación de las pruebas.

**Tabla 93:** Planificación de las pruebas.

N°	Actividades	Semana 1					Semana 2				
		1D	2D	3D	4D	5D	1D	2D	3D	4D	5D
1	Planeación.	X	X								
2	Diseño.		X	X	X						
3	Implementación.					X	X	X	X		
4	Evaluación de resultados.								X	X	X

## Diseño e Implementación

### ETL

### Staging

**Tabla 94:** Caso de Prueba Carga Variedades - Staging

Caso de Prueba N° 1

<b>Prioridad</b>	Alta	<b>Estado</b>	1
<b>Objeto a analizar:</b> Transformación “Variedades”			
<b>Objetivo:</b> Verificar el correcto funcionamiento de la transformación “Variedades” en la carga de la tabla “tmp_variedades” de la base de datos Staging.			
<b>Descripción:</b> Se ejecutará la transformación “Variedades” individualmente, para verificar que los datos sean cargados correctamente en la tabla “tmp_variedades”.			
<b>Pre condiciones</b>	1. Base de datos Staging en funcionamiento.		
<b>Entradas</b>	Tabla “variedades” de la Base de datos relacional “monitoreo”.		
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Ingresar a la transformación en la herramienta Spoon de Pentaho Data Integration.</li> <li>2. Ejecutar la transformación.</li> <li>3. Verificar los datos en la tabla de la base de datos Staging.</li> </ol>		
<b>Resultados Esperados</b>	Los datos de la tabla “variedades” de la base de datos relacional “monitoreo” se transfirieron correctamente a la tabla “tmp_variedades” de la base de datos “bd_staging”.		
<b>Post condiciones</b>	Ninguna		
<b>Observaciones:</b> Ninguna			

**Tabla 95:** Caso de Prueba Carga Estudios - Staging

<b>Caso de Prueba N° 2</b>			
<b>Prioridad</b>	Alta	<b>Estado</b>	1
<b>Objeto a analizar:</b> Transformación “Estudios”			

<b>Objetivo:</b> Verificar el correcto funcionamiento de la transformación “estudios” en la carga de la tabla “tmp_estudios” de la base de datos Staging.	
<b>Descripción:</b> Se ejecutará la transformación “Estudios” individualmente, para verificar que los datos sean cargados correctamente en la tabla “tmp_estudios”.	
<b>Pre condiciones</b>	1. Base de datos Staging en funcionamiento.
<b>Entradas</b>	Tabla “estudios” de la Base de datos relacional “monitoreo”.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Ingresar a la transformación en la herramienta Spoon de Pentaho Data Integration.</li> <li>2. Ejecutar la transformación.</li> <li>3. Verificar los datos en la tabla de la base de datos Staging.</li> </ol>
<b>Resultados Esperados</b>	Los datos de la tabla “ <b>estudios</b> ” de la base de datos relacional “ <b>monitoreo</b> ” se transfirieron correctamente a la tabla “ <b>tmp_estudios</b> ” de la base de datos “ <b>bd_staging</b> ”.
<b>Post condiciones</b>	Ninguna
<b>Observaciones:</b> Ninguna	

**Tabla 96:** Caso de Prueba Carga Estacion 1 - Staging

<b>Caso de Prueba N° 3</b>			
<b>Prioridad</b>	Alta	<b>Estado</b>	1
<b>Objeto a analizar:</b> Transformación “Estacion 1”			
<b>Objetivo:</b>			

Verificar el correcto funcionamiento de la transformación “Estacion 1” en la carga de la tabla “tmp_estacion_1” de la base de datos Staging.	
<b>Descripción:</b> Se ejecutará la transformación “Estudios” individualmente, para verificar que los datos sean cargados correctamente en la tabla “tmp_estacion_1”.	
<b>Pre condiciones</b>	<ol style="list-style-type: none"> <li>1. Base de Datos db_patfa_files en funcionamiento.</li> <li>2. Base de datos Staging en funcionamiento.</li> </ol>
<b>Entradas</b>	Tabla “estacion_iniap” de la Base de Datos Relacional ”db_patfa_files”.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Ingresar a la transformación en la herramienta Spoon de Pentaho Data Integration.</li> <li>2. Ejecutar la transformación.</li> <li>3. Verificar los datos en la tabla de la base de datos Staging.</li> </ol>
<b>Resultados Esperados</b>	Los datos de la tabla “estacion_iniap” se transfirieron correctamente a la tabla “tmp_estacion_1” de la base de datos “bd_staging”.
<b>Post condiciones</b>	Ninguna
<b>Observaciones:</b> Ninguna	

## Dimensiones

**Tabla 97:** Caso de Prueba Carga Dim\_Estudio - Data Warehouse

<b>Caso de Prueba N° 4</b>			
<b>Prioridad</b>	Alta	<b>Estado</b>	1
<b>Objeto a analizar:</b>			

Transformación “Dim_Estudio”	
<b>Objetivo:</b> Verificar el correcto funcionamiento de la transformación “Dim_Estudio” en la carga de la tabla “Dim_Estudio” de la base de datos del Data Warehouse.	
<b>Descripción:</b> Se ejecutará la transformación “Dim_Estudio” individualmente, para verificar que los datos sean limpiados y cargados correctamente en la tabla “Dim_Estudio”.	
<b>Pre condiciones</b>	<ol style="list-style-type: none"> <li>1. Base de datos “bd_staging” en funcionamiento.</li> <li>2. Base de datos “dw_patfa” en funcionamiento.</li> </ol>
<b>Entradas</b>	Tablas “tmp_estudio y tmp_monitoreos” de la Base de datos “bd_staging”.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Ingresar a la transformación en la herramienta Spoon de Pentaho Data Integration.</li> <li>2. Ejecutar la transformación.</li> <li>3. Verificar los datos en la tabla de la base de datos del Data Warehouse.</li> </ol>
<b>Resultados Esperados</b>	Los datos presentes en la dimensión <b>estudio</b> del Data Warehouse, deben estar cargados correctamente.
<b>Post condiciones</b>	Ninguna
<b>Observaciones:</b> Ninguna	

**Tabla 98:** Caso de Prueba Carga Dim\_Finca - Data Warehouse

<b>Caso de Prueba N° 5</b>			
<b>Prioridad</b>	Alta	<b>Estado</b>	1
<b>Objeto a analizar:</b>			

Transformación “Dim_Finca”	
<b>Objetivo:</b> Verificar el correcto funcionamiento de la transformación “Dim_Finca” en la carga de la tabla “Dim_Finca” de la base de datos del Data Warehouse.	
<b>Descripción:</b> Se ejecutará la transformación “Dim_Finca” individualmente, para verificar que los datos sean limpiados y cargados correctamente en la tabla “Dim_Finca”.	
<b>Pre condiciones</b>	<ol style="list-style-type: none"> <li>1. Base de datos “bd_staging” en funcionamiento.</li> <li>2. Base de datos “dw_patfa” en funcionamiento.</li> </ol>
<b>Entradas</b>	Tablas “tmp_finca, tmp_variedades y tmp_finca_variedad” de la Base de datos “bd_staging”.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Ingresar a la transformación en la herramienta Spoon de Pentaho Data Integration.</li> <li>2. Ejecutar la transformación.</li> <li>3. Verificar los datos en la tabla de la base de datos del Data Warehouse.</li> </ol>
<b>Resultados Esperados</b>	Los datos presentes en la dimensión <b>finca</b> del Data Warehouse, deben estar cargados correctamente.
<b>Post condiciones</b>	Ninguna
<b>Observaciones:</b> Ninguna	

## Hechos

**Tabla 99:** Caso de Prueba Carga Fact\_Severidad – Data Warehouse

<b>Caso de Prueba N° 6</b>			
<b>Prioridad</b>	Alta	<b>Estado</b>	1

<b>Objeto a analizar:</b> Transformación “Fact_Severidad”	
<b>Objetivo:</b> Verificar el correcto funcionamiento de la transformación “Fact_Severidad” en la carga de la tabla “Fact_Severidad” de la base de datos del Data Warehouse.	
<b>Descripción:</b> Se ejecutará la transformación “Fact_Severidad” individualmente, para verificar que los datos sean limpiados y cargados correctamente en la tabla “Fact_Severidad”.	
<b>Pre condiciones</b>	<ol style="list-style-type: none"> <li>1. Base de datos “bd_staging” en funcionamiento.</li> <li>2. Base de datos “dw_patfa” en funcionamiento.</li> </ol>
<b>Entradas</b>	Tablas “tmp_datos, tmp_plantas, tmp_monitoreos, tmp_estudios y tmp_finca_variedad” de la Base de datos “bd_staging”.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Ingresar a la transformación en la herramienta Spoon de Pentaho Data Integration.</li> <li>2. Ejecutar la transformación.</li> <li>3. Verificar los datos en la tabla de la base de datos del Data Warehouse.</li> </ol>
<b>Resultados Esperados</b>	Los datos presentes en el hecho de <b>severidad</b> del Data Warehouse, deben estar cargados correctamente.
<b>Post condiciones</b>	Ninguna
<b>Observaciones:</b> Ninguna	

**Tabla 100:** Caso de Prueba Carga Fact\_Clima – Data Warehouse

<b>Caso de Prueba N° 7</b>			
<b>Prioridad</b>	Alta	<b>Estado</b>	1
<b>Objeto a analizar:</b> Transformación “Fact_Clima”			
<b>Objetivo:</b> Verificar el correcto funcionamiento de la transformación “Fact_Clima” en la carga de la tabla “Fact_Clima” de la base de datos del Data Warehouse.			
<b>Descripción:</b> Se ejecutará la transformación “Fact_Clima” individualmente, para verificar que los datos sean limpiados y cargados correctamente en la tabla “Fact_Clima”.			
<b>Pre condiciones</b>	<ol style="list-style-type: none"> <li>1. Base de datos “bd_staging” en funcionamiento.</li> <li>2. Base de datos “dw_patfa” en funcionamiento.</li> </ol>		
<b>Entradas</b>	Tablas “tmp_estacion_1, tmp_estacion_2, tmp_estacion_3 y tmp_estaciones” de la Base de datos “bd_staging”.		
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Ingresar a la transformación en la herramienta Spoon de Pentaho Data Integration.</li> <li>2. Ejecutar la transformación.</li> <li>3. Verificar los datos en la tabla de la base de datos del Data Warehouse.</li> </ol>		
<b>Resultados Esperados</b>	Los datos presentes en el hecho de <b>clima</b> del Data Warehouse, deben estar cargados correctamente.		
<b>Post condiciones</b>	Ninguna		
<b>Observaciones:</b> Ninguna			

## Script

**Tabla 101:** Caso de Prueba Sp\_Proceso\_ETL\_1 – Data Warehouse

<b>Caso de Prueba N° 8</b>			
<b>Prioridad</b>	Alta	<b>Estado</b>	1
<b>Objeto a analizar:</b> Procedimiento almacenado “Sp_Proceso_ETL”			
<b>Objetivo:</b> Verificar el correcto funcionamiento del procedimiento almacenado “Sp_Proceso_ETL” al insertar registros en la tabla “ <b>etl_fecha_carga</b> ”.			
<b>Descripción:</b> Se ejecutará el procedimiento almacenado de la base de datos, con la configuración necesaria para inserción de registros de tal forma que se logre validar dicho proceso.			
<b>Pre condiciones</b>		<ol style="list-style-type: none"> <li>1. Base de datos “dw_patfa” en funcionamiento.</li> <li>2. Tabla “<b>etl_fecha_carga</b>”.</li> </ol>	
<b>Entradas</b>		<ol style="list-style-type: none"> <li>1. <b>ID:</b> Identificador único del registro que se desea actualizar o eliminar.</li> <li>2. <b>Módulo:</b> Nombre del hecho que será cargado en el Data Warehouse.</li> <li>3. <b>Fecha de Inicio:</b> Fecha inicial del rango que será considerado para la carga del módulo.</li> <li>4. <b>Fecha de Fin:</b> Fecha final del rango que será considerado para la carga del módulo.</li> <li>5. <b>Forzar:</b> Bandera que establece si el módulo está habilitado para la carga.</li> <li>6. <b>Acción:</b> Establece el proceso que llevara a cabo el procedimiento.</li> </ol>	
<b>Pasos</b>		<ol style="list-style-type: none"> <li>1. Ingresar a Padmin.</li> </ol>	

	<ol style="list-style-type: none"> <li>2. Llamar al script con los parámetros correspondientes.</li> <li>3. Ejecutar la consulta.</li> <li>4. Verificar el resultado en la tabla “<b>etl_fecha_carga</b>”.</li> </ol>
<b>Resultados Esperados</b>	Inserción de registro exitosa.
<b>Post condiciones</b>	Ninguna
<b>Observaciones:</b> Ninguna	

## Front End

### General

**Tabla 102:** Caso de Prueba Información\_Graficos\_General

<b>Caso de Prueba N° 9</b>			
<b>Prioridad</b>	Media	<b>Estado</b>	1
<b>Objeto a analizar:</b> Información del módulo de general.			
<b>Objetivo:</b> Verificar que se muestre información de los diferentes gráficos en el módulo de general.			
<b>Descripción:</b> Se verificará que la información este enfocada y detallada a cada gráfico.			
<b>Pre condiciones</b>		Sitio web en funcionamiento. Módulo General el funcionamiento.	
<b>Entradas</b>		Ninguna.	
<b>Pasos</b>		<ol style="list-style-type: none"> <li>1. Ingresar al Navegador.</li> <li>2. Digitar la URL en la barra de búsqueda.</li> <li>3. Ingresar al Sitio Web.</li> <li>4. Ir a la sección de general.</li> <li>5. Visualizar la descripción de cada gráfico.</li> </ol>	

<b>Resultados Esperados</b>	Información detallada de cada gráfico.
<b>Post condiciones</b>	Ninguna.
<b>Observaciones:</b> Ninguna	

## Estaciones

**Tabla 103:** Caso de Prueba Información\_Graficos\_Estaciones

<b>Caso de Prueba N° 10</b>			
<b>Prioridad</b>	Media	<b>Estado</b>	1
<b>Objeto a analizar:</b> Información del módulo de estaciones.			
<b>Objetivo:</b> Verificar que se muestre información de las diferentes estaciones existentes en el proyecto.			
<b>Descripción:</b> Se verificará que se dé a conocer la información de cada estación como: <ul style="list-style-type: none"> <li>• Nombre de la estación.</li> <li>• Cantón al que pertenece.</li> <li>• Parroquia al que pertenece la estación.</li> <li>• La densidad</li> <li>• Coordenadas.</li> </ul>			
<b>Pre condiciones</b>	<ol style="list-style-type: none"> <li>1. Sitio web en funcionamiento.</li> <li>2. Módulo Estaciones en funcionamiento.</li> <li>3. Sección Estaciones en habilitado.</li> </ol>		
<b>Entradas</b>	Ninguna.		
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Ingresar al Navegador.</li> <li>2. Digitar la URL en la barra de búsqueda.</li> <li>3. Ingresar al Sitio Web.</li> <li>4. Ir al módulo estaciones.</li> </ol>		

	<ol style="list-style-type: none"> <li>5. Ir a la sección de Estaciones.</li> <li>6. Visualizar la descripción.</li> </ol>
<b>Resultados Esperados</b>	Información detallada de cada estación.
<b>Post condiciones</b>	Ninguna.
<b>Observaciones:</b> Ninguna	

**Tabla 104:** Caso de Prueba Elementos\_Clima\_Estaciones

<b>Caso de Prueba N° 11</b>			
<b>Prioridad</b>	Alta	<b>Estado</b>	1
<b>Objeto a analizar:</b> Elementos del clima en la sección estaciones.			
<b>Objetivo:</b> Verificar que se muestre de manera correcta los elementos del clima en la sección estación.			
<b>Descripción:</b> Se verificará que los datos a filtrar correspondan a los elementos del clima, que antes se consideraron en este proyecto.			
<b>Pre condiciones</b>	<ol style="list-style-type: none"> <li>1. Sitio web en funcionamiento.</li> <li>2. Módulo Estaciones en funcionamiento.</li> <li>3. Sección Estaciones en habilitado.</li> </ol>		
<b>Entradas</b>	Ninguna.		
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Ingresar al Navegador.</li> <li>2. Digitar la URL en la barra de búsqueda.</li> <li>3. Ingresar al Sitio Web.</li> <li>4. Ir al módulo estaciones.</li> <li>5. Ir a la sección de Estaciones.</li> <li>6. Visualizar los elementos del clima a través de una lista.</li> </ol>		
<b>Resultados Esperados</b>	Lista desplegable con los elementos del clima.		

<b>Post condiciones</b>	Si no se selecciona un elemento del clima, no se podrá deshabilitar las demás configuraciones.
<b>Observaciones:</b> Ninguna	

**Tabla 105:** Caso de Prueba Periodo\_Existente\_Estaciones

<b>Caso de Prueba N° 12</b>			
<b>Prioridad</b>	Alta	<b>Estado</b>	1
<b>Objeto a analizar:</b> Periodo de tiempo existentes (Años) en la sección estaciones.			
<b>Objetivo:</b> Verificar que se muestre los periodos existentes en la sección estaciones, a través de una lista.			
<b>Descripción:</b> Se verificará que los datos sean traídos desde el back end y sean mostrados a través de una lista			
<b>Pre condiciones</b>	<ol style="list-style-type: none"> <li>1. Sitio web en funcionamiento.</li> <li>2. Módulo Estaciones en funcionamiento.</li> <li>3. Sección Estaciones en habilitado.</li> <li>4. Tener los campos de configuración.</li> <li>5. Seleccionar un elemento del clima.</li> </ol>		
<b>Entradas</b>	Ninguna.		
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Ingresar al Navegador.</li> <li>2. Digitar la URL en la barra de búsqueda.</li> <li>3. Ingresar al Sitio Web.</li> <li>4. Ir al módulo estaciones.</li> <li>5. Ir a la sección de Estaciones.</li> <li>6. Seleccionar un elemento del clima.</li> <li>7. Visualizar los años existentes.</li> </ol>		
<b>Resultados Esperados</b>	Lista desplegable con los años existentes.		

<b>Post condiciones</b>	Si no se selecciona un periodo de tiempo, no se podrá deshabilitar las demás configuraciones.
<b>Observaciones:</b> Ninguna	

**Tabla 106:** Caso de Prueba Rango\_Fechas\_General

<b>Caso de Prueba N° 13</b>			
<b>Prioridad</b>	Baja	<b>Estado</b>	1
<b>Objeto a analizar:</b> Fechas dentro del rango del año seleccionado en la sección de estaciones.			
<b>Objetivo:</b> Verificar que las fechas tanto “Desde” como “Hasta” se muestren dentro del rango del año seleccionado.			
<b>Descripción:</b> Se verificará que las fechas estén en el rango del año seleccionado, para que los gráficos se personalicen dentro de dicho rango.			
<b>Pre condiciones</b>	<ol style="list-style-type: none"> <li>1. Sitio web en funcionamiento.</li> <li>2. Módulo Estaciones en funcionamiento.</li> <li>3. Sección Estaciones en habilitado.</li> <li>4. Tener los campos de configuración.</li> <li>5. Seleccionar un elemento del clima.</li> <li>6. Seleccionar un periodo de tiempo (Años).</li> </ol>		
<b>Entradas</b>	Ninguna.		
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Ingresar al Navegador.</li> <li>2. Digitar la URL en la barra de búsqueda.</li> <li>3. Ingresar al Sitio Web.</li> <li>4. Ir al módulo estaciones.</li> <li>5. Ir a la sección de Estaciones.</li> <li>6. Seleccionar un elemento del clima.</li> <li>7. Seleccionar un periodo de tiempo existente.</li> </ol>		

	8. Visualizar de los rangos del año seleccionado.
<b>Resultados Esperados</b>	Rango de fechas.
<b>Post condiciones</b>	Ninguna.
<b>Observaciones:</b> Ninguna	

**Tabla 107:** Caso de Prueba Temperatura\_Gráfico\_Estaciones

<b>Caso de Prueba N° 14</b>			
<b>Prioridad</b>	Alta	<b>Estado</b>	1
<b>Objeto a analizar:</b> Visualización del gráfico correspondiente a temperatura dentro de las estaciones.			
<b>Objetivo:</b> Verificar que se muestre el gráfico de temperatura y a la par la severidad de las estaciones en general.			
<b>Descripción:</b> Se verificará que los datos obtenidos desde la RES API, sean mostrados de manera correcta a través de un gráfico.			
<b>Pre condiciones</b>	<ol style="list-style-type: none"> <li>1. Sitio web en funcionamiento.</li> <li>2. Módulo Estaciones en funcionamiento.</li> <li>3. Sección Estaciones en habilitado.</li> <li>4. Tener los campos de configuración.</li> </ol>		
<b>Entradas</b>	Ninguna.		
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Ingresar al Navegador.</li> <li>2. Digitar la URL en la barra de búsqueda.</li> <li>3. Ingresar al Sitio Web.</li> <li>4. Ir al módulo estaciones.</li> <li>5. Ir a la sección de Estaciones.</li> <li>6. Seleccionar temperatura como uno de los elementos del clima.</li> <li>7. Seleccionar un periodo de tiempo existente</li> </ol>		

	8. Visualizar gráfico.
<b>Resultados Esperados</b>	Gráfico de temperatura y severidad de las estaciones en general.
<b>Post condiciones</b>	Ninguna.
<b>Observaciones:</b> Ninguna	

## Back End

### Estaciones

**Tabla 108:** Caso de Prueba Get\_Estaciones

Caso de Prueba N° 15			
<b>Prioridad</b>	Alta	<b>Estado</b>	1
<b>Objeto a analizar:</b> Ruta “/api/estaciones”			
<b>Objetivo:</b> Verificar el correcto funcionamiento de la petición GET a la ruta “/api/estaciones”.			
<b>Descripción:</b> Se realizará una petición GET a la ruta “/api/estaciones” de la REST API, con el fin de verificar que exista una respuesta y que los datos presentes en la misma estén correctos.			
<b>Pre condiciones</b>	<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>		
<b>Entradas</b>	Ninguna		
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> <li>2. Crear y Configurar una nueva petición.</li> <li>3. Ejecutar la consulta.</li> </ol>		

<b>Resultados Esperados</b>	Archivo JSON con tres elementos: <b>Ejecución</b> , <b>Mensaje</b> y <b>Resultado</b> , este último debe contener los años y meses que contengan datos relacionados al clima, con sus respectivas fechas mínimas y máximas.
<b>Post condiciones</b>	Si existe algún error al procesar la consulta, el archivo JSON debe contener el elemento <b>Ejecución</b> en Falso y un <b>Mensaje</b> que indique que se debe contactar al administrador, conjuntamente con la creación de un nuevo registro en el archivo “ <b>error.log</b> ”.
<b>Observaciones:</b> Ninguna	

**Tabla 109:** Caso de Prueba Post\_CS\_Mensual

<b>Caso de Prueba N° 16</b>			
<b>Prioridad</b>	Alta	<b>Estado</b>	1
<b>Objeto a analizar:</b> Ruta “/api/estaciones”			
<b>Objetivo:</b> Verificar el correcto funcionamiento de la petición POST a la ruta “/api/estaciones”.			
<b>Descripción:</b> Se realizará una petición POST a la ruta “/api/estaciones” de la REST API, con el fin de verificar que exista una respuesta y que los datos presentes en la misma estén correctos.			
<b>Pre condiciones</b>	<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>		

<b>Entradas</b>	<ol style="list-style-type: none"> <li>1. <b>Estación:</b> identificador de la estación de la cual se desea consultar los datos.</li> <li>2. <b>Fecha de Inicio:</b> Fecha Inicial del periodo donde se consultan datos.</li> <li>3. <b>Fecha de Fin:</b> Fecha Final del periodo donde se consultan los datos.</li> </ol>
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> <li>2. Crear y Configurar una nueva petición.</li> <li>3. Generar el archivo JSON con los campos necesarios para el envío.</li> <li>4. Ejecutar la consulta.</li> </ol>
<b>Resultados Esperados</b>	<p>Devolución de un archivo JSON con los campos: <b>Ejecución, Mensaje y Resultado</b>, donde este último debe contener tres subcampos: <b>Clima, Meses y Severidad</b>, donde clima y severidad contendrán los promedios mensuales de sus respectivas medidas, mientras que meses contendrá los meses disponibles en las medidas de clima y severidad.</p>
<b>Post condiciones</b>	<p>Si existe algún error al procesar la consulta, el archivo JSON debe contener el elemento <b>Ejecución</b> en Falso y un <b>Mensaje</b> que indique que se debe contactar al administrador, conjuntamente con la creación de un nuevo registro en el archivo “<b>error.log</b>”.</p>
<p><b>Observaciones:</b> Ninguna</p>	

**Tabla 110:** Caso de Prueba Post\_CS\_Mensual\_1

<b>Caso de Prueba N° 17</b>			
<b>Prioridad</b>	Media	<b>Estado</b>	1

<b>Objeto a analizar:</b> Ruta “/api/estaciones”	
<b>Objetivo:</b> Verificar la validación de los campos de la petición POST a la ruta “/api/estaciones”.	
<b>Descripción:</b> Se realizará una petición POST a la ruta “/api/estaciones” de la REST API, con el fin de verificar que exista una respuesta y la correcta validación de los diferentes campos enviados en la consulta.	
<b>Pre condiciones</b>	<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>
<b>Entradas</b>	<ol style="list-style-type: none"> <li>1. <b>Estación:</b> Tiene que ser de tipo entero.</li> <li>2. <b>Fecha de Inicio:</b> Tiene que ser una cadena de caracteres con el siguiente formato: “año-mes-día”. Ej: 2021-26-9.</li> <li>3. <b>Fecha de Fin:</b> Tiene que ser una cadena de caracteres con el siguiente formato: “año-mes-día”. Ej: 2021-26-9.</li> </ol>
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> <li>2. Crear y Configurar una nueva petición.</li> <li>3. Generar el archivo JSON con los campos necesarios para él envío. Dichos campos se llenarán con tipos de datos inadecuados.</li> <li>4. Ejecutar la consulta.</li> </ol>
<b>Resultados Esperados</b>	Devolución de un archivo JSON, que contenga los campos: <b>Ejecución, Mensaje y Resultado</b> , donde <b>ejecución</b> debe ser falso, <b>mensaje</b> contener información sobre el error en el tipo de los datos y resultado debe ser nulo.
<b>Post condiciones</b>	Ninguna

<p><b>Observaciones:</b></p> <p>Ninguna</p>
---

**Tabla 111:** Caso de Prueba Post\_CS\_Mensual\_2

<b>Caso de Prueba N° 18</b>	
<b>Prioridad</b>	Media
<b>Estado</b>	1
<p><b>Objeto a analizar:</b></p> <p>Ruta “/api/estaciones”</p>	
<p><b>Objetivo:</b></p> <p>Verificar la validación de los campos obligatorios de la petición POST a la ruta “/api/estaciones”.</p>	
<p><b>Descripción:</b></p> <p>Se realizará una petición POST a la ruta “/api/estaciones” de la REST API, con el fin de verificar que exista una respuesta y la validación de campos obligatorios de la consulta.</p>	
<p><b>Pre condiciones</b></p>	<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>
<p><b>Entradas</b></p>	<ol style="list-style-type: none"> <li>1. <b>Estación:</b> Campo Obligatorio.</li> <li>2. <b>Fecha de Inicio:</b> Campo Obligatorio.</li> <li>3. <b>Fecha de Fin:</b> Campo Obligatorio.</li> </ol>
<p><b>Pasos</b></p>	<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> <li>2. Crear y Configurar una nueva petición.</li> </ol>

	<p>3. Enviar un archivo JSON vacío.</p> <p>4. Ejecutar la consulta.</p>
<b>Resultados Esperados</b>	Devolución de un archivo JSON, que contenga los campos: <b>Ejecución, Mensaje y Resultado</b> , donde <b>ejecución</b> debe ser falso, <b>mensaje</b> contener información sobre los campos faltantes dentro de la consulta y resultado debe ser nulo.
<b>Post condiciones</b>	Ninguna
<b>Observaciones:</b> Ninguna	

**Tabla 112:** Caso de Prueba Post\_CS\_Mensual\_3

<b>Caso de Prueba N° 19</b>			
<b>Prioridad</b>	Media	<b>Estado</b>	1
<b>Objeto a analizar:</b> Ruta “/api/estaciones”			
<b>Objetivo:</b> Verificar la correcta validación de que las fechas pertenezcan al mismo año en la petición POST a la ruta “/api/estaciones”.			
<b>Descripción:</b> Se realizará una petición POST a la ruta “/api/estaciones” de la REST API, con el fin de verificar que exista una respuesta y que se validen correctamente que la <b>fecha de inicio</b> y <b>fecha de fin</b> pertenezcan al mismo año.			
<b>Pre condiciones</b>	<p>1. Base de datos y procedimientos almacenados en funcionamiento.</p> <p>2. Servidor de la REST API en funcionamiento.</p>		

	3. Habilitar las consultas desde cualquier destino.
<b>Entradas</b>	<ol style="list-style-type: none"> <li>1. Estación.</li> <li>2. Fecha de Inicio.</li> <li>3. Fecha de Fin.</li> </ol>
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> <li>2. Crear y Configurar una nueva petición.</li> <li>3. Generar el archivo JSON con los campos necesarios para él envío. Se llenarán las fechas con valores de distintos años.</li> <li>4. Ejecutar la consulta.</li> </ol>
<b>Resultados Esperados</b>	Devolución de un archivo JSON, que contenga los campos: <b>Ejecución, Mensaje y Resultado</b> , donde <b>ejecución</b> debe ser falso, el <b>mensaje</b> debe indicar que las fechas deben pertenecer al mismo año y resultado debe ser nulo.
<b>Post condiciones</b>	Ninguna
<b>Observaciones:</b> Ninguna	

**Tabla 113:** Caso de Prueba Post\_CS\_Mensual\_4

<b>Caso de Prueba N° 20</b>			
<b>Prioridad</b>	Media	<b>Estado</b>	1
<b>Objeto a analizar:</b> Ruta “/api/estaciones”			
<b>Objetivo:</b>			

<p>Verificar la correcta validación de que las fechas sean coherentes en la petición POST a la ruta “/api/estaciones”.</p>	
<p><b>Descripción:</b>  Se realizará una petición POST a la ruta “/api/estaciones” de la REST API, con el fin de verificar que exista una respuesta y que se validen correctamente que la <b>fecha de inicio</b> sea menor que la <b>fecha de fin</b>.</p>	
<p><b>Pre condiciones</b></p>	<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>
<p><b>Entradas</b></p>	<ol style="list-style-type: none"> <li>1. Estación.</li> <li>2. Fecha de Inicio.</li> <li>3. Fecha de Fin.</li> </ol>
<p><b>Pasos</b></p>	<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> <li>2. Crear y Configurar una nueva petición.</li> <li>3. Generar el archivo JSON con los campos necesarios para él envío. Se llenará el campo de la fecha de inicio con un valor mayor a la fecha de fin.</li> <li>4. Ejecutar la consulta.</li> </ol>
<p><b>Resultados Esperados</b></p>	<p>Devolución de un archivo JSON, que contenga los campos: <b>Ejecución, Mensaje y Resultado</b>, donde <b>ejecución</b> debe ser falso, el <b>mensaje</b> debe indicar que la fecha de inicio</p>

	debe ser menor que la fecha de fin y resultado debe ser nulo.
<b>Post condiciones</b>	Ninguna
<b>Observaciones:</b> Ninguna	

**Tabla 114:** Caso de Prueba Post\_CS\_Semanal

<b>Caso de Prueba N° 21</b>			
<b>Prioridad</b>	Alta	<b>Estado</b>	1
<b>Objeto a analizar:</b> Ruta “/api/estaciones/semanas”			
<b>Objetivo:</b> Verificar el correcto funcionamiento de la petición POST a la ruta “/api/estaciones/semanas”.			
<b>Descripción:</b> Se realizará una petición POST a la ruta “/api/estaciones/semanas” de la REST API, con el fin de verificar que exista una respuesta y que los datos presentes en la misma estén correctos.			
<b>Pre condiciones</b>		<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>	
<b>Entradas</b>		<ol style="list-style-type: none"> <li>1. <b>Estación:</b> identificador de la estación de la cual se desea consultar los datos.</li> <li>2. <b>Mes:</b> Especifica el mes del cual se desea obtener la consulta.</li> </ol>	

	<p>3. <b>Fecha de Inicio:</b> Fecha Inicial del periodo donde se consultan datos.</p> <p>4. <b>Fecha de Fin:</b> Fecha Final del periodo donde se consultan los datos.</p>
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> <li>2. Crear y Configurar una nueva petición.</li> <li>3. Generar el archivo JSON con los campos necesarios para él envió.</li> <li>4. Ejecutar la consulta.</li> </ol>
<b>Resultados Esperados</b>	<p>Devolución de un archivo JSON con los campos: <b>Ejecución, Mensaje y Resultado</b>, donde este último debe contener tres subcampos: <b>Clima, Semanas y Severidad</b>, donde clima y severidad contendrán los promedios semanales de sus respectivas medidas, mientras que semanas contendrá las semanas disponibles en las medidas de clima y severidad.</p>
<b>Post condiciones</b>	<p>Si existe algún error al procesar la consulta, el archivo JSON debe contener el elemento <b>Ejecución</b> en Falso y un <b>Mensaje</b> que indique que se debe contactar al administrador, conjuntamente con la creación de un nuevo registro en el archivo <b>“error.log”</b>.</p>

**Observaciones:**

Ninguna

**Tabla 115:** Caso de Prueba Post\_CS\_Semanal\_1

<b>Caso de Prueba N° 22</b>			
<b>Prioridad</b>	Media	<b>Estado</b>	1
<b>Objeto a analizar:</b> Ruta “/api/estaciones/semanas”			
<b>Objetivo:</b> Verificar la validación de los campos de la petición POST a la ruta “/api/estaciones/semanas”.			
<b>Descripción:</b> Se realizará una petición POST a la ruta “/api/estaciones/semanas” de la REST API, con el fin de verificar que exista una respuesta y la correcta validación de los diferentes campos enviados en la consulta.			
<b>Pre condiciones</b>		<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>	
<b>Entradas</b>		<ol style="list-style-type: none"> <li>1. <b>Estación:</b> Tiene que ser de tipo entero.</li> <li>2. <b>Mes:</b> Tiene que ser una cadena de caracteres de una longitud máxima de 100.</li> <li>3. <b>Fecha de Inicio:</b> Tiene que ser una cadena de caracteres con el siguiente formato: “año-mes-día”. Ej: 2021-26-9.</li> </ol>	

	4. <b>Fecha de Fin:</b> Tiene que ser una cadena de caracteres con el siguiente formato: “año-mes-día”. Ej: 2021-26-9.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> <li>2. Crear y Configurar una nueva petición.</li> <li>3. Generar el archivo JSON con los campos necesarios para él envió. Dichos campos se llenarán con tipos de datos inadecuados.</li> <li>4. Ejecutar la consulta.</li> </ol>
<b>Resultados Esperados</b>	Devolución de un archivo JSON, que contenga los campos: <b>Ejecución, Mensaje y Resultado</b> , donde <b>ejecución</b> debe ser falso, <b>mensaje</b> contener información sobre el error en el tipo de los datos y resultado debe ser nulo.
<b>Post condiciones</b>	Ninguna
<b>Observaciones:</b> Ninguna	

**Tabla 116:** Caso de Prueba Post\_CS\_Semanal\_2

<b>Caso de Prueba N° 23</b>			
<b>Prioridad</b>	Media	<b>Estado</b>	1
<b>Objeto a analizar:</b> Ruta “/api/estaciones/semanas”			
<b>Objetivo:</b>			

<p>Verificar la validación de los campos obligatorios de la petición POST a la ruta “/api/estaciones/semanas”.</p>	
<p><b>Descripción:</b> Se realizará una petición POST a la ruta “/api/estaciones/semanas” de la REST API, con el fin de verificar que exista una respuesta y la validación de campos obligatorios de la consulta.</p>	
<p><b>Pre condiciones</b></p>	<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>
<p><b>Entradas</b></p>	<ol style="list-style-type: none"> <li>1. <b>Estación:</b> Campo Obligatorio.</li> <li>2. <b>Mes:</b> Campo Obligatorio.</li> <li>3. <b>Fecha de Inicio:</b> Campo Obligatorio.</li> <li>4. <b>Fecha de Fin:</b> Campo Obligatorio.</li> </ol>
<p><b>Pasos</b></p>	<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> <li>2. Crear y Configurar una nueva petición.</li> <li>3. Enviar un archivo JSON vacío.</li> <li>4. Ejecutar la consulta.</li> </ol>
<p><b>Resultados Esperados</b></p>	<p>Devolución de un archivo JSON, que contenga los campos: <b>Ejecución, Mensaje y Resultado</b>, donde <b>ejecución</b> debe ser falso, <b>mensaje</b> contener información sobre los</p>

	campos faltantes dentro de la consulta y resultado debe ser nulo.
<b>Post condiciones</b>	Ninguna
<b>Observaciones:</b> Ninguna	

**Tabla 117:** Caso de Prueba Post\_CS\_Semanal\_3

<b>Caso de Prueba N° 24</b>			
<b>Prioridad</b>	Media	<b>Estado</b>	1
<b>Objeto a analizar:</b> Ruta “/api/estaciones/semanas”			
<b>Objetivo:</b> Verificar la correcta validación de que las fechas pertenezcan al mismo año en la petición POST a la ruta “/api/estaciones/semanas”.			
<b>Descripción:</b> Se realizará una petición POST a la ruta “/api/estaciones/semanas” de la REST API, con el fin de verificar que exista una respuesta y que se validen correctamente que la <b>fecha de inicio</b> y <b>fecha de fin</b> pertenezcan al mismo año.			
<b>Pre condiciones</b>	<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>		
<b>Entradas</b>	<ol style="list-style-type: none"> <li>1. Estación.</li> <li>2. Mes.</li> <li>3. Fecha de Inicio.</li> <li>4. Fecha de Fin.</li> </ol>		
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> </ol>		

	<ol style="list-style-type: none"> <li>2. Crear y Configurar una nueva petición.</li> <li>3. Generar el archivo JSON con los campos necesarios para él envío. Se llenarán las fechas con valores de distintos años.</li> <li>4. Ejecutar la consulta.</li> </ol>
<b>Resultados Esperados</b>	Devolución de un archivo JSON, que contenga los campos: <b>Ejecución, Mensaje y Resultado</b> , donde <b>ejecución</b> debe ser falso, el <b>mensaje</b> debe indicar que las fechas deben pertenecer al mismo año y resultado debe ser nulo.
<b>Post condiciones</b>	Ninguna
<b>Observaciones:</b> Ninguna	

**Tabla 118:** Caso de Prueba Post\_CS\_Semanal\_4

<b>Caso de Prueba N° 25</b>			
<b>Prioridad</b>	Media	<b>Estado</b>	1
<b>Objeto a analizar:</b> Ruta “/api/estaciones/semanas”			
<b>Objetivo:</b> Verificar la correcta validación de que las fechas sean coherentes en la petición POST a la ruta “/api/estaciones/semanas”.			
<b>Descripción:</b> Se realizará una petición POST a la ruta “/api/estaciones/semanas” de la REST API, con el fin de verificar que exista una respuesta y que se validen correctamente que la <b>fecha de inicio</b> sea menor que la <b>fecha de fin</b> .			

<b>Pre condiciones</b>	<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>
<b>Entradas</b>	<ol style="list-style-type: none"> <li>1. Estación.</li> <li>2. Mes.</li> <li>3. Fecha de Inicio.</li> <li>4. Fecha de Fin.</li> </ol>
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> <li>2. Crear y Configurar una nueva petición.</li> <li>3. Generar el archivo JSON con los campos necesarios para él envió. Se llenará el campo de la fecha de inicio con un valor mayor a la fecha de fin.</li> <li>4. Ejecutar la consulta.</li> </ol>
<b>Resultados Esperados</b>	Devolución de un archivo JSON, que contenga los campos: <b>Ejecución, Mensaje y Resultado</b> , donde <b>ejecución</b> debe ser falso, el <b>mensaje</b> debe indicar que la fecha de inicio debe ser menor que la fecha de fin y resultado debe ser nulo.
<b>Post condiciones</b>	Ninguna
<b>Observaciones:</b> Ninguna	

**Tabla 119:** Caso de Prueba Post\_CS\_Monitoreos

<b>Caso de Prueba N° 26</b>			
<b>Prioridad</b>	Alta	<b>Estado</b>	1
<b>Objeto a analizar:</b> Ruta “/api/estaciones/monitoreos”			
<b>Objetivo:</b> Verificar el correcto funcionamiento de la petición POST a la ruta “/api/estaciones/monitoreos”.			
<b>Descripción:</b> Se realizará una petición POST a la ruta “/api/estaciones/monitoreos” de la REST API, con el fin de verificar que exista una respuesta y que los datos presentes en la misma estén correctos.			
<b>Pre condiciones</b>		<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>	
<b>Entradas</b>		<ol style="list-style-type: none"> <li>1. <b>Estación:</b> identificador de la estación de la cual se desea consultar los datos.</li> <li>2. <b>Fecha de Inicio:</b> Fecha Inicial del periodo donde se consultan datos.</li> <li>3. <b>Fecha de Fin:</b> Fecha Final del periodo donde se consultan los datos.</li> </ol>	
<b>Pasos</b>		<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> <li>2. Crear y Configurar una nueva petición.</li> </ol>	

	<p>3. Generar el archivo JSON con los campos necesarios para él envió.</p> <p>4. Ejecutar la consulta.</p>
<b>Resultados Esperados</b>	Devolución de un archivo JSON con los campos: <b>Ejecución, Mensaje y Resultado</b> , donde este último debe contener dos subcampos: <b>Clima y Severidad</b> , donde clima y severidad contendrán los promedios en función de los monitoreos de sus respectivas medidas.
<b>Post condiciones</b>	Si existe algún error al procesar la consulta, el archivo JSON debe contener el elemento <b>Ejecución</b> en Falso y un <b>Mensaje</b> que indique que se debe contactar al administrador, conjuntamente con la creación de un nuevo registro en el archivo <b>“error.log”</b> .
<b>Observaciones:</b> Ninguna	

**Tabla 120:** Caso de Prueba Post\_CS\_Monitoreos\_1

<b>Caso de Prueba N° 27</b>			
<b>Prioridad</b>	Media	<b>Estado</b>	1
<b>Objeto a analizar:</b> Ruta “/api/estaciones/monitoreos”			
<b>Objetivo:</b> Verificar la validación de los campos de la petición POST a la ruta “/api/estaciones/monitoreos”.			
<b>Descripción:</b>			

<p>Se realizará una petición POST a la ruta “/api/estaciones/monitoreos” de la REST API, con el fin de verificar que exista una respuesta y la correcta validación de los diferentes campos enviados en la consulta.</p>	
<p><b>Pre condiciones</b></p>	<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>
<p><b>Entradas</b></p>	<ol style="list-style-type: none"> <li>1. <b>Estación:</b> Tiene que ser de tipo entero.</li> <li>2. <b>Fecha de Inicio:</b> Tiene que ser una cadena de caracteres con el siguiente formato: “año-mes-día”. Ej: 2021-26-9.</li> <li>3. <b>Fecha de Fin:</b> Tiene que ser una cadena de caracteres con el siguiente formato: “año-mes-día”. Ej: 2021-26-9.</li> </ol>
<p><b>Pasos</b></p>	<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> <li>2. Crear y Configurar una nueva petición.</li> <li>3. Generar el archivo JSON con los campos necesarios para él envió. Dichos campos se llenarán con tipos de datos inadecuados.</li> <li>4. Ejecutar la consulta.</li> </ol>

<b>Resultados Esperados</b>	Devolución de un archivo JSON, que contenga los campos: <b>Ejecución, Mensaje y Resultado</b> , donde <b>ejecución</b> debe ser falso, <b>mensaje</b> contener información sobre el error en el tipo de los datos y resultado debe ser nulo.
<b>Post condiciones</b>	Ninguna
<b>Observaciones:</b> Ninguna	

**Tabla 121:** Caso de Prueba Post\_CS\_Monitoreos\_2

<b>Caso de Prueba N° 28</b>			
<b>Prioridad</b>	Media	<b>Estado</b>	1
<b>Objeto a analizar:</b> Ruta “/api/estaciones/monitoreos”			
<b>Objetivo:</b> Verificar la validación de los campos obligatorios de la petición POST a la ruta “/api/estaciones/monitoreos”.			
<b>Descripción:</b> Se realizará una petición POST a la ruta “/api/estaciones/monitoreos” de la REST API, con el fin de verificar que exista una respuesta y la validación de campos obligatorios de la consulta.			
<b>Pre condiciones</b>		<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>	
<b>Entradas</b>		<ol style="list-style-type: none"> <li>1. <b>Estación:</b> Campo Obligatorio.</li> </ol>	

	<p>2. <b>Fecha de Inicio:</b> Campo Obligatorio.</p> <p>3. <b>Fecha de Fin:</b> Campo Obligatorio.</p>
<b>Pasos</b>	<p>1. Ingresar a la Herramienta Postman.</p> <p>2. Crear y Configurar una nueva petición.</p> <p>3. Enviar un archivo JSON vacío.</p> <p>4. Ejecutar la consulta.</p>
<b>Resultados Esperados</b>	<p>Devolución de un archivo JSON, que contenga los campos: <b>Ejecución, Mensaje y Resultado</b>, donde <b>ejecución</b> debe ser falso, <b>mensaje</b> contener información sobre los campos faltantes dentro de la consulta y resultado debe ser nulo.</p>
<b>Post condiciones</b>	Ninguna
<b>Observaciones:</b>	
Ninguna	

**Tabla 122:** Caso de Prueba Post\_CS\_Monitoreos\_3

<b>Caso de Prueba N° 29</b>			
<b>Prioridad</b>	Media	<b>Estado</b>	1
<b>Objeto a analizar:</b>			
Ruta “/api/estaciones/monitoreos”			
<b>Objetivo:</b>			
Verificar la correcta validación de que las fechas pertenezcan al mismo año en la petición POST a la ruta “/api/estaciones/monitoreos”.			
<b>Descripción:</b>			

<p>Se realizará una petición POST a la ruta “/api/estaciones/monitoreos” de la REST API, con el fin de verificar que exista una respuesta y que se validen correctamente que la <b>fecha de inicio</b> y <b>fecha de fin</b> pertenezcan al mismo año.</p>	
<p><b>Pre condiciones</b></p>	<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>
<p><b>Entradas</b></p>	<ol style="list-style-type: none"> <li>1. Estación.</li> <li>2. Fecha de Inicio.</li> <li>3. Fecha de Fin.</li> </ol>
<p><b>Pasos</b></p>	<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> <li>2. Crear y Configurar una nueva petición.</li> <li>3. Generar el archivo JSON con los campos necesarios para él envió. Se llenarán las fechas con valores de distintos años.</li> <li>4. Ejecutar la consulta.</li> </ol>
<p><b>Resultados Esperados</b></p>	<p>Devolución de un archivo JSON, que contenga los campos: <b>Ejecución, Mensaje y Resultado</b>, donde <b>ejecución</b> debe ser falso, el <b>mensaje</b> debe indicar que las fechas deben pertenecer al mismo año y resultado debe ser nulo.</p>
<p><b>Post condiciones</b></p>	<p>Ninguna</p>
<p><b>Observaciones:</b></p> <p>Ninguna</p>	

**Tabla 123:** Caso de Prueba Post\_CS\_Monitoreos\_4

<b>Caso de Prueba N° 30</b>			
<b>Prioridad</b>	Media	<b>Estado</b>	1
<b>Objeto a analizar:</b> Ruta “/api/estaciones/monitoreos”			
<b>Objetivo:</b> Verificar la correcta validación de que las fechas sean coherentes en la petición POST a la ruta “/api/estaciones/monitoreos”.			
<b>Descripción:</b> Se realizará una petición POST a la ruta “/api/estaciones/monitoreos” de la REST API, con el fin de verificar que exista una respuesta y que se validen correctamente que la <b>fecha de inicio</b> sea menor que la <b>fecha de fin</b> .			
<b>Pre condiciones</b>		<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>	
<b>Entradas</b>		<ol style="list-style-type: none"> <li>1. Estación.</li> <li>2. Fecha de Inicio.</li> <li>3. Fecha de Fin.</li> </ol>	
<b>Pasos</b>		<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> <li>2. Crear y Configurar una nueva petición.</li> <li>3. Generar el archivo JSON con los campos necesarios para él envío. Se llenará el campo de la fecha de inicio con un valor mayor a la fecha de fin.</li> </ol>	

	4. Ejecutar la consulta.
<b>Resultados Esperados</b>	Devolución de un archivo JSON, que contenga los campos: <b>Ejecución, Mensaje y Resultado</b> , donde <b>ejecución</b> debe ser falso, el <b>mensaje</b> debe indicar que la fecha de inicio debe ser menor que la fecha de fin y resultado debe ser nulo.
<b>Post condiciones</b>	Ninguna
<b>Observaciones:</b> Ninguna	

**Tabla 124:** Caso de Prueba Post\_CS\_Monitoreos\_Semanas

<b>Caso de Prueba N° 31</b>			
<b>Prioridad</b>	Alta	<b>Estado</b>	1
<b>Objeto a analizar:</b> Ruta “/api/estaciones/monitoreos/semanas”			
<b>Objetivo:</b> Verificar el correcto funcionamiento de la petición POST a la ruta “/api/estaciones/monitoreos/semanas”.			
<b>Descripción:</b> Se realizará una petición POST a la ruta “/api/estaciones/monitoreos/semanas” de la REST API, con el fin de verificar que exista una respuesta y que los datos presentes en la misma estén correctos.			
<b>Pre condiciones</b>		<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>	

<b>Entradas</b>	<ol style="list-style-type: none"> <li>1. <b>Estación:</b> identificador de la estación de la cual se desea consultar los datos.</li> <li>2. <b>ID Tiempo:</b> identificador de tiempo para conocer la fecha en la que se realizó el monitoreo.</li> <li>3. <b>Fecha de Inicio:</b> Fecha Inicial del periodo donde se consultan datos.</li> <li>4. <b>Fecha de Fin:</b> Fecha Final del periodo donde se consultan los datos.</li> </ol>
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> <li>2. Crear y Configurar una nueva petición.</li> <li>3. Generar el archivo JSON con los campos necesarios para él envió.</li> <li>4. Ejecutar la consulta.</li> </ol>
<b>Resultados Esperados</b>	<p>Devolución de un archivo JSON con los campos: <b>Ejecución, Mensaje y Resultado</b>, donde este último debe contener dos subcampos: <b>Clima y Severidad</b>, donde clima y severidad contendrán los promedios en función de los monitoreos de sus respectivas medidas.</p>
<b>Post condiciones</b>	<p>Si existe algún error al procesar la consulta, el archivo JSON debe contener el elemento <b>Ejecución</b> en</p>

	Falso y un <b>Mensaje</b> que indique que se debe contactar al administrador, conjuntamente con la creación de un nuevo registro en el archivo “ <b>error.log</b> ”.
<b>Observaciones:</b> Ninguna	

**Tabla 125:** Caso de Prueba Post\_CS\_Monitoreos\_Semanas\_1

<b>Caso de Prueba N° 32</b>			
<b>Prioridad</b>	Media	<b>Estado</b>	1
<b>Objeto a analizar:</b> Ruta “/api/estaciones/monitoreos/semanas”			
<b>Objetivo:</b> Verificar la validación de los campos de la petición POST a la ruta “/api/estaciones/monitoreos/semanas”.			
<b>Descripción:</b> Se realizará una petición POST a la ruta “/api/estaciones/monitoreos/semanas” de la REST API, con el fin de verificar que exista una respuesta y la correcta validación de los diferentes campos enviados en la consulta.			
<b>Pre condiciones</b>		<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>	
<b>Entradas</b>		<ol style="list-style-type: none"> <li>1. <b>Estación:</b> Tiene que ser de tipo entero.</li> <li>2. <b>ID Tiempo:</b> Tiene que ser de tipo entero.</li> </ol>	

	<p>3. <b>Fecha de Inicio:</b> Tiene que ser una cadena de caracteres con el siguiente formato: “año-mes-día”. Ej: 2021-26-9.</p> <p>4. <b>Fecha de Fin:</b> Tiene que ser una cadena de caracteres con el siguiente formato: “año-mes-día”. Ej: 2021-26-9.</p>
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> <li>2. Crear y Configurar una nueva petición.</li> <li>3. Generar el archivo JSON con los campos necesarios para él envió. Dichos campos se llenarán con tipos de datos inadecuados.</li> <li>4. Ejecutar la consulta.</li> </ol>
<b>Resultados Esperados</b>	<p>Devolución de un archivo JSON, que contenga los campos: <b>Ejecución, Mensaje y Resultado</b>, donde <b>ejecución</b> debe ser falso, <b>mensaje</b> contener información sobre el error en el tipo de los datos y resultado debe ser nulo.</p>
<b>Post condiciones</b>	Ninguna
<b>Observaciones:</b>	
Ninguna	

**Tabla 126:** Caso de Prueba Post\_CS\_Monitoreos\_Semanas\_2

**Caso de Prueba N° 33**

<b>Prioridad</b>	Media	<b>Estado</b>	1
<b>Objeto a analizar:</b> Ruta “/api/estaciones/monitoreos/semanas”			
<b>Objetivo:</b> Verificar la validación de los campos obligatorios de la petición POST a la ruta “/api/estaciones/monitoreos/semanas”.			
<b>Descripción:</b> Se realizará una petición POST a la ruta “/api/estaciones/monitoreos/semanas” de la REST API, con el fin de verificar que exista una respuesta y la validación de campos obligatorios de la consulta.			
<b>Pre condiciones</b>		<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>	
<b>Entradas</b>		<ol style="list-style-type: none"> <li>1. <b>Estación:</b> Campo Obligatorio.</li> <li>2. <b>ID Tiempo:</b> Campo Obligatorio.</li> <li>3. <b>Fecha de Inicio:</b> Campo Obligatorio.</li> <li>4. <b>Fecha de Fin:</b> Campo Obligatorio.</li> </ol>	
<b>Pasos</b>		<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> <li>2. Crear y Configurar una nueva petición.</li> <li>3. Enviar un archivo JSON vacío.</li> <li>4. Ejecutar la consulta.</li> </ol>	

<b>Resultados Esperados</b>	Devolución de un archivo JSON, que contenga los campos: <b>Ejecución, Mensaje y Resultado</b> , donde <b>ejecución</b> debe ser falso, <b>mensaje</b> contener información sobre los campos faltantes dentro de la consulta y resultado debe ser nulo.
<b>Post condiciones</b>	Ninguna
<b>Observaciones:</b> Ninguna	

**Tabla 127:** Caso de Prueba Post\_CS\_Monitoreos\_Semanas\_3

<b>Caso de Prueba N° 34</b>			
<b>Prioridad</b>	Media	<b>Estado</b>	1
<b>Objeto a analizar:</b> Ruta “/api/estaciones/monitoreos/semanas”			
<b>Objetivo:</b> Verificar la correcta validación de que las fechas pertenezcan al mismo año en la petición POST a la ruta “/api/estaciones/monitoreos/semanas”.			
<b>Descripción:</b> Se realizará una petición POST a la ruta “/api/estaciones/monitoreos/semanas” de la REST API, con el fin de verificar que exista una respuesta y que se validen correctamente que la <b>fecha de inicio</b> y <b>fecha de fin</b> pertenezcan al mismo año.			
<b>Pre condiciones</b>		<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>	
<b>Entradas</b>		<ol style="list-style-type: none"> <li>1. Estación.</li> <li>2. ID Tiempo.</li> </ol>	

	3. Fecha de Inicio. 4. Fecha de Fin.
<b>Pasos</b>	1. Ingresar a la Herramienta Postman. 2. Crear y Configurar una nueva petición. 3. Generar el archivo JSON con los campos necesarios para él envío. Se llenarán las fechas con valores de distintos años. 4. Ejecutar la consulta.
<b>Resultados Esperados</b>	Devolución de un archivo JSON, que contenga los campos: <b>Ejecución, Mensaje y Resultado</b> , donde <b>ejecución</b> debe ser falso, el <b>mensaje</b> debe indicar que las fechas deben pertenecer al mismo año y resultado debe ser nulo.
<b>Post condiciones</b>	Ninguna
<b>Observaciones:</b>	
Ninguna	

**Tabla 128:** Caso de Prueba Post\_CS\_Monitoreos\_Semanas\_4

<b>Caso de Prueba N° 35</b>			
<b>Prioridad</b>	Media	<b>Estado</b>	1
<b>Objeto a analizar:</b>			
Ruta “/api/estaciones/monitoreos/semanas”			
<b>Objetivo:</b>			
Verificar la correcta validación de que las fechas sean coherentes en la petición POST a la ruta “/api/estaciones/monitoreos/semanas”.			
<b>Descripción:</b>			

<p>Se realizará una petición POST a la ruta “/api/estaciones/monitoreos/semanas” de la REST API, con el fin de verificar que exista una respuesta y que se validen correctamente que la <b>fecha de inicio</b> sea menor que la <b>fecha de fin</b>.</p>	
<p><b>Pre condiciones</b></p>	<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>
<p><b>Entradas</b></p>	<ol style="list-style-type: none"> <li>1. Estación.</li> <li>2. Fecha de Inicio.</li> <li>3. Fecha de Fin.</li> </ol>
<p><b>Pasos</b></p>	<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> <li>2. Crear y Configurar una nueva petición.</li> <li>3. Generar el archivo JSON con los campos necesarios para él envié. Se llenará el campo de la fecha de inicio con un valor mayor a la fecha de fin.</li> <li>4. Ejecutar la consulta.</li> </ol>
<p><b>Resultados Esperados</b></p>	<p>Devolución de un archivo JSON, que contenga los campos: <b>Ejecución, Mensaje y Resultado</b>, donde <b>ejecución</b> debe ser falso, el <b>mensaje</b> debe indicar que la fecha de inicio debe ser menor que la fecha de fin y resultado debe ser nulo.</p>
<p><b>Post condiciones</b></p>	<p>Ninguna</p>
<p><b>Observaciones:</b></p>	

Ninguna

**Tabla 129:** Caso de Prueba Post\_CS\_Diario

<b>Caso de Prueba N° 36</b>			
<b>Prioridad</b>	Alta	<b>Estado</b>	1
<b>Objeto a analizar:</b> Ruta “/api/estaciones/dias”			
<b>Objetivo:</b> Verificar el correcto funcionamiento de la petición POST a la ruta “/api/estaciones/dias”.			
<b>Descripción:</b> Se realizará una petición POST a la ruta “/api/estaciones/dias” de la REST API, con el fin de verificar que exista una respuesta y que los datos presentes en la misma estén correctos.			
<b>Pre condiciones</b>		<ol style="list-style-type: none"><li>1. Base de datos y procedimientos almacenados en funcionamiento.</li><li>2. Servidor de la REST API en funcionamiento.</li><li>3. Habilitar las consultas desde cualquier destino.</li></ol>	
<b>Entradas</b>		<ol style="list-style-type: none"><li>1. <b>Estación:</b> identificador de la estación de la cual se desea consultar los datos.</li><li>2. <b>Mes:</b> Especifica el mes del cual se desea obtener la consulta.</li><li>3. <b>Semana:</b> Especifica el mes del cual se desea obtener la consulta.</li></ol>	

	<p>4. <b>Fecha de Inicio:</b> Fecha Inicial del periodo donde se consultan datos.</p> <p>5. <b>Fecha de Fin:</b> Fecha Final del periodo donde se consultan los datos.</p>
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> <li>2. Crear y Configurar una nueva petición.</li> <li>3. Generar el archivo JSON con los campos necesarios para él envió.</li> <li>4. Ejecutar la consulta.</li> </ol>
<b>Resultados Esperados</b>	<p>Devolución de un archivo JSON con los campos: <b>Ejecución, Mensaje y Resultado</b>, donde este último debe contener tres subcampos: <b>Clima, Días y Severidad</b>, donde clima y severidad contendrán los promedios diarios de sus respectivas medidas, mientras que días contendrá los días disponibles en las medidas de clima y severidad.</p>
<b>Post condiciones</b>	<p>Si existe algún error al procesar la consulta, el archivo JSON debe contener el elemento <b>Ejecución</b> en Falso y un <b>Mensaje</b> que indique que se debe contactar al administrador, conjuntamente con la creación de un nuevo registro en el archivo <b>“error.log”</b>.</p>

**Observaciones:**

Ninguna

**Tabla 130:** Caso de Prueba Post\_CS\_Diario\_1

<b>Caso de Prueba N° 37</b>			
<b>Prioridad</b>	Media	<b>Estado</b>	1
<b>Objeto a analizar:</b> Ruta “/api/estaciones/dias”			
<b>Objetivo:</b> Verificar la validación de los campos de la petición POST a la ruta “/api/estaciones/dias”.			
<b>Descripción:</b> Se realizará una petición POST a la ruta “/api/estaciones/dias” de la REST API, con el fin de verificar que exista una respuesta y la correcta validación de los diferentes campos enviados en la consulta.			
<b>Pre condiciones</b>		<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>	
<b>Entradas</b>		<ol style="list-style-type: none"> <li>1. <b>Estación:</b> Tiene que ser de tipo entero.</li> <li>2. <b>Mes:</b> Tiene que ser una cadena de caracteres de una longitud máxima de 100.</li> <li>3. <b>Semana:</b> Tiene que ser una cadena de caracteres de una longitud máxima de 100.</li> <li>4. <b>Fecha de Inicio:</b> Tiene que ser una cadena de caracteres</li> </ol>	

	<p>con el siguiente formato: “año-mes-día”. Ej: 2021-26-9.</p> <p>5. <b>Fecha de Fin:</b> Tiene que ser una cadena de caracteres con el siguiente formato: “año-mes-día”. Ej: 2021-26-9.</p>
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> <li>2. Crear y Configurar una nueva petición.</li> <li>3. Generar el archivo JSON con los campos necesarios para él envío. Dichos campos se llenarán con tipos de datos inadecuados.</li> <li>4. Ejecutar la consulta.</li> </ol>
<b>Resultados Esperados</b>	<p>Devolución de un archivo JSON, que contenga los campos: <b>Ejecución, Mensaje y Resultado</b>, donde <b>ejecución</b> debe ser falso, <b>mensaje</b> contener información sobre el error en el tipo de los datos y resultado debe ser nulo.</p>
<b>Post condiciones</b>	Ninguna
<b>Observaciones:</b>	
Ninguna	

**Tabla 131:** Caso de Prueba Post\_CS\_Diario\_2

<b>Caso de Prueba N° 38</b>			
<b>Prioridad</b>	Media	<b>Estado</b>	1
<b>Objeto a analizar:</b>			

Ruta “/api/estaciones/dias”	
<b>Objetivo:</b> Verificar la validación de los campos obligatorios de la petición POST a la ruta “/api/estaciones/dias”.	
<b>Descripción:</b> Se realizará una petición POST a la ruta “/api/estaciones/dias” de la REST API, con el fin de verificar que exista una respuesta y la validación de campos obligatorios de la consulta.	
<b>Pre condiciones</b>	<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>
<b>Entradas</b>	<ol style="list-style-type: none"> <li>1. <b>Estación:</b> Campo Obligatorio.</li> <li>2. <b>Mes:</b> Campo Obligatorio.</li> <li>3. <b>Semana:</b> Campo Obligatorio.</li> <li>4. <b>Fecha de Inicio:</b> Campo Obligatorio.</li> <li>5. <b>Fecha de Fin:</b> Campo Obligatorio.</li> </ol>
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> <li>2. Crear y Configurar una nueva petición.</li> <li>3. Enviar un archivo JSON vacío.</li> <li>4. Ejecutar la consulta.</li> </ol>
<b>Resultados Esperados</b>	Devolución de un archivo JSON, que contenga los campos: <b>Ejecución,</b>

	<b>Mensaje y Resultado</b> , donde <b>ejecución</b> debe ser falso, <b>mensaje</b> contener información sobre los campos faltantes dentro de la consulta y resultado debe ser nulo.
<b>Post condiciones</b>	Ninguna
<b>Observaciones:</b> Ninguna	

**Tabla 132:** Caso de Prueba Post\_CS\_Diario\_3

<b>Caso de Prueba N° 39</b>			
<b>Prioridad</b>	Media	<b>Estado</b>	1
<b>Objeto a analizar:</b> Ruta “/api/estaciones/días”			
<b>Objetivo:</b> Verificar la correcta validación de que las fechas pertenezcan al mismo año en la petición POST a la ruta “/api/estaciones/diario”.			
<b>Descripción:</b> Se realizará una petición POST a la ruta “/api/estaciones/diario” de la REST API, con el fin de verificar que exista una respuesta y que se validen correctamente que la <b>fecha de inicio</b> y <b>fecha de fin</b> pertenezcan al mismo año.			
<b>Pre condiciones</b>		<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>	
<b>Entradas</b>		<ol style="list-style-type: none"> <li>1. Estación.</li> <li>2. Mes.</li> <li>3. Semana.</li> <li>4. Fecha de Inicio.</li> </ol>	

	5. Fecha de Fin.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> <li>2. Crear y Configurar una nueva petición.</li> <li>3. Generar el archivo JSON con los campos necesarios para él envío. Se llenarán las fechas con valores de distintos años.</li> <li>4. Ejecutar la consulta.</li> </ol>
<b>Resultados Esperados</b>	Devolución de un archivo JSON, que contenga los campos: <b>Ejecución, Mensaje y Resultado</b> , donde <b>ejecución</b> debe ser falso, el <b>mensaje</b> debe indicar que las fechas deben pertenecer al mismo año y resultado debe ser nulo.
<b>Post condiciones</b>	Ninguna
<b>Observaciones:</b> Ninguna	

**Tabla 133:** Caso de Prueba Post\_CS\_Diario\_4

<b>Caso de Prueba N° 40</b>			
<b>Prioridad</b>	Media	<b>Estado</b>	1
<b>Objeto a analizar:</b> Ruta “/api/estaciones/diario”			
<b>Objetivo:</b> Verificar la correcta validación de que las fechas sean coherentes en la petición POST a la ruta “/api/estaciones/diario”.			
<b>Descripción:</b>			

<p>Se realizará una petición POST a la ruta “/api/estaciones/diario” de la REST API, con el fin de verificar que exista una respuesta y que se validen correctamente que la <b>fecha de inicio</b> sea menor que la <b>fecha de fin</b>.</p>	
<p><b>Pre condiciones</b></p>	<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>
<p><b>Entradas</b></p>	<ol style="list-style-type: none"> <li>1. Estación.</li> <li>2. Mes.</li> <li>3. Semana.</li> <li>4. Fecha de Inicio.</li> <li>5. Fecha de Fin.</li> </ol>
<p><b>Pasos</b></p>	<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> <li>2. Crear y Configurar una nueva petición.</li> <li>3. Generar el archivo JSON con los campos necesarios para él envío. Se llenará el campo de la fecha de inicio con un valor mayor a la fecha de fin.</li> <li>4. Ejecutar la consulta.</li> </ol>
<p><b>Resultados Esperados</b></p>	<p>Devolución de un archivo JSON, que contenga los campos: <b>Ejecución, Mensaje y Resultado</b>, donde <b>ejecución</b> debe ser falso, el <b>mensaje</b> debe indicar que la fecha de inicio debe ser menor que la fecha de fin y resultado debe ser nulo.</p>

<b>Post condiciones</b>	Ninguna
<b>Observaciones:</b> Ninguna	

**Tabla 134:** Caso de Prueba Post\_CS\_Datos

<b>Caso de Prueba N° 41</b>			
<b>Prioridad</b>	Alta	<b>Estado</b>	1
<b>Objeto a analizar:</b> Ruta “/api/estaciones/datos”			
<b>Objetivo:</b> Verificar el correcto funcionamiento de la petición POST a la ruta “/api/estaciones/datos”.			
<b>Descripción:</b> Se realizará una petición POST a la ruta “/api/estaciones/datos” de la REST API, con el fin de verificar que exista una respuesta y que los datos presentes en la misma estén correctos.			
<b>Pre condiciones</b>		<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>	
<b>Entradas</b>		<ol style="list-style-type: none"> <li>1. <b>Estación:</b> identificador de la estación de la cual se desea consultar los datos.</li> </ol>	
<b>Pasos</b>		<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> <li>2. Crear y Configurar una nueva petición.</li> </ol>	

	<p>3. Generar el archivo JSON con los campos necesarios para él envió.</p> <p>4. Ejecutar la consulta.</p>
<b>Resultados Esperados</b>	Devolución de un archivo JSON con los campos: <b>Ejecución</b> , <b>Mensaje</b> y <b>Resultado</b> , donde este ultimo debe contener un subcampo: <b>Estaciones</b> , el cual contiene datos descriptivos de las estaciones.
<b>Post condiciones</b>	Si existe algún error al procesar la consulta, el archivo JSON debe contener el elemento <b>Ejecución</b> en Falso y un <b>Mensaje</b> que indique que se debe contactar al administrador, conjuntamente con la creación de un nuevo registro en el archivo <b>“error.log”</b> .
<b>Observaciones:</b> Ninguna	

**Tabla 135:** Caso de Prueba Post\_CS\_Datos\_1

<b>Caso de Prueba N° 42</b>			
<b>Prioridad</b>	Media	<b>Estado</b>	1
<b>Objeto a analizar:</b> Ruta “/api/estaciones/datos”			
<b>Objetivo:</b> Verificar la validación de los campos de la petición POST a la ruta “/api/estaciones/datos”.			
<b>Descripción:</b>			

Se realizará una petición POST a la ruta “/api/estaciones/datos” de la REST API, con el fin de verificar que exista una respuesta y la correcta validación de los diferentes campos enviados en la consulta.	
<b>Pre condiciones</b>	<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>
<b>Entradas</b>	<ol style="list-style-type: none"> <li>1. <b>Estación:</b> Tiene que ser de tipo entero.</li> </ol>
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> <li>2. Crear y Configurar una nueva petición.</li> <li>3. Generar el archivo JSON con los campos necesarios para él envió. Dichos campos se llenarán con tipos de datos inadecuados.</li> <li>4. Ejecutar la consulta.</li> </ol>
<b>Resultados Esperados</b>	Devolución de un archivo JSON, que contenga los campos: <b>Ejecución, Mensaje y Resultado</b> , donde <b>ejecución</b> debe ser falso, <b>mensaje</b> contener información sobre el error en el tipo de dato y resultado debe ser nulo.
<b>Post condiciones</b>	Ninguna
<b>Observaciones:</b> Ninguna	

**Tabla 136:** Caso de Prueba Post\_CS\_Datos\_2

<b>Caso de Prueba N° 43</b>			
<b>Prioridad</b>	Media	<b>Estado</b>	1
<b>Objeto a analizar:</b> Ruta “/api/estaciones/datos”			
<b>Objetivo:</b> Verificar la validación de los campos obligatorios de la petición POST a la ruta “/api/estaciones/datos”.			
<b>Descripción:</b> Se realizará una petición POST a la ruta “/api/estaciones/datos” de la REST API, con el fin de verificar que exista una respuesta y la validación de campos obligatorios de la consulta.			
<b>Pre condiciones</b>		<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>	
<b>Entradas</b>		<ol style="list-style-type: none"> <li>1. <b>Estación:</b> Campo Obligatorio.</li> </ol>	
<b>Pasos</b>		<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> <li>2. Crear y Configurar una nueva petición.</li> <li>3. Enviar un archivo JSON vacío.</li> <li>4. Ejecutar la consulta.</li> </ol>	
<b>Resultados Esperados</b>		Devolución de un archivo JSON, que contenga los campos: <b>Ejecución, Mensaje y Resultado</b> , donde	

	<b>ejecución</b> debe ser falso, <b>mensaje</b> contener información sobre el campo faltante de la consulta y resultado debe ser nulo.
<b>Post condiciones</b>	Ninguna
<b>Observaciones:</b> Ninguna	

## Incidencia

**Tabla 137:** Caso de Prueba Get\_Incidencia

<b>Caso de Prueba N° 44</b>			
<b>Prioridad</b>	Alta	<b>Estado</b>	1
<b>Objeto a analizar:</b> Ruta “/api/incidencia”			
<b>Objetivo:</b> Verificar el correcto funcionamiento de la petición GET a la ruta “/api/incidencia”.			
<b>Descripción:</b> Se realizará una petición GET a la ruta “/api/incidencia” de la REST API, con el fin de verificar que exista una respuesta y que los datos presentes en la misma estén correctos.			
<b>Pre condiciones</b>		<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>	
<b>Entradas</b>		Ninguna	
<b>Pasos</b>		<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> <li>2. Crear y Configurar una nueva petición.</li> </ol>	

	3. Ejecutar la consulta.
<b>Resultados Esperados</b>	Archivo JSON con tres elementos: <b>Ejecución, Mensaje y Resultado</b> , este último debe contener los años que contengan datos relacionados a la incidencia, con sus respectivas fechas mínimas y máximas.
<b>Post condiciones</b>	Si existe algún error al procesar la consulta, el archivo JSON debe contener el elemento <b>Ejecución</b> en Falso y un <b>Mensaje</b> que indique que se debe contactar al administrador, conjuntamente con la creación de un nuevo registro en el archivo <b>“error.log”</b> .
<b>Observaciones:</b> Ninguna	

**Tabla 138:** Caso de Prueba Post\_Incidencia\_Mensual

<b>Caso de Prueba N° 45</b>			
<b>Prioridad</b>	Alta	<b>Estado</b>	1
<b>Objeto a analizar:</b> Ruta “/api/incidencia”			
<b>Objetivo:</b> Verificar el correcto funcionamiento de la petición POST a la ruta “/api/incidencia”.			
<b>Descripción:</b> Se realizará una petición POST a la ruta “/api/incidencia” de la REST API, con el fin de verificar que exista una respuesta y que los datos presentes en la misma estén correctos.			

<b>Pre condiciones</b>	<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>
<b>Entradas</b>	<ol style="list-style-type: none"> <li>1. <b>Fecha de Inicio:</b> Fecha Inicial del periodo donde se consultan datos.</li> <li>2. <b>Fecha de Fin:</b> Fecha Final del periodo donde se consultan los datos.</li> </ol>
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> <li>2. Crear y Configurar una nueva petición.</li> <li>3. Generar el archivo JSON con los campos necesarios para él envió.</li> <li>4. Ejecutar la consulta.</li> </ol>
<b>Resultados Esperados</b>	<p>Devolución de un archivo JSON con los campos: <b>Ejecución, Mensaje y Resultado</b>, donde este último debe contener un subcampo: <b>Incidencia</b>, que contendrá los promedios mensuales de dicha medida.</p>
<b>Post condiciones</b>	<p>Si existe algún error al procesar la consulta, el archivo JSON debe contener el elemento <b>Ejecución</b> en Falso y un <b>Mensaje</b> que indique que se debe contactar al administrador,</p>

	conjuntamente con la creación de un nuevo registro en el archivo “error.log”.
<b>Observaciones:</b> Ninguna	

**Tabla 139:** Caso de Prueba Post\_Incidencia\_Mensual\_1

<b>Caso de Prueba N° 46</b>			
<b>Prioridad</b>	Media	<b>Estado</b>	1
<b>Objeto a analizar:</b> Ruta “/api/incidencia”			
<b>Objetivo:</b> Verificar la validación de los campos de la petición POST a la ruta “/api/incidencia”.			
<b>Descripción:</b> Se realizará una petición POST a la ruta “/api/incidencia” de la REST API, con el fin de verificar que exista una respuesta y la correcta validación de los diferentes campos enviados en la consulta.			
<b>Pre condiciones</b>		<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>	
<b>Entradas</b>		<ol style="list-style-type: none"> <li>1. <b>Fecha de Inicio:</b> Tiene que ser una cadena de caracteres con el siguiente formato: “año-mes-día”. Ej: 2021-26-9.</li> <li>2. <b>Fecha de Fin:</b> Tiene que ser una cadena de caracteres con</li> </ol>	

	el siguiente formato: “año-mes-día”. Ej: 2021-26-9.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> <li>2. Crear y Configurar una nueva petición.</li> <li>3. Generar el archivo JSON con los campos necesarios para él envió. Dichos campos se llenarán con tipos de datos inadecuados.</li> <li>4. Ejecutar la consulta.</li> </ol>
<b>Resultados Esperados</b>	Devolución de un archivo JSON, que contenga los campos: <b>Ejecución, Mensaje y Resultado</b> , donde <b>ejecución</b> debe ser falso, <b>mensaje</b> contener información sobre el error en el tipo de los datos y resultado debe ser nulo.
<b>Post condiciones</b>	Ninguna
<b>Observaciones:</b> Ninguna	

**Tabla 140:** Caso de Prueba Post\_Incidencia\_Mensual\_2

<b>Caso de Prueba N° 47</b>			
<b>Prioridad</b>	Media	<b>Estado</b>	1
<b>Objeto a analizar:</b> Ruta “/api/incidencia”			
<b>Objetivo:</b> Verificar la validación de los campos obligatorios de la petición POST a la ruta “/api/incidencia”.			
<b>Descripción:</b>			

Se realizará una petición POST a la ruta “/api/incidencia” de la REST API, con el fin de verificar que exista una respuesta y la validación de campos obligatorios de la consulta.	
<b>Pre condiciones</b>	<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>
<b>Entradas</b>	<ol style="list-style-type: none"> <li>1. <b>Fecha de Inicio:</b> Campo Obligatorio.</li> <li>2. <b>Fecha de Fin:</b> Campo Obligatorio.</li> </ol>
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> <li>2. Crear y Configurar una nueva petición.</li> <li>3. Enviar un archivo JSON vacío.</li> <li>4. Ejecutar la consulta.</li> </ol>
<b>Resultados Esperados</b>	Devolución de un archivo JSON, que contenga los campos: <b>Ejecución, Mensaje y Resultado</b> , donde <b>ejecución</b> debe ser falso, <b>mensaje</b> contener información sobre los campos faltantes dentro de la consulta y resultado debe ser nulo.
<b>Post condiciones</b>	Ninguna
<b>Observaciones:</b> Ninguna	

**Tabla 141:** Caso de Prueba Post\_Incidencia\_Mensual\_3

<b>Caso de Prueba N° 48</b>			
<b>Prioridad</b>	Media	<b>Estado</b>	1
<b>Objeto a analizar:</b> Ruta “/api/incidencia”			
<b>Objetivo:</b> Verificar la correcta validación de que las fechas pertenezcan al mismo año en la petición POST a la ruta “/api/incidencia”.			
<b>Descripción:</b> Se realizará una petición POST a la ruta “/api/incidencia” de la REST API, con el fin de verificar que exista una respuesta y que se validen correctamente que la <b>fecha de inicio</b> y <b>fecha de fin</b> pertenezcan al mismo año.			
<b>Pre condiciones</b>		<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>	
<b>Entradas</b>		<ol style="list-style-type: none"> <li>1. Fecha de Inicio.</li> <li>2. Fecha de Fin.</li> </ol>	
<b>Pasos</b>		<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> <li>2. Crear y Configurar una nueva petición.</li> <li>3. Generar el archivo JSON con los campos necesarios para él envió. Se llenarán las fechas con valores de distintos años.</li> <li>4. Ejecutar la consulta.</li> </ol>	
<b>Resultados Esperados</b>		Devolución de un archivo JSON, que contenga los campos: <b>Ejecución,</b>	

	<b>Mensaje y Resultado</b> , donde <b>ejecución</b> debe ser falso, el <b>mensaje</b> debe indicar que las fechas deben pertenecer al mismo año y resultado debe ser nulo.
<b>Post condiciones</b>	Ninguna
<b>Observaciones:</b> Ninguna	

**Tabla 142:** Caso de Prueba Post\_Incidencia\_Mensual\_4

<b>Caso de Prueba N° 49</b>			
<b>Prioridad</b>	Media	<b>Estado</b>	1
<b>Objeto a analizar:</b> Ruta “/api/incidencia”			
<b>Objetivo:</b> Verificar la correcta validación de que las fechas sean coherentes en la petición POST a la ruta “/api/incidencia”.			
<b>Descripción:</b> Se realizará una petición POST a la ruta “/api/incidencia” de la REST API, con el fin de verificar que exista una respuesta y que se validen correctamente que la <b>fecha de inicio</b> sea menor que la <b>fecha de fin</b> .			
<b>Pre condiciones</b>		<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>	
<b>Entradas</b>		<ol style="list-style-type: none"> <li>1. Fecha de Inicio.</li> <li>2. Fecha de Fin.</li> </ol>	
<b>Pasos</b>		<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> </ol>	

	<ol style="list-style-type: none"> <li>2. Crear y Configurar una nueva petición.</li> <li>3. Generar el archivo JSON con los campos necesarios para él envío. Se llenará el campo de la fecha de inicio con un valor mayor a la fecha de fin.</li> <li>4. Ejecutar la consulta.</li> </ol>
<b>Resultados Esperados</b>	Devolución de un archivo JSON, que contenga los campos: <b>Ejecución, Mensaje y Resultado</b> , donde <b>ejecución</b> debe ser falso, el <b>mensaje</b> debe indicar que la fecha de inicio debe ser menor que la fecha de fin y resultado debe ser nulo.
<b>Post condiciones</b>	Ninguna
<b>Observaciones:</b> Ninguna	

**Tabla 143:** Caso de Prueba Post\_Incidencia\_Semanal

<b>Caso de Prueba N° 50</b>			
<b>Prioridad</b>	Alta	<b>Estado</b>	1
<b>Objeto a analizar:</b> Ruta “/api/incidencia/semanas”			
<b>Objetivo:</b> Verificar el correcto funcionamiento de la petición POST a la ruta “/api/incidencia/semanas”.			
<b>Descripción:</b> Se realizará una petición POST a la ruta “/api/incidencia/semanas” de la REST API, con el fin de verificar que exista una respuesta y que los datos presentes en la misma estén correctos.			

<b>Pre condiciones</b>	<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>
<b>Entradas</b>	<ol style="list-style-type: none"> <li>1. <b>Mes:</b> Especifica el mes del cual se desea obtener la consulta.</li> <li>2. <b>Fecha de Inicio:</b> Fecha Inicial del periodo donde se consultan datos.</li> <li>3. <b>Fecha de Fin:</b> Fecha Final del periodo donde se consultan los datos.</li> </ol>
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> <li>2. Crear y Configurar una nueva petición.</li> <li>3. Generar el archivo JSON con los campos necesarios para él envió.</li> <li>4. Ejecutar la consulta.</li> </ol>
<b>Resultados Esperados</b>	<p>Devolución de un archivo JSON con los campos: <b>Ejecución, Mensaje y Resultado</b>, donde este último debe contener un subcampo: <b>Incidencia</b>, que contendrá los promedios semanales de dicha medida.</p>
<b>Post condiciones</b>	<p>Si existe algún error al procesar la consulta, el archivo JSON debe</p>

	contener el elemento <b>Ejecución</b> en Falso y un <b>Mensaje</b> que indique que se debe contactar al administrador, conjuntamente con la creación de un nuevo registro en el archivo “ <b>error.log</b> ”.
<b>Observaciones:</b> Ninguna	

**Tabla 144:** Caso de Prueba Post\_Incidencia\_Semanal\_1

<b>Caso de Prueba N° 51</b>			
<b>Prioridad</b>	Media	<b>Estado</b>	1
<b>Objeto a analizar:</b> Ruta “/api/incidencia/semanas”			
<b>Objetivo:</b> Verificar la validación de los campos de la petición POST a la ruta “/api/incidencia/semanas”.			
<b>Descripción:</b> Se realizará una petición POST a la ruta “/api/incidencia/semanas” de la REST API, con el fin de verificar que exista una respuesta y la correcta validación de los diferentes campos enviados en la consulta.			
<b>Pre condiciones</b>		<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>	
<b>Entradas</b>		<ol style="list-style-type: none"> <li>1. <b>Mes:</b> Tiene que ser una cadena de caracteres de una longitud máxima de 100.</li> </ol>	

	<p>2. <b>Fecha de Inicio:</b> Tiene que ser una cadena de caracteres con el siguiente formato: “año-mes-día”. Ej: 2021-26-9.</p> <p>3. <b>Fecha de Fin:</b> Tiene que ser una cadena de caracteres con el siguiente formato: “año-mes-día”. Ej: 2021-26-9.</p>
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> <li>2. Crear y Configurar una nueva petición.</li> <li>3. Generar el archivo JSON con los campos necesarios para él envió. Dichos campos se llenarán con tipos de datos inadecuados.</li> <li>4. Ejecutar la consulta.</li> </ol>
<b>Resultados Esperados</b>	Devolución de un archivo JSON, que contenga los campos: <b>Ejecución, Mensaje y Resultado</b> , donde <b>ejecución</b> debe ser falso, <b>mensaje</b> contener información sobre el error en el tipo de los datos y resultado debe ser nulo.
<b>Post condiciones</b>	Ninguna
<b>Observaciones:</b>	
Ninguna	

**Tabla 145:** Caso de Prueba Post\_Incidencia\_Semanal\_2

**Caso de Prueba N° 52**

<b>Prioridad</b>	Media	<b>Estado</b>	1
<b>Objeto a analizar:</b> Ruta “/api/incidencia/semanas”			
<b>Objetivo:</b> Verificar la validación de los campos obligatorios de la petición POST a la ruta “/api/incidencia/semanas”.			
<b>Descripción:</b> Se realizará una petición POST a la ruta “/api/incidencia/semanas” de la REST API, con el fin de verificar que exista una respuesta y la validación de campos obligatorios de la consulta.			
<b>Pre condiciones</b>		<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>	
<b>Entradas</b>		<ol style="list-style-type: none"> <li>1. <b>Mes:</b> Campo Obligatorio.</li> <li>2. <b>Fecha de Inicio:</b> Campo Obligatorio.</li> <li>3. <b>Fecha de Fin:</b> Campo Obligatorio.</li> </ol>	
<b>Pasos</b>		<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> <li>2. Crear y Configurar una nueva petición.</li> <li>3. Enviar un archivo JSON vacío.</li> <li>4. Ejecutar la consulta.</li> </ol>	
<b>Resultados Esperados</b>		Devolución de un archivo JSON, que contenga los campos: <b>Ejecución, Mensaje y Resultado</b> , donde	

	<b>ejecución</b> debe ser falso, <b>mensaje</b> contener información sobre los campos faltantes dentro de la consulta y resultado debe ser nulo.
<b>Post condiciones</b>	Ninguna
<b>Observaciones:</b> Ninguna	

**Tabla 146:** Caso de Prueba Post\_Incidencia\_Semanal\_3

<b>Caso de Prueba N° 53</b>			
<b>Prioridad</b>	Media	<b>Estado</b>	1
<b>Objeto a analizar:</b> Ruta “/api/incidencia/semanas”			
<b>Objetivo:</b> Verificar la correcta validación de que las fechas pertenezcan al mismo año en la petición POST a la ruta “/api/estaciones/semanas”.			
<b>Descripción:</b> Se realizará una petición POST a la ruta “/api/incidencia/semanas” de la REST API, con el fin de verificar que exista una respuesta y que se validen correctamente que la <b>fecha de inicio</b> y <b>fecha de fin</b> pertenezcan al mismo año.			
<b>Pre condiciones</b>		<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>	
<b>Entradas</b>		<ol style="list-style-type: none"> <li>1. Mes.</li> <li>2. Fecha de Inicio.</li> <li>3. Fecha de Fin.</li> </ol>	
<b>Pasos</b>		<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> </ol>	

	<ol style="list-style-type: none"> <li>2. Crear y Configurar una nueva petición.</li> <li>3. Generar el archivo JSON con los campos necesarios para él envió. Se llenarán las fechas con valores de distintos años.</li> <li>4. Ejecutar la consulta.</li> </ol>
<b>Resultados Esperados</b>	Devolución de un archivo JSON, que contenga los campos: <b>Ejecución, Mensaje y Resultado</b> , donde <b>ejecución</b> debe ser falso, el <b>mensaje</b> debe indicar que las fechas deben pertenecer al mismo año y resultado debe ser nulo.
<b>Post condiciones</b>	Ninguna
<b>Observaciones:</b> Ninguna	

**Tabla 147:** Caso de Prueba Post\_Incidencia\_Semanal\_4

<b>Caso de Prueba N° 54</b>			
<b>Prioridad</b>	Media	<b>Estado</b>	1
<b>Objeto a analizar:</b> Ruta “/api/incidencia/semanas”			
<b>Objetivo:</b> Verificar la correcta validación de que las fechas sean coherentes en la petición POST a la ruta “/api/incidencia/semanas”.			
<b>Descripción:</b> Se realizará una petición POST a la ruta “/api/incidencia/semanas” de la REST API, con el fin de verificar que exista una respuesta y que se validen correctamente que la <b>fecha de inicio</b> sea menor que la <b>fecha de fin</b> .			

<b>Pre condiciones</b>	<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>
<b>Entradas</b>	<ol style="list-style-type: none"> <li>1. Mes.</li> <li>2. Fecha de Inicio.</li> <li>3. Fecha de Fin.</li> </ol>
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> <li>2. Crear y Configurar una nueva petición.</li> <li>3. Generar el archivo JSON con los campos necesarios para él envío. Se llenará el campo de la fecha de inicio con un valor mayor a la fecha de fin.</li> <li>4. Ejecutar la consulta.</li> </ol>
<b>Resultados Esperados</b>	Devolución de un archivo JSON, que contenga los campos: <b>Ejecución, Mensaje y Resultado</b> , donde <b>ejecución</b> debe ser falso, el <b>mensaje</b> debe indicar que la fecha de inicio debe ser menor que la fecha de fin y resultado debe ser nulo.
<b>Post condiciones</b>	Ninguna
<b>Observaciones:</b>	
Ninguna	

**Tabla 148:** Caso de Prueba Post\_Incidencia\_Diario

<b>Caso de Prueba N° 55</b>			
<b>Prioridad</b>	Alta	<b>Estado</b>	1
<b>Objeto a analizar:</b> Ruta “/api/incidencia/dias”			
<b>Objetivo:</b> Verificar el correcto funcionamiento de la petición POST a la ruta “/api/incidencia/dias”.			
<b>Descripción:</b> Se realizará una petición POST a la ruta “/api/incidencia/dias” de la REST API, con el fin de verificar que exista una respuesta y que los datos presentes en la misma estén correctos.			
<b>Pre condiciones</b>		<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>	
<b>Entradas</b>		<ol style="list-style-type: none"> <li>1. <b>Mes:</b> Especifica el mes del cual se desea obtener la consulta.</li> <li>2. <b>Semana:</b> Especifica el mes del cual se desea obtener la consulta.</li> <li>3. <b>Fecha de Inicio:</b> Fecha Inicial del periodo donde se consultan datos.</li> <li>4. <b>Fecha de Fin:</b> Fecha Final del periodo donde se consultan los datos.</li> </ol>	

<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> <li>2. Crear y Configurar una nueva petición.</li> <li>3. Generar el archivo JSON con los campos necesarios para él envió.</li> <li>4. Ejecutar la consulta.</li> </ol>
<b>Resultados Esperados</b>	Devolución de un archivo JSON con los campos: <b>Ejecución</b> , <b>Mensaje</b> y <b>Resultado</b> , donde este último debe contener un subcampo: <b>Incidencia</b> , que contendrá los promedios diarios de dicha medida.
<b>Post condiciones</b>	Si existe algún error al procesar la consulta, el archivo JSON debe contener el elemento <b>Ejecución</b> en Falso y un <b>Mensaje</b> que indique que se debe contactar al administrador, conjuntamente con la creación de un nuevo registro en el archivo <b>“error.log”</b> .
<b>Observaciones:</b> Ninguna	

**Tabla 149:** Caso de Prueba Post\_Incidencia\_Diario\_1

<b>Caso de Prueba N° 56</b>			
<b>Prioridad</b>	Media	<b>Estado</b>	1
<b>Objeto a analizar:</b> Ruta “/api/incidencia/dias”			
<b>Objetivo:</b>			

<p>Verificar la validación de los campos de la petición POST a la ruta “/api/incidencia/dias”.</p>	
<p><b>Descripción:</b>          Se realizará una petición POST a la ruta “/api/incidencia/dias” de la REST API, con el fin de verificar que exista una respuesta y la correcta validación de los diferentes campos enviados en la consulta.</p>	
<p><b>Pre condiciones</b></p>	<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>
<p><b>Entradas</b></p>	<ol style="list-style-type: none"> <li>1. <b>Mes:</b> Tiene que ser una cadena de caracteres de una longitud máxima de 100.</li> <li>2. <b>Semana:</b> Tiene que ser una cadena de caracteres de una longitud máxima de 100.</li> <li>3. <b>Fecha de Inicio:</b> Tiene que ser una cadena de caracteres con el siguiente formato: “año-mes-día”. Ej: 2021-26-9.</li> <li>4. <b>Fecha de Fin:</b> Tiene que ser una cadena de caracteres con el siguiente formato: “año-mes-día”. Ej: 2021-26-9.</li> </ol>
<p><b>Pasos</b></p>	<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> <li>2. Crear y Configurar una nueva petición.</li> </ol>

	<p>3. Generar el archivo JSON con los campos necesarios para él envío. Dichos campos se llenarán con tipos de datos inadecuados.</p> <p>4. Ejecutar la consulta.</p>
<b>Resultados Esperados</b>	Devolución de un archivo JSON, que contenga los campos: <b>Ejecución, Mensaje y Resultado</b> , donde <b>ejecución</b> debe ser falso, <b>mensaje</b> contener información sobre el error en el tipo de los datos y resultado debe ser nulo.
<b>Post condiciones</b>	Ninguna
<b>Observaciones:</b> Ninguna	

**Tabla 150:** Caso de Prueba Post\_Incidencia\_Diario\_2

<b>Caso de Prueba N° 57</b>			
<b>Prioridad</b>	Media	<b>Estado</b>	1
<b>Objeto a analizar:</b> Ruta “/api/incidencia/dias”			
<b>Objetivo:</b> Verificar la validación de los campos obligatorios de la petición POST a la ruta “/api/incidencia/dias”.			
<b>Descripción:</b> Se realizará una petición POST a la ruta “/api/incidencia/dias” de la REST API, con el fin de verificar que exista una respuesta y la validación de campos obligatorios de la consulta.			
<b>Pre condiciones</b>		1. Base de datos y procedimientos almacenados en funcionamiento.	

	<ol style="list-style-type: none"> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>
<b>Entradas</b>	<ol style="list-style-type: none"> <li>1. <b>Mes:</b> Campo Obligatorio.</li> <li>2. <b>Semana:</b> Campo Obligatorio.</li> <li>3. <b>Fecha de Inicio:</b> Campo Obligatorio.</li> <li>4. <b>Fecha de Fin:</b> Campo Obligatorio.</li> </ol>
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> <li>2. Crear y Configurar una nueva petición.</li> <li>3. Enviar un archivo JSON vacío.</li> <li>4. Ejecutar la consulta.</li> </ol>
<b>Resultados Esperados</b>	Devolución de un archivo JSON, que contenga los campos: <b>Ejecución, Mensaje y Resultado</b> , donde <b>ejecución</b> debe ser falso, <b>mensaje</b> contener información sobre los campos faltantes dentro de la consulta y resultado debe ser nulo.
<b>Post condiciones</b>	Ninguna
<b>Observaciones:</b> Ninguna	

**Tabla 151:** Caso de Prueba Post\_Incidencia\_Diario\_3

<b>Caso de Prueba N° 58</b>			
<b>Prioridad</b>	Media	<b>Estado</b>	1
<b>Objeto a analizar:</b>			

Ruta “/api/incidencia/dias”	
<b>Objetivo:</b> Verificar la correcta validación de que las fechas pertenezcan al mismo año en la petición POST a la ruta “/api/incidencia/diario”.	
<b>Descripción:</b> Se realizará una petición POST a la ruta “/api/incidencia/diario” de la REST API, con el fin de verificar que exista una respuesta y que se validen correctamente que la <b>fecha de inicio</b> y <b>fecha de fin</b> pertenezcan al mismo año.	
<b>Pre condiciones</b>	<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>
<b>Entradas</b>	<ol style="list-style-type: none"> <li>1. Mes.</li> <li>2. Semana.</li> <li>3. Fecha de Inicio.</li> <li>4. Fecha de Fin.</li> </ol>
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> <li>2. Crear y Configurar una nueva petición.</li> <li>3. Generar el archivo JSON con los campos necesarios para él envió. Se llenarán las fechas con valores de distintos años.</li> <li>4. Ejecutar la consulta.</li> </ol>
<b>Resultados Esperados</b>	Devolución de un archivo JSON, que contenga los campos: <b>Ejecución, Mensaje y Resultado</b> , donde <b>ejecución</b> debe ser falso, el <b>mensaje</b>

	debe indicar que las fechas deben pertenecer al mismo año y resultado debe ser nulo.
<b>Post condiciones</b>	Ninguna
<b>Observaciones:</b> Ninguna	

**Tabla 152:** Caso de Prueba Post\_Incidencia\_Diario\_4

<b>Caso de Prueba N° 59</b>			
<b>Prioridad</b>	Media	<b>Estado</b>	1
<b>Objeto a analizar:</b> Ruta “/api/incidencia/diario”			
<b>Objetivo:</b> Verificar la correcta validación de que las fechas sean coherentes en la petición POST a la ruta “/api/incidencia/diario”.			
<b>Descripción:</b> Se realizará una petición POST a la ruta “/api/incidencia/diario” de la REST API, con el fin de verificar que exista una respuesta y que se validen correctamente que la <b>fecha de inicio</b> sea menor que la <b>fecha de fin</b> .			
<b>Pre condiciones</b>		<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>	
<b>Entradas</b>		<ol style="list-style-type: none"> <li>1. Mes.</li> <li>2. Semana.</li> <li>3. Fecha de Inicio.</li> <li>4. Fecha de Fin.</li> </ol>	
<b>Pasos</b>		<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> </ol>	

	<ol style="list-style-type: none"> <li>2. Crear y Configurar una nueva petición.</li> <li>3. Generar el archivo JSON con los campos necesarios para él envío. Se llenará el campo de la fecha de inicio con un valor mayor a la fecha de fin.</li> <li>4. Ejecutar la consulta.</li> </ol>
<b>Resultados Esperados</b>	Devolución de un archivo JSON, que contenga los campos: <b>Ejecución, Mensaje y Resultado</b> , donde <b>ejecución</b> debe ser falso, el <b>mensaje</b> debe indicar que la fecha de inicio debe ser menor que la fecha de fin y resultado debe ser nulo.
<b>Post condiciones</b>	Ninguna
<b>Observaciones:</b> Ninguna	

## Severidad

**Tabla 153:** Caso de Prueba Get\_Severidad

<b>Caso de Prueba N° 60</b>			
<b>Prioridad</b>	Alta	<b>Estado</b>	1
<b>Objeto a analizar:</b> Ruta “/api/severidad”			
<b>Objetivo:</b> Verificar el correcto funcionamiento de la petición GET a la ruta “/api/severidad”.			
<b>Descripción:</b> Se realizará una petición GET a la ruta “/api/severidad” de la REST API, con el fin de verificar que exista una respuesta y que los datos presentes en la misma estén correctos.			

<b>Pre condiciones</b>	<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>
<b>Entradas</b>	Ninguna
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> <li>2. Crear y Configurar una nueva petición.</li> <li>3. Ejecutar la consulta.</li> </ol>
<b>Resultados Esperados</b>	Archivo JSON con tres elementos: <b>Ejecución</b> , <b>Mensaje</b> y <b>Resultado</b> , este último debe contener los años que contengan datos relacionados a la severidad, con sus respectivas fechas mínimas y máximas.
<b>Post condiciones</b>	Si existe algún error al procesar la consulta, el archivo JSON debe contener el elemento <b>Ejecución</b> en Falso y un <b>Mensaje</b> que indique que se debe contactar al administrador, conjuntamente con la creación de un nuevo registro en el archivo " <b>error.log</b> ".
<b>Observaciones:</b>	Ninguna

**Tabla 154:** Caso de Prueba Post\_Severidad\_Mensual

<b>Caso de Prueba N° 61</b>			
<b>Prioridad</b>	Alta	<b>Estado</b>	1
<b>Objeto a analizar:</b> Ruta “/api/severidad/meses”			
<b>Objetivo:</b> Verificar el correcto funcionamiento de la petición POST a la ruta “/api/severidad/meses”.			

<b>Descripción:</b> Se realizará una petición POST a la ruta “/api/severidad/meses” de la REST API, con el fin de verificar que exista una respuesta y que los datos presentes en la misma estén correctos.	
<b>Pre condiciones</b>	<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>
<b>Entradas</b>	<ol style="list-style-type: none"> <li>1. <b>Fecha de Inicio:</b> Fecha Inicial del periodo donde se consultan datos.</li> <li>2. <b>Fecha de Fin:</b> Fecha Final del periodo donde se consultan los datos.</li> </ol>
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> <li>2. Crear y Configurar una nueva petición.</li> <li>3. Generar el archivo JSON con los campos necesarios para él envío.</li> <li>4. Ejecutar la consulta.</li> </ol>
<b>Resultados Esperados</b>	Devolución de un archivo JSON con los campos: <b>Ejecución</b> , <b>Mensaje</b> y <b>Resultado</b> , donde este último debe contener como subcampos los diferentes años contenidos en el periodo, los cuales son una lista de los promedios mensuales.
<b>Post condiciones</b>	Si existe algún error al procesar la consulta, el archivo JSON debe contener el elemento <b>Ejecución</b> en Falso y un <b>Mensaje</b> que indique que se debe contactar al administrador, conjuntamente con la creación de un nuevo registro en el archivo “ <b>error.log</b> ”.
<b>Observaciones:</b> Ninguna	

**Tabla 155:** Caso de Prueba Post\_Severidad\_Mensual\_1

<b>Caso de Prueba N° 62</b>			
<b>Prioridad</b>	Media	<b>Estado</b>	1
<b>Objeto a analizar:</b> Ruta “/api/severidad/meses”			
<b>Objetivo:</b> Verificar la validación de los campos de la petición POST a la ruta “/api/severidad/meses”.			
<b>Descripción:</b> Se realizará una petición POST a la ruta “/api/severidad/meses” de la REST API, con el fin de verificar que exista una respuesta y la correcta validación de los diferentes campos enviados en la consulta.			
<b>Pre condiciones</b>	<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>		
<b>Entradas</b>	<ol style="list-style-type: none"> <li>1. <b>Fecha de Inicio:</b> Tiene que ser una cadena de caracteres con el siguiente formato: “año-mes-día”. Ej: 2021-26-9.</li> <li>2. <b>Fecha de Fin:</b> Tiene que ser una cadena de caracteres con el siguiente formato: “año-mes-día”. Ej: 2021-26-9.</li> </ol>		
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> <li>2. Crear y Configurar una nueva petición.</li> <li>3. Generar el archivo JSON con los campos necesarios para él envío. Dichos campos se llenarán con tipos de datos inadecuados.</li> <li>4. Ejecutar la consulta.</li> </ol>		
<b>Resultados Esperados</b>	Devolución de un archivo JSON, que contenga los campos: <b>Ejecución, Mensaje y Resultado,</b>		

	donde <b>ejecución</b> debe ser falso, <b>mensaje</b> contener información sobre el error en el tipo de los datos y resultado debe ser nulo.
<b>Post condiciones</b>	Ninguna
<b>Observaciones:</b> Ninguna	

**Tabla 156:** Caso de Prueba Post\_Severidad\_Mensual\_2

<b>Caso de Prueba N° 63</b>			
<b>Prioridad</b>	Media	<b>Estado</b>	1
<b>Objeto a analizar:</b> Ruta “/api/severidad/meses”			
<b>Objetivo:</b> Verificar la validación de los campos obligatorios de la petición POST a la ruta “/api/severidad/meses”.			
<b>Descripción:</b> Se realizará una petición POST a la ruta “/api/severidad/meses” de la REST API, con el fin de verificar que exista una respuesta y la validación de campos obligatorios de la consulta.			
<b>Pre condiciones</b>	<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>		
<b>Entradas</b>	<ol style="list-style-type: none"> <li>1. <b>Fecha de Inicio:</b> Campo Obligatorio.</li> <li>2. <b>Fecha de Fin:</b> Campo Obligatorio.</li> </ol>		
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> <li>2. Crear y Configurar una nueva petición.</li> <li>3. Enviar un archivo JSON vacío.</li> <li>4. Ejecutar la consulta.</li> </ol>		
<b>Resultados Esperados</b>	Devolución de un archivo JSON, que contenga los campos: <b>Ejecución</b> , <b>Mensaje</b> y <b>Resultado</b> , donde <b>ejecución</b> debe ser falso, <b>mensaje</b> contener		

	información sobre los campos faltantes dentro de la consulta y resultado debe ser nulo.
<b>Post condiciones</b>	Ninguna
<b>Observaciones:</b> Ninguna	

**Tabla 157:** Caso de Prueba Post\_Severidad\_Mensual\_3

<b>Caso de Prueba N° 64</b>			
<b>Prioridad</b>	Media	<b>Estado</b>	1
<b>Objeto a analizar:</b> Ruta “/api/severidad/meses”			
<b>Objetivo:</b> Verificar la correcta validación de que las fechas sean coherentes en la petición POST a la ruta “/api/severidad/meses”.			
<b>Descripción:</b> Se realizará una petición POST a la ruta “/api/severidad/meses” de la REST API, con el fin de verificar que exista una respuesta y que se validen correctamente que la <b>fecha de inicio</b> sea menor que la <b>fecha de fin</b> .			
<b>Pre condiciones</b>	<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>		
<b>Entradas</b>	<ol style="list-style-type: none"> <li>1. Fecha de Inicio.</li> <li>2. Fecha de Fin.</li> </ol>		
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> <li>2. Crear y Configurar una nueva petición.</li> <li>3. Generar el archivo JSON con los campos necesarios para él envío. Se llenará el campo de la fecha de inicio con un valor mayor a la fecha de fin.</li> <li>4. Ejecutar la consulta.</li> </ol>		
<b>Resultados Esperados</b>	Devolución de un archivo JSON, que contenga los campos: <b>Ejecución, Mensaje y Resultado</b> , donde		

	<b>ejecución</b> debe ser falso, el <b>mensaje</b> debe indicar que la fecha de inicio debe ser menor que la fecha de fin y resultado debe ser nulo.
<b>Post condiciones</b>	Ninguna
<b>Observaciones:</b> Ninguna	

**Tabla 158:** Caso de Prueba Post\_Severidad\_Anual

<b>Caso de Prueba N° 65</b>			
<b>Prioridad</b>	Alta	<b>Estado</b>	1
<b>Objeto a analizar:</b> Ruta “/api/severidad/anios”			
<b>Objetivo:</b> Verificar el correcto funcionamiento de la petición POST a la ruta “/api/severidad/anios”.			
<b>Descripción:</b> Se realizará una petición POST a la ruta “/api/severidad/anios” de la REST API, con el fin de verificar que exista una respuesta y que los datos presentes en la misma estén correctos.			
<b>Pre condiciones</b>	<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>		
<b>Entradas</b>	<ol style="list-style-type: none"> <li>1. <b>Fecha de Inicio:</b> Fecha Inicial del periodo donde se consultan datos.</li> <li>2. <b>Fecha de Fin:</b> Fecha Final del periodo donde se consultan los datos.</li> </ol>		
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> <li>2. Crear y Configurar una nueva petición.</li> <li>3. Generar el archivo JSON con los campos necesarios para él envío.</li> <li>4. Ejecutar la consulta.</li> </ol>		

<b>Resultados Esperados</b>	Devolución de un archivo JSON con los campos: <b>Ejecución, Mensaje y Resultado</b> , donde este último debe contener un subcampo: <b>Severidad</b> , el cual contiene los promedios anuales de dicha medida.
<b>Post condiciones</b>	Si existe algún error al procesar la consulta, el archivo JSON debe contener el elemento <b>Ejecución</b> en Falso y un <b>Mensaje</b> que indique que se debe contactar al administrador, conjuntamente con la creación de un nuevo registro en el archivo <b>“error.log”</b> .
<b>Observaciones:</b> Ninguna	

**Tabla 159:** Caso de Prueba Post\_Severidad\_Anual\_1

<b>Caso de Prueba N° 66</b>			
<b>Prioridad</b>	Media	<b>Estado</b>	1
<b>Objeto a analizar:</b> Ruta “/api/severidad/anios”			
<b>Objetivo:</b> Verificar la validación de los campos de la petición POST a la ruta “/api/severidad/anios”.			
<b>Descripción:</b> Se realizará una petición POST a la ruta “/api/severidad/anios” de la REST API, con el fin de verificar que exista una respuesta y la correcta validación de los diferentes campos enviados en la consulta.			
<b>Pre condiciones</b>	<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>		
<b>Entradas</b>	<ol style="list-style-type: none"> <li>1. <b>Fecha de Inicio:</b> Tiene que ser una cadena de caracteres con el siguiente formato: “año-mes-día”. Ej: 2021-26-9.</li> </ol>		

	2. <b>Fecha de Fin:</b> Tiene que ser una cadena de caracteres con el siguiente formato: “año-mes-día”. Ej: 2021-26-9.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> <li>2. Crear y Configurar una nueva petición.</li> <li>3. Generar el archivo JSON con los campos necesarios para él envío. Dichos campos se llenarán con tipos de datos inadecuados.</li> <li>4. Ejecutar la consulta.</li> </ol>
<b>Resultados Esperados</b>	Devolución de un archivo JSON, que contenga los campos: <b>Ejecución, Mensaje y Resultado</b> , donde <b>ejecución</b> debe ser falso, <b>mensaje</b> contener información sobre el error en el tipo de los datos y resultado debe ser nulo.
<b>Post condiciones</b>	Ninguna
<b>Observaciones:</b> Ninguna	

**Tabla 160:** Caso de Prueba Post\_Severidad\_Anuar\_2

<b>Caso de Prueba N° 67</b>			
<b>Prioridad</b>	Media	<b>Estado</b>	1
<b>Objeto a analizar:</b> Ruta “/api/severidad/anios”			
<b>Objetivo:</b> Verificar la validación de los campos obligatorios de la petición POST a la ruta “/api/severidad/anios”.			
<b>Descripción:</b> Se realizará una petición POST a la ruta “/api/severidad/anios” de la REST API, con el fin de verificar que exista una respuesta y la validación de campos obligatorios de la consulta.			
<b>Pre condiciones</b>	1. Base de datos y procedimientos almacenados en funcionamiento.		

	<ol style="list-style-type: none"> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>
<b>Entradas</b>	<ol style="list-style-type: none"> <li>1. <b>Fecha de Inicio:</b> Campo Obligatorio.</li> <li>2. <b>Fecha de Fin:</b> Campo Obligatorio.</li> </ol>
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> <li>2. Crear y Configurar una nueva petición.</li> <li>3. Enviar un archivo JSON vacío.</li> <li>4. Ejecutar la consulta.</li> </ol>
<b>Resultados Esperados</b>	Devolución de un archivo JSON, que contenga los campos: <b>Ejecución, Mensaje y Resultado</b> , donde <b>ejecución</b> debe ser falso, <b>mensaje</b> contener información sobre los campos faltantes dentro de la consulta y resultado debe ser nulo.
<b>Post condiciones</b>	Ninguna
<b>Observaciones:</b> Ninguna	

**Tabla 161:** Caso de Prueba Post\_Severidad\_Anuar\_3

<b>Caso de Prueba N° 68</b>			
<b>Prioridad</b>	Media	<b>Estado</b>	1
<b>Objeto a analizar:</b> Ruta “/api/severidad/anios”			
<b>Objetivo:</b> Verificar la correcta validación de que las fechas sean coherentes en la petición POST a la ruta “/api/severidad/anios”.			
<b>Descripción:</b> Se realizará una petición POST a la ruta “/api/severidad/anios” de la REST API, con el fin de verificar que exista una respuesta y que se validen correctamente que la <b>fecha de inicio</b> sea menor que la <b>fecha de fin</b> .			
<b>Pre condiciones</b>	<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> </ol>		

	3. Habilitar las consultas desde cualquier destino.
<b>Entradas</b>	1. Fecha de Inicio. 2. Fecha de Fin.
<b>Pasos</b>	1. Ingresar a la Herramienta Postman. 2. Crear y Configurar una nueva petición. 3. Generar el archivo JSON con los campos necesarios para el envío. Se llenará el campo de la fecha de inicio con un valor mayor a la fecha de fin. 4. Ejecutar la consulta.
<b>Resultados Esperados</b>	Devolución de un archivo JSON, que contenga los campos: <b>Ejecución, Mensaje y Resultado</b> , donde <b>ejecución</b> debe ser falso, el <b>mensaje</b> debe indicar que la fecha de inicio debe ser menor que la fecha de fin y resultado debe ser nulo.
<b>Post condiciones</b>	Ninguna
<b>Observaciones:</b> Ninguna	

**Tabla 162:** Caso de Prueba Post\_Severidad\_Fincas

<b>Caso de Prueba N° 69</b>			
<b>Prioridad</b>	Alta	<b>Estado</b>	1
<b>Objeto a analizar:</b> Ruta “/api/severidad/fincas”			
<b>Objetivo:</b> Verificar el correcto funcionamiento de la petición POST a la ruta “/api/severidad/fincas”.			
<b>Descripción:</b> Se realizará una petición POST a la ruta “/api/severidad/fincas” de la REST API, con el fin de verificar que exista una respuesta y que los datos presentes en la misma estén correctos.			
<b>Pre condiciones</b>		1. Base de datos y procedimientos almacenados en funcionamiento.	

	<p>2. Servidor de la REST API en funcionamiento.</p> <p>3. Habilitar las consultas desde cualquier destino.</p>
<b>Entradas</b>	<p>1. <b>Fecha de Inicio:</b> Fecha Inicial del periodo donde se consultan datos.</p> <p>2. <b>Fecha de Fin:</b> Fecha Final del periodo donde se consultan los datos.</p>
<b>Pasos</b>	<p>1. Ingresar a la Herramienta Postman.</p> <p>2. Crear y Configurar una nueva petición.</p> <p>3. Generar el archivo JSON con los campos necesarios para él envío.</p> <p>4. Ejecutar la consulta.</p>
<b>Resultados Esperados</b>	<p>Devolución de un archivo JSON con los campos: <b>Ejecución</b>, <b>Mensaje</b> y <b>Resultado</b>, donde este último debe contener dos subcampos: <b>Severidad</b> y <b>Fincas</b>, donde el primero contiene clasificado en años los promedios mensuales de la medida, mientras que el segundo contiene los datos descriptivos de las fincas presentes en los resultados.</p>
<b>Post condiciones</b>	<p>Si existe algún error al procesar la consulta, el archivo JSON debe contener el elemento <b>Ejecución</b> en Falso y un <b>Mensaje</b> que indique que se debe contactar al administrador, conjuntamente con la creación de un nuevo registro en el archivo “<b>error.log</b>”.</p>

<p><b>Observaciones:</b></p> <p>Ninguna</p>
---

**Tabla 163:** Caso de Prueba Post\_Severidad\_Fincas\_1

<b>Caso de Prueba N° 70</b>	
<b>Prioridad</b>	Media
<b>Estado</b>	1
<b>Objeto a analizar:</b>	
Ruta “/api/severidad/fincas”	
<b>Objetivo:</b>	
Verificar la validación de los campos de la petición POST a la ruta “/api/severidad/fincas”.	
<b>Descripción:</b>	
Se realizará una petición POST a la ruta “/api/severidad/fincas” de la REST API, con el fin de verificar que exista una respuesta y la correcta validación de los diferentes campos enviados en la consulta.	
<b>Pre condiciones</b>	<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>
<b>Entradas</b>	<ol style="list-style-type: none"> <li>1. <b>Fecha de Inicio:</b> Tiene que ser una cadena de caracteres con el siguiente formato: “año-mes-día”. Ej: 2021-26-9.</li> <li>2. <b>Fecha de Fin:</b> Tiene que ser una cadena de caracteres con el siguiente formato: “año-mes-día”. Ej: 2021-26-9.</li> </ol>
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> <li>2. Crear y Configurar una nueva petición.</li> <li>3. Generar el archivo JSON con los campos necesarios para él envío. Dichos campos se llenarán con tipos de datos inadecuados.</li> <li>4. Ejecutar la consulta.</li> </ol>

<b>Resultados Esperados</b>	Devolución de un archivo JSON, que contenga los campos: <b>Ejecución, Mensaje y Resultado</b> , donde <b>ejecución</b> debe ser falso, <b>mensaje</b> contener información sobre el error en el tipo de los datos y resultado debe ser nulo.
<b>Post condiciones</b>	Ninguna
<b>Observaciones:</b> Ninguna	

**Tabla 164:** Caso de Prueba Post\_Severidad\_Fincas\_2

<b>Caso de Prueba N° 71</b>			
<b>Prioridad</b>	Media	<b>Estado</b>	1
<b>Objeto a analizar:</b> Ruta “/api/severidad/fincas”			
<b>Objetivo:</b> Verificar la validación de los campos obligatorios de la petición POST a la ruta “/api/severidad/fincas”.			
<b>Descripción:</b> Se realizará una petición POST a la ruta “/api/severidad/fincas” de la REST API, con el fin de verificar que exista una respuesta y la validación de campos obligatorios de la consulta.			
<b>Pre condiciones</b>	<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>		
<b>Entradas</b>	<ol style="list-style-type: none"> <li>1. <b>Fecha de Inicio:</b> Campo Obligatorio.</li> <li>2. <b>Fecha de Fin:</b> Campo Obligatorio.</li> </ol>		
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> <li>2. Crear y Configurar una nueva petición.</li> <li>3. Enviar un archivo JSON vacío.</li> <li>4. Ejecutar la consulta.</li> </ol>		

<b>Resultados Esperados</b>	Devolución de un archivo JSON, que contenga los campos: <b>Ejecución, Mensaje y Resultado</b> , donde <b>ejecución</b> debe ser falso, <b>mensaje</b> contener información sobre los campos faltantes dentro de la consulta y resultado debe ser nulo.
<b>Post condiciones</b>	Ninguna
<b>Observaciones:</b> Ninguna	

**Tabla 165:** Caso de Prueba Post\_Severidad\_Fincas\_3

<b>Caso de Prueba N° 72</b>			
<b>Prioridad</b>	Media	<b>Estado</b>	1
<b>Objeto a analizar:</b> Ruta “/api/severidad/fincas”			
<b>Objetivo:</b> Verificar la correcta validación de que las fechas sean coherentes en la petición POST a la ruta “/api/severidad/fincas”.			
<b>Descripción:</b> Se realizará una petición POST a la ruta “/api/severidad/fincas” de la REST API, con el fin de verificar que exista una respuesta y que se validen correctamente que la <b>fecha de inicio</b> sea menor que la <b>fecha de fin</b> .			
<b>Pre condiciones</b>	<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>		
<b>Entradas</b>	<ol style="list-style-type: none"> <li>1. Fecha de Inicio.</li> <li>2. Fecha de Fin.</li> </ol>		
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> <li>2. Crear y Configurar una nueva petición.</li> <li>3. Generar el archivo JSON con los campos necesarios para él envió. Se llenará el campo de la fecha de inicio con un valor mayor a la fecha de fin.</li> </ol>		

	4. Ejecutar la consulta.
<b>Resultados Esperados</b>	Devolución de un archivo JSON, que contenga los campos: <b>Ejecución, Mensaje y Resultado</b> , donde <b>ejecución</b> debe ser falso, el <b>mensaje</b> debe indicar que la fecha de inicio debe ser menor que la fecha de fin y resultado debe ser nulo.
<b>Post condiciones</b>	Ninguna
<b>Observaciones:</b> Ninguna	

**Tabla 166:** Caso de Prueba Post\_Severidad\_Plantas

<b>Caso de Prueba N° 73</b>			
<b>Prioridad</b>	Alta	<b>Estado</b>	1
<b>Objeto a analizar:</b> Ruta “/api/severidad/fincas/plantas”			
<b>Objetivo:</b> Verificar el correcto funcionamiento de la petición POST a la ruta “/api/severidad/fincas/plantas”.			
<b>Descripción:</b> Se realizará una petición POST a la ruta “/api/severidad/fincas/plantas” de la REST API, con el fin de verificar que exista una respuesta y que los datos presentes en la misma estén correctos.			
<b>Pre condiciones</b>	<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>		
<b>Entradas</b>	<ol style="list-style-type: none"> <li>1. <b>Año:</b> Especifica el año del cual se desea obtener la consulta.</li> <li>2. <b>ID Finca:</b> Identificador de la finca de la cual se desea obtener la consulta.</li> <li>3. <b>Fecha de Inicio:</b> Fecha Inicial del periodo donde se consultan datos.</li> </ol>		

	4. <b>Fecha de Fin:</b> Fecha Final del periodo donde se consultan los datos.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> <li>2. Crear y Configurar una nueva petición.</li> <li>3. Generar el archivo JSON con los campos necesarios para él envío.</li> <li>4. Ejecutar la consulta.</li> </ol>
<b>Resultados Esperados</b>	Devolución de un archivo JSON con los campos: <b>Ejecución, Mensaje y Resultado</b> , donde este último debe contener dos subcampos: <b>Severidad y Plantas</b> , donde el primero contendrá los promedios por planta y el segundo contendrá datos descriptivos de las plantas.
<b>Post condiciones</b>	Si existe algún error al procesar la consulta, el archivo JSON debe contener el elemento <b>Ejecución</b> en Falso y un <b>Mensaje</b> que indique que se debe contactar al administrador, conjuntamente con la creación de un nuevo registro en el archivo “ <b>error.log</b> ”.
<b>Observaciones:</b> Ninguna	

**Tabla 167:** Caso de Prueba Post\_Severidad\_Plantas\_1

<b>Caso de Prueba N° 74</b>			
<b>Prioridad</b>	Media	<b>Estado</b>	1
<b>Objeto a analizar:</b> Ruta “/api/severidad/fincas/plantas”			
<b>Objetivo:</b> Verificar la validación de los campos de la petición POST a la ruta “/api/severidad/fincas/plantas”.			
<b>Descripción:</b> Se realizará una petición POST a la ruta “/api/severidad/fincas/plantas” de la REST API, con el fin de verificar que exista una respuesta y la correcta validación de los diferentes campos enviados en la consulta.			

<b>Pre condiciones</b>	<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>
<b>Entradas</b>	<ol style="list-style-type: none"> <li>1. <b>Año:</b> Es de tipo entero.</li> <li>2. <b>ID Finca:</b> Es de tipo entero.</li> <li>3. <b>Fecha de Inicio:</b> Tiene que ser una cadena de caracteres con el siguiente formato: “año-mes-día”. Ej: 2021-26-9.</li> <li>4. <b>Fecha de Fin:</b> Tiene que ser una cadena de caracteres con el siguiente formato: “año-mes-día”. Ej: 2021-26-9.</li> </ol>
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> <li>2. Crear y Configurar una nueva petición.</li> <li>3. Generar el archivo JSON con los campos necesarios para él envío. Dichos campos se llenarán con tipos de datos inadecuados.</li> <li>4. Ejecutar la consulta.</li> </ol>
<b>Resultados Esperados</b>	<p>Devolución de un archivo JSON, que contenga los campos: <b>Ejecución, Mensaje y Resultado</b>, donde <b>ejecución</b> debe ser falso, <b>mensaje</b> contener información sobre el error en el tipo de los datos y resultado debe ser nulo.</p>

<b>Post condiciones</b>	Ninguna
<b>Observaciones:</b> Ninguna	

**Tabla 168:** Caso de Prueba Post\_Severidad\_Plantas\_2

<b>Caso de Prueba N° 75</b>	
<b>Prioridad</b>	Media
<b>Estado</b>	1
<b>Objeto a analizar:</b> Ruta “/api/severidad/fincas/plantas”	
<b>Objetivo:</b> Verificar la validación de los campos obligatorios de la petición POST a la ruta “/api/severidad/fincas/plantas”.	
<b>Descripción:</b> Se realizará una petición POST a la ruta “/api/severidad/fincas/plantas” de la REST API, con el fin de verificar que exista una respuesta y la validación de campos obligatorios de la consulta.	
<b>Pre condiciones</b>	<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>
<b>Entradas</b>	<ol style="list-style-type: none"> <li>1. <b>Mes:</b> Campo Obligatorio.</li> <li>2. <b>ID Tiempo:</b> Campo Obligatorio.</li> <li>3. <b>Fecha de Inicio:</b> Campo Obligatorio.</li> <li>4. <b>Fecha de Fin:</b> Campo Obligatorio.</li> </ol>
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> <li>2. Crear y Configurar una nueva petición.</li> <li>3. Enviar un archivo JSON vacío.</li> <li>4. Ejecutar la consulta.</li> </ol>
<b>Resultados Esperados</b>	Devolución de un archivo JSON, que contenga los campos: <b>Ejecución, Mensaje y Resultado</b> , donde <b>ejecución</b> debe ser falso, <b>mensaje</b> contener información

	sobre los campos faltantes dentro de la consulta y resultado debe ser nulo.
<b>Post condiciones</b>	Ninguna
<b>Observaciones:</b>	Ninguna

**Tabla 169:** Caso de Prueba Post\_Severidad\_Plantas\_3

<b>Caso de Prueba N° 76</b>			
<b>Prioridad</b>	Media	<b>Estado</b>	1
<b>Objeto a analizar:</b> Ruta “/api/severidad/fincas/plantas”			
<b>Objetivo:</b> Verificar la correcta validación de que las fechas sean coherentes en la petición POST a la ruta “/api/severidad/fincas/plantas”.			
<b>Descripción:</b> Se realizará una petición POST a la ruta “/api/severidad/fincas/plantas” de la REST API, con el fin de verificar que exista una respuesta y que se validen correctamente que la <b>fecha de inicio</b> sea menor que la <b>fecha de fin</b> .			
<b>Pre condiciones</b>	<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>		
<b>Entradas</b>	<ol style="list-style-type: none"> <li>1. Año.</li> <li>2. ID Finca.</li> <li>3. Fecha de Inicio.</li> <li>4. Fecha de Fin.</li> </ol>		
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> <li>2. Crear y Configurar una nueva petición.</li> <li>3. Generar el archivo JSON con los campos necesarios para él envió. Se llenará el campo de la fecha de inicio con un valor mayor a la fecha de fin.</li> </ol>		

	4. Ejecutar la consulta.
<b>Resultados Esperados</b>	Devolución de un archivo JSON, que contenga los campos: <b>Ejecución, Mensaje y Resultado</b> , donde <b>ejecución</b> debe ser falso, el <b>mensaje</b> debe indicar que la fecha de inicio debe ser menor que la fecha de fin y resultado debe ser nulo.
<b>Post condiciones</b>	Ninguna
<b>Observaciones:</b> Ninguna	

**Tabla 170:** Caso de Prueba Post\_Severidad\_Frutos

<b>Caso de Prueba N° 77</b>			
<b>Prioridad</b>	Alta	<b>Estado</b>	1
<b>Objeto a analizar:</b> Ruta “/api/severidad/fincas/plantas/frutos”			
<b>Objetivo:</b> Verificar el correcto funcionamiento de la petición POST a la ruta “/api/severidad/fincas/plantas/frutos”.			
<b>Descripción:</b> Se realizará una petición POST a la ruta “/api/severidad/fincas/plantas/frutos” de la REST API, con el fin de verificar que exista una respuesta y que los datos presentes en la misma estén correctos.			
<b>Pre condiciones</b>	<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>		
<b>Entradas</b>	1. <b>Año:</b> Especifica el año del cual se desea obtener la consulta.		

	<ol style="list-style-type: none"> <li>2. <b>ID Finca:</b> identificador de la finca de la cual se desea obtener la consulta.</li> <li>3. <b>ID Planta:</b> identificador de la planta de la cual se desea obtener la consulta.</li> <li>4. <b>Fecha de Inicio:</b> Fecha Inicial del periodo donde se consultan datos.</li> <li>5. <b>Fecha de Fin:</b> Fecha Final del periodo donde se consultan los datos.</li> </ol>
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> <li>2. Crear y Configurar una nueva petición.</li> <li>3. Generar el archivo JSON con los campos necesarios para él envió.</li> <li>4. Ejecutar la consulta.</li> </ol>
<b>Resultados Esperados</b>	<p>Devolución de un archivo JSON con los campos: <b>Ejecución</b>, <b>Mensaje</b> y <b>Resultado</b>, donde este último debe contener tres subcampos: <b>Severidad</b>, <b>Severidad Promedio</b> y <b>Severidad Total</b>, donde el primero contendrá los promedios por fruto de la severidad, el segundo el promedio de la severidad en todos los frutos y el ultimo la suma de todas las severidades.</p>
<b>Post condiciones</b>	<p>Si existe algún error al procesar la consulta, el archivo JSON debe contener el elemento <b>Ejecución</b> en Falso y un <b>Mensaje</b> que indique que se debe contactar al administrador, conjuntamente con la creación de un nuevo registro en el archivo <b>“error.log”</b>.</p>

<p><b>Observaciones:</b></p> <p>Ninguna</p>
---

**Tabla 171:** Caso de Prueba Post\_Severidad\_Frutos\_1

<b>Caso de Prueba N° 78</b>	
<b>Prioridad</b>	Media <b>Estado</b> 1
<b>Objeto a analizar:</b> Ruta “/api/severidad/fincas/plantas/frutos”	
<b>Objetivo:</b> Verificar la validación de los campos de la petición POST a la ruta “/api/severidad/fincas/plantas/frutos”.	
<b>Descripción:</b> Se realizará una petición POST a la ruta “/api/severidad/fincas/plantas/frutos” de la REST API, con el fin de verificar que exista una respuesta y la correcta validación de los diferentes campos enviados en la consulta.	
<b>Pre condiciones</b>	<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>
<b>Entradas</b>	<ol style="list-style-type: none"> <li>1. <b>Año:</b> Es de tipo entero.</li> <li>2. <b>ID Finca:</b> Es de tipo entero.</li> <li>3. <b>ID Planta:</b> Es de tipo entero.</li> <li>4. <b>Fecha de Inicio:</b> Tiene que ser una cadena de caracteres con el siguiente formato: “año-mes-día”. Ej: 2021-26-9.</li> <li>5. <b>Fecha de Fin:</b> Tiene que ser una cadena de caracteres con el siguiente formato: “año-mes-día”. Ej: 2021-26-9.</li> </ol>
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> </ol>

	<p>2. Crear y Configurar una nueva petición.</p> <p>3. Generar el archivo JSON con los campos necesarios para él envío. Dichos campos se llenarán con tipos de datos inadecuados.</p> <p>4. Ejecutar la consulta.</p>
<b>Resultados Esperados</b>	<p>Devolución de un archivo JSON, que contenga los campos: <b>Ejecución, Mensaje y Resultado</b>, donde <b>ejecución</b> debe ser falso, <b>mensaje</b> contener información sobre el error en el tipo de los datos y resultado debe ser nulo.</p>
<b>Post condiciones</b>	Ninguna
<b>Observaciones:</b>	
Ninguna	

**Tabla 172:** Caso de Prueba Post\_Severidad\_Frutos\_2

<b>Caso de Prueba N° 79</b>			
<b>Prioridad</b>	Media	<b>Estado</b>	1
<b>Objeto a analizar:</b>			
Ruta “/api/severidad/fincas/plantas/frutos”			
<b>Objetivo:</b>			
Verificar la validación de los campos obligatorios de la petición POST a la ruta “/api/severidad/fincas/plantas/frutos”.			
<b>Descripción:</b>			
Se realizará una petición POST a la ruta “/api/severidad/fincas/plantas/frutos” de la REST API, con el fin de verificar que exista una respuesta y la validación de campos obligatorios de la consulta.			
<b>Pre condiciones</b>		1. Base de datos y procedimientos almacenados en funcionamiento.	

	<p>2. Servidor de la REST API en funcionamiento.</p> <p>3. Habilitar las consultas desde cualquier destino.</p>
<b>Entradas</b>	<p>1. <b>Año:</b> Campo Obligatorio.</p> <p>2. <b>ID Finca:</b> Campo Obligatorio.</p> <p>3. <b>ID Planta:</b> Campo Obligatorio.</p> <p>4. <b>Fecha de Inicio:</b> Campo Obligatorio.</p> <p>5. <b>Fecha de Fin:</b> Campo Obligatorio.</p>
<b>Pasos</b>	<p>1. Ingresar a la Herramienta Postman.</p> <p>2. Crear y Configurar una nueva petición.</p> <p>3. Enviar un archivo JSON vacío.</p> <p>4. Ejecutar la consulta.</p>
<b>Resultados Esperados</b>	<p>Devolución de un archivo JSON, que contenga los campos: <b>Ejecución, Mensaje y Resultado</b>, donde <b>ejecución</b> debe ser falso, <b>mensaje</b> contener información sobre los campos faltantes dentro de la consulta y <b>resultado</b> debe ser nulo.</p>
<b>Post condiciones</b>	Ninguna
<b>Observaciones:</b>	
Ninguna	

**Tabla 173:** Caso de Prueba Post\_Severidad\_Plantas\_4

<b>Caso de Prueba N° 80</b>			
<b>Prioridad</b>	Media	<b>Estado</b>	1
<b>Objeto a analizar:</b>			
Ruta “/api/severidad/fincas/plantas/frutos”			
<b>Objetivo:</b>			

<p>Verificar la correcta validación de que las fechas sean coherentes en la petición POST a la ruta “/api/severidad/fincas/plantas/frutos”.</p>	
<p><b>Descripción:</b>  Se realizará una petición POST a la ruta “/api/severidad/fincas/plantas/frutos” de la REST API, con el fin de verificar que exista una respuesta y que se validen correctamente que la <b>fecha de inicio</b> sea menor que la <b>fecha de fin</b>.</p>	
<p><b>Pre condiciones</b></p>	<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>
<p><b>Entradas</b></p>	<ol style="list-style-type: none"> <li>1. Año.</li> <li>2. ID Finca.</li> <li>3. ID Planta.</li> <li>4. Semana.</li> <li>5. Fecha de Inicio.</li> <li>6. Fecha de Fin.</li> </ol>
<p><b>Pasos</b></p>	<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> <li>2. Crear y Configurar una nueva petición.</li> <li>3. Generar el archivo JSON con los campos necesarios para él envío. Se llenará el campo de la fecha de inicio con un valor mayor a la fecha de fin.</li> <li>4. Ejecutar la consulta.</li> </ol>
<p><b>Resultados Esperados</b></p>	<p>Devolución de un archivo JSON, que contenga los campos: <b>Ejecución, Mensaje y Resultado</b>, donde <b>ejecución</b> debe ser falso, el <b>mensaje</b> debe indicar que la fecha de inicio debe</p>

	ser menor que la fecha de fin y resultado debe ser nulo.
<b>Post condiciones</b>	Ninguna
<b>Observaciones:</b> Ninguna	

**Tabla 174:** Caso de Prueba Post\_Severidad\_Estudio

<b>Caso de Prueba N° 81</b>			
<b>Prioridad</b>	Alta	<b>Estado</b>	1
<b>Objeto a analizar:</b> Ruta “/api/severidad/estudio”			
<b>Objetivo:</b> Verificar el correcto funcionamiento de la petición POST a la ruta “/api/severidad/estudio”.			
<b>Descripción:</b> Se realizará una petición POST a la ruta “/api/severidad/estudio” de la REST API, con el fin de verificar que exista una respuesta y que los datos presentes en la misma estén correctos.			
<b>Pre condiciones</b>	<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>		
<b>Entradas</b>	<ol style="list-style-type: none"> <li>1. <b>Fecha de Inicio:</b> Fecha Inicial del periodo donde se consultan datos.</li> <li>2. <b>Fecha de Fin:</b> Fecha Final del periodo donde se consultan los datos.</li> </ol>		
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> <li>2. Crear y Configurar una nueva petición.</li> </ol>		

	<p>3. Generar el archivo JSON con los campos necesarios para él envío.</p> <p>4. Ejecutar la consulta.</p>
<b>Resultados Esperados</b>	<p>Devolución de un archivo JSON con los campos: <b>Ejecución</b>, <b>Mensaje</b> y <b>Resultado</b>, donde este último debe contener dos subcampos: <b>Severidad</b> y <b>Estudios</b>, donde el primero contiene clasificado en años los promedios mensuales de la medida, mientras que el segundo contiene los datos descriptivos de los estudios presentes en los resultados.</p>
<b>Post condiciones</b>	<p>Si existe algún error al procesar la consulta, el archivo JSON debe contener el elemento <b>Ejecución</b> en Falso y un <b>Mensaje</b> que indique que se debe contactar al administrador, conjuntamente con la creación de un nuevo registro en el archivo <b>“error.log”</b>.</p>
<b>Observaciones:</b>	
Ninguna	

**Tabla 175:** Caso de Prueba Post\_Severidad\_Estudio\_1

<b>Caso de Prueba N° 82</b>			
<b>Prioridad</b>	Media	<b>Estado</b>	1
<b>Objeto a analizar:</b>			
Ruta “/api/severidad/estudio”			
<b>Objetivo:</b>			
Verificar la validación de los campos de la petición POST a la ruta “/api/severidad/estudio”.			
<b>Descripción:</b>			

Se realizará una petición POST a la ruta “/api/severidad/estudio” de la REST API, con el fin de verificar que exista una respuesta y la correcta validación de los diferentes campos enviados en la consulta.	
<b>Pre condiciones</b>	<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>
<b>Entradas</b>	<ol style="list-style-type: none"> <li>1. <b>Fecha de Inicio:</b> Tiene que ser una cadena de caracteres con el siguiente formato: “año-mes-día”. Ej: 2021-26-9.</li> <li>2. <b>Fecha de Fin:</b> Tiene que ser una cadena de caracteres con el siguiente formato: “año-mes-día”. Ej: 2021-26-9.</li> </ol>
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> <li>2. Crear y Configurar una nueva petición.</li> <li>3. Generar el archivo JSON con los campos necesarios para él envió. Dichos campos se llenarán con tipos de datos inadecuados.</li> <li>4. Ejecutar la consulta.</li> </ol>
<b>Resultados Esperados</b>	Devolución de un archivo JSON, que contenga los campos: <b>Ejecución, Mensaje y Resultado</b> , donde <b>ejecución</b> debe ser falso, <b>mensaje</b> contener información sobre el error en el tipo de los datos y resultado debe ser nulo.
<b>Post condiciones</b>	Ninguna
<b>Observaciones:</b> Ninguna	

**Tabla 176:** Caso de Prueba Post\_Severidad\_Estudio\_2

<b>Caso de Prueba N° 83</b>			
<b>Prioridad</b>	Media	<b>Estado</b>	1
<b>Objeto a analizar:</b> Ruta “/api/severidad/estudio”			
<b>Objetivo:</b> Verificar la validación de los campos obligatorios de la petición POST a la ruta “/api/severidad/estudio”.			
<b>Descripción:</b> Se realizará una petición POST a la ruta “/api/severidad/estudio” de la REST API, con el fin de verificar que exista una respuesta y la validación de campos obligatorios de la consulta.			
<b>Pre condiciones</b>	<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>		
<b>Entradas</b>	<ol style="list-style-type: none"> <li>1. <b>Fecha de Inicio:</b> Campo Obligatorio.</li> <li>2. <b>Fecha de Fin:</b> Campo Obligatorio.</li> </ol>		
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> <li>2. Crear y Configurar una nueva petición.</li> <li>3. Enviar un archivo JSON vacío.</li> <li>4. Ejecutar la consulta.</li> </ol>		
<b>Resultados Esperados</b>	Devolución de un archivo JSON, que contenga los campos: <b>Ejecución, Mensaje y Resultado</b> , donde <b>ejecución</b> debe ser falso, <b>mensaje</b> contener información sobre los campos faltantes		

	dentro de la consulta y resultado debe ser nulo.
<b>Post condiciones</b>	Ninguna
<b>Observaciones:</b> Ninguna	

**Tabla 177:** Caso de Prueba Post\_Severidad\_Estudio\_3

<b>Caso de Prueba N° 84</b>			
<b>Prioridad</b>	Media	<b>Estado</b>	1
<b>Objeto a analizar:</b> Ruta “/api/severidad/estudio”			
<b>Objetivo:</b> Verificar la correcta validación de que las fechas sean coherentes en la petición POST a la ruta “/api/severidad/estudio”.			
<b>Descripción:</b> Se realizará una petición POST a la ruta “/api/severidad/estudio” de la REST API, con el fin de verificar que exista una respuesta y que se validen correctamente que la <b>fecha de inicio</b> sea menor que la <b>fecha de fin</b> .			
<b>Pre condiciones</b>		<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>	
<b>Entradas</b>		<ol style="list-style-type: none"> <li>1. Fecha de Inicio.</li> <li>2. Fecha de Fin.</li> </ol>	
<b>Pasos</b>		<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> <li>2. Crear y Configurar una nueva petición.</li> <li>3. Generar el archivo JSON con los campos necesarios para él envío. Se</li> </ol>	

	llenará el campo de la fecha de inicio con un valor mayor a la fecha de fin. 4. Ejecutar la consulta.
<b>Resultados Esperados</b>	Devolución de un archivo JSON, que contenga los campos: <b>Ejecución, Mensaje y Resultado</b> , donde <b>ejecución</b> debe ser falso, el <b>mensaje</b> debe indicar que la fecha de inicio debe ser menor que la fecha de fin y resultado debe ser nulo.
<b>Post condiciones</b>	Ninguna
<b>Observaciones:</b> Ninguna	

**Tabla 178:** Caso de Prueba Post\_Severidad\_Monitoreo

<b>Caso de Prueba N° 85</b>			
<b>Prioridad</b>	Alta	<b>Estado</b>	1
<b>Objeto a analizar:</b> Ruta “/api/severidad/estudio/monitoreo”			
<b>Objetivo:</b> Verificar el correcto funcionamiento de la petición POST a la ruta “/api/severidad/estudio/monitoreo”.			
<b>Descripción:</b> Se realizará una petición POST a la ruta “/api/severidad/estudio/monitoreo” de la REST API, con el fin de verificar que exista una respuesta y que los datos presentes en la misma estén correctos.			
<b>Pre condiciones</b>	<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>		
<b>Entradas</b>	<ol style="list-style-type: none"> <li>1. <b>Año:</b> Especifica el año del cual se desea obtener la consulta.</li> </ol>		

	<p>2. <b>ID Estudio:</b> Identificador del estudio del cual se desea obtener la consulta.</p> <p>3. <b>Fecha de Inicio:</b> Fecha Inicial del periodo donde se consultan datos.</p> <p>4. <b>Fecha de Fin:</b> Fecha Final del periodo donde se consultan los datos.</p>
<b>Pasos</b>	<p>1. Ingresar a la Herramienta Postman.</p> <p>2. Crear y Configurar una nueva petición.</p> <p>3. Generar el archivo JSON con los campos necesarios para él envío.</p> <p>4. Ejecutar la consulta.</p>
<b>Resultados Esperados</b>	<p>Devolución de un archivo JSON con los campos: <b>Ejecución, Mensaje y Resultado</b>, donde este último debe contener dos subcampos: <b>Severidad y Monitoreos</b>, donde el primero contendrá los promedios por monitoreo y el segundo contendrá datos descriptivos de los monitoreos.</p>
<b>Post condiciones</b>	<p>Si existe algún error al procesar la consulta, el archivo JSON debe contener el elemento <b>Ejecución</b> en Falso y un <b>Mensaje</b> que indique que se debe contactar al administrador, conjuntamente con la creación de un nuevo registro en el archivo <b>“error.log”</b>.</p>
<b>Observaciones:</b>	
Ninguna	

**Tabla 179:** Caso de Prueba Post\_Severidad\_Monitoreo\_1

<b>Caso de Prueba N° 86</b>			
<b>Prioridad</b>	Media	<b>Estado</b>	1
<b>Objeto a analizar:</b>			
Ruta “/api/severidad/estudio/monitoreo”			
<b>Objetivo:</b>			

<p>Verificar la validación de los campos de la petición POST a la ruta “/api/severidad/estudio/monitoreo”.</p>	
<p><b>Descripción:</b>  Se realizará una petición POST a la ruta “/api/severidad/estudio/monitoreo” de la REST API, con el fin de verificar que exista una respuesta y la correcta validación de los diferentes campos enviados en la consulta.</p>	
<p><b>Pre condiciones</b></p>	<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>
<p><b>Entradas</b></p>	<ol style="list-style-type: none"> <li>1. <b>Año:</b> Es de tipo entero.</li> <li>2. <b>ID Estudio:</b> Es de tipo entero.</li> <li>3. <b>Fecha de Inicio:</b> Tiene que ser una cadena de caracteres con el siguiente formato: “año-mes-día”. Ej: 2021-26-9.</li> <li>4. <b>Fecha de Fin:</b> Tiene que ser una cadena de caracteres con el siguiente formato: “año-mes-día”. Ej: 2021-26-9.</li> </ol>
<p><b>Pasos</b></p>	<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> <li>2. Crear y Configurar una nueva petición.</li> <li>3. Generar el archivo JSON con los campos necesarios para él envío. Dichos campos se llenarán con tipos de datos inadecuados.</li> <li>4. Ejecutar la consulta.</li> </ol>
<p><b>Resultados Esperados</b></p>	<p>Devolución de un archivo JSON, que contenga los campos: <b>Ejecución, Mensaje y Resultado</b>, donde <b>ejecución</b> debe ser falso, <b>mensaje</b></p>

	contener información sobre el error en el tipo de los datos y resultado debe ser nulo.
<b>Post condiciones</b>	Ninguna
<b>Observaciones:</b> Ninguna	

**Tabla 180:** Caso de Prueba Post\_Severidad\_Monitoreo\_2

<b>Caso de Prueba N° 87</b>			
<b>Prioridad</b>	Media	<b>Estado</b>	1
<b>Objeto a analizar:</b> Ruta “/api/severidad/estudio/monitoreo”			
<b>Objetivo:</b> Verificar la validación de los campos obligatorios de la petición POST a la ruta “/api/severidad/estudio/monitoreo”.			
<b>Descripción:</b> Se realizará una petición POST a la ruta “/api/severidad/estudio/monitoreo” de la REST API, con el fin de verificar que exista una respuesta y la validación de campos obligatorios de la consulta.			
<b>Pre condiciones</b>	<ol style="list-style-type: none"> <li>1. Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2. Servidor de la REST API en funcionamiento.</li> <li>3. Habilitar las consultas desde cualquier destino.</li> </ol>		
<b>Entradas</b>	<ol style="list-style-type: none"> <li>1. <b>Mes:</b> Campo Obligatorio.</li> <li>2. <b>ID Estudio:</b> Campo Obligatorio.</li> <li>3. <b>Fecha de Inicio:</b> Campo Obligatorio.</li> <li>4. <b>Fecha de Fin:</b> Campo Obligatorio.</li> </ol>		
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Ingresar a la Herramienta Postman.</li> <li>2. Crear y Configurar una nueva petición.</li> <li>3. Enviar un archivo JSON vacío.</li> <li>4. Ejecutar la consulta.</li> </ol>		

<b>Resultados Esperados</b>	Devolución de un archivo JSON, que contenga los campos: <b>Ejecución, Mensaje y Resultado</b> , donde <b>ejecución</b> debe ser falso, <b>mensaje</b> contener información sobre los campos faltantes dentro de la consulta y resultado debe ser nulo.
<b>Post condiciones</b>	Ninguna
<b>Observaciones:</b> Ninguna	

**Tabla 181:** Caso de Prueba Post\_Severidad\_Monitoreo\_3

<b>Caso de Prueba N° 88</b>			
<b>Prioridad</b>	Media	<b>Estado</b>	1
<b>Objeto a analizar:</b> Ruta “/api/severidad/estudio/monitoreo”			
<b>Objetivo:</b> Verificar la correcta validación de que las fechas sean coherentes en la petición POST a la ruta “/api/severidad/estudio/monitoreo”.			
<b>Descripción:</b> Se realizará una petición POST a la ruta “/api/severidad/estudio/monitoreo” de la REST API, con el fin de verificar que exista una respuesta y que se validen correctamente que la <b>fecha de inicio</b> sea menor que la <b>fecha de fin</b> .			
<b>Pre condiciones</b>	<ol style="list-style-type: none"> <li>1.Base de datos y procedimientos almacenados en funcionamiento.</li> <li>2.Servidor de la REST API en funcionamiento.</li> <li>3.Habilitar las consultas desde cualquier destino.</li> </ol>		
<b>Entradas</b>	<ol style="list-style-type: none"> <li>1.Año.</li> <li>2.ID Estudio.</li> <li>3.Fecha de Inicio.</li> <li>4.Fecha de Fin.</li> </ol>		
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1.Ingresar a la Herramienta Postman.</li> <li>2.Crear y Configurar una nueva petición.</li> </ol>		

	3. Generar el archivo JSON con los campos necesarios para él envío. Se llenará el campo de la fecha de inicio con un valor mayor a la fecha de fin. 4. Ejecutar la consulta.
<b>Resultados Esperados</b>	Devolución de un archivo JSON, que contenga los campos: <b>Ejecución, Mensaje y Resultado</b> , donde <b>ejecución</b> debe ser falso, el <b>mensaje</b> debe indicar que la fecha de inicio debe ser menor que la fecha de fin y resultado debe ser nulo.
<b>Post condiciones</b>	Ninguna
<b>Observaciones:</b> Ninguna	

### 3.7 Evaluación de los Resultados

#### ETL

**Tabla 182:** Evaluación de los resultados ETL.

<b>Caso Prueba</b>	<b>Estado</b>
<b>STAGING</b>	
CP 1	1
CP 2	1
CP 3	1
CP 4	1
CP 5	1
CP 6	1
CP 7	1
CP 8	1
CP 9	1
CP 10	1
CP 11	1
CP 12	

<b>DIMENSIONES</b>	
CP 13	1
CP 14	1
CP 15	1
CP 16	1
CP 17	1
<b>HECHOS</b>	
CP 18	1
CP 19	1
<b>SCRIPT</b>	
CP 20	1
CP 21	1
CP 22	1
CP 23	1
<b>Total</b>	<b>23</b>

## Front End

**Tabla 183:** Evaluación de los resultados Front end.

<b>Caso Prueba</b>	<b>Estado</b>
<b>GENERAL</b>	
CP 24	1
CP 25	1
CP 26	1
CP 27	1
CP 28	1
CP 29	1
CP 30	1
CP 31	1
CP 32	1
CP 33	1
CP 34	1

CP 35	1
CP 36	1
CP 37	1
CP 38	1
CP 39	1
<b>ESTACIONES</b>	
CP 40	1
CP 41	1
CP 42	1
CP 43	1
CP 44	1
CP 45	1
CP 46	1
CP 47	1
CP 48	1
CP 49	1
CP 50	1
CP 51	1
CP 52	1
CP 53	1
CP 54	1
CP 55	1
CP 56	1
CP 57	1
CP 58	1
CP 59	1
CP 60	1
CP 61	1
CP 62	1
CP 63	1
CP 64	1

CP 65	1
CP 66	1
CP 67	1
CP 68	1
CP 69	1
CP 70	1
CP 71	1
CP 72	1
CP 73	1
CP 74	1
CP 75	1
CP 76	1
CP 77	1
CP 78	1
CP 79	1
CP 80	1
CP 81	1
CP 82	1
CP 83	1
CP 84	1
CP 85	1
CP 86	1
CP 87	1
CP 88	1
CP 89	1
CP 90	1
CP 91	1
CP 92	1
CP 93	1
CP 94	1
CP 95	1

CP 96	1
CP 97	1
CP 98	1
CP 99	1
CP 100	1
CP 101	1
CP 102	1
CP 103	1
CP 104	1
CP 105	1
CP 106	1
CP 107	1
CP 108	1
CP 109	1
CP 110	1
CP 111	1
CP 112	1
CP 113	1
CP 114	1
CP 115	1
CP 116	1
CP 117	1
CP 118	1
CP 119	1
CP 120	1
CP 121	1
CP 122	1
CP 123	1
CP 124	1
CP 125	1
CP 126	1

CP 127	1
<b>SEVERIDAD</b>	
CP 128	1
CP 129	1
CP 130	1
CP 131	1
CP 132	1
CP 133	1
CP 134	1
CP 135	1
CP 136	1
CP 137	1
CP 138	1
CP 139	1
CP 140	1
CP 141	1
CP 142	1
CP 143	1
<b>Total</b>	120

## Back End

**Tabla 184:** Evaluación de los resultados Back End.

<b>Caso Prueba</b>	<b>Estado</b>
<b>ESTACIONES</b>	
CP 144	1
CP 145	1
CP 146	1
CP 147	1
CP 148	1
CP 149	1
CP 150	1

CP 151	1
CP 152	1
CP 153	1
CP 154	1
CP 155	1
CP 156	1
CP 157	1
CP 158	1
CP 159	1
CP 160	1
CP 161	1
CP 162	1
CP 163	1
CP 164	1
CP 165	1
CP 166	1
CP 167	1
CP 168	1
CP 169	1
CP 170	1
CP 171	1
CP 172	1
<b>INCIDENCIA</b>	
CP 173	1
CP 174	1
CP 175	1
CP 176	1
CP 177	1
CP 178	1
CP 179	1
CP 180	1

CP 181	1
CP 182	1
CP 183	1
CP 184	1
CP 185	1
CP 186	1
CP 187	1
CP 188	1
<b>SEVERIDAD</b>	
CP 189	1
CP 190	1
CP 191	1
CP 192	1
CP 193	1
CP 194	1
CP 195	1
CP 196	1
CP 197	1
CP 198	1
CP 199	1
CP 200	1
CP 201	1
CP 202	1
CP 203	1
CP 204	1
CP 205	1
CP 206	1
CP 207	1
CP 208	1
CP 209	1
CP 210	1

CP 211	1
CP 212	1
CP 213	1
CP 214	1
CP 215	1
CP 216	1
CP 217	1
<b>Total</b>	73

### Resultado general

**Tabla 185:** Evaluación de resultados general.

Descripción	Casos Prueba Exitosos	Casos de Prueba Diseñados	Porcentaje
ETL	23	23	100%
Front End	120	120	100%
Back End	73	73	100%
<b>TOTAL</b>			100%

### 3.8 Fase de Transición

#### Despliegue

El Despliegue del Sistema se divide en dos secciones, la primera enfocada a instalar la herramienta Pentaho Data Integration y segunda a desplegar en el servidor de producción el Sitio Web.

#### Instalación de Pentaho Data Integration

Pentaho Data Integration es un software portable que no necesita instalación, pero tiene como requerimiento que el sistema cuente con el JDK y el JRE instalados, ambos tienen que ser de la misma versión y ser compatibles. En este caso tomaremos en cuenta la instalación para Windows, ya que el ETL desarrollo está configurado de base para funcionar sin ningún cambio en dicho sistema operativo.

Para descargar e instalar los componentes de JAVA en Windows, se puede utilizar el siguiente enlace:

CLIC AQUÍ PARA ACCEDER AL RESPOSITORIO

El enlace nos llevara al Sitio Oficial de Oracle para descargar los componentes de JAVA, se recomienda descargar la última versión, debido a que la versión de Pentaho Data Integration es compatible con JAVA 11 o superior.

Una vez que se haya descargado e instalado los componentes de JAVA, procedemos a descargar Pentaho Data Integration 9.3, a través del siguiente enlace:

CLIC AQUÍ PARA ACCEDER AL RESPOSITORIO

El enlace descargara automáticamente un zip con el software.

Una vez terminada la descarga, descomprimos el archivo e ingresamos a la carpeta “data-integration” que da como resultado; dentro de la carpeta se debe ejecutar el archivo por lotes de Windows denominado “Spoon.bat”. Si no existe ningún error, el software se abrirá, caso contrario se nos avisara si es que existe algún problema con los componentes de JAVA instalados.

A partir de este punto se recomienda seguir los pasos detallados en el Manual del ETL.

### **Despliegue del Sitio Web**

Para desplegar el Sitio Web desarrollado se tomará en cuenta las características del servidor propuestas para el proyecto.

El Sitio Web cuenta de dos partes:

- **Front End:** El front end consiste únicamente de archivos HTML, CSS y JAVASCRIPT. Este componente se desplegará en un servidor apache que corre en el host del servidor de aplicaciones.
- **Back End:** El back end es una REST API desarrollada con el framework FLASK de PYTHON. Este componente será desplegado en el servidor GUINICORN dentro de un contenedor de Docker, que estará a la escucha en el puerto 3000. Dicho contenedor se ejecutará el servidor de aplicaciones.
- **Data Base:** En este componente está la base de datos, que es **POSTGRESQL**, y corre en otro host enfocado a base de datos.

El despliegue será a través de Docker, excepto la base de datos, que debe realizarse manualmente.

## Archivos subidos al servidor

Al servidor se subieron todos los ficheros necesarios para el despliegue del proyecto. Los ficheros se encuentran en la ruta “/home/patfa-pxpaguays/PATFADW” del servidor de aplicaciones. Los ficheros se encuentran clasificados en las siguientes carpetas:

- **“backend”**: En esta carpeta se encuentran todos los archivos de codificación del Back End, incluido el DockerFile para su despliegue.
- **“BD”**: En esta carpeta se pueden encontrar dos ficheros “DW.sql” y “Staging.sql”, estos ficheros contienen las tablas, procedimientos almacenados, funciones y vistas para ambas bases de datos.
- **“ETL”**: En esta carpeta se encuentran todos los archivos del ETL.
- **“frontend”**: En esta carpeta se encuentra el proyecto de desarrollo del front end, el cual será compilado y colocado en el servidor de apache.

## Instalación del Docker

Como se dijo antes, el despliegue de la aplicación se realizará a través de Docker, así que como primer paso se realizará su instalación. La instalación que se explicara a continuación únicamente es compatible con Ubuntu 22.04, Sistema Operativo del servidor de aplicaciones. Para instalar Docker se debe utilizar los siguientes comandos en la consola:

```
$ sudo apt-get update

$ sudo apt-get install \
ca-certificates \
curl \
gnupg \
lsb-release

$ sudo mkdir -p /etc/apt/keyrings

$ curl -fsSL
https://download.docker.com/linux/ubuntu/gpg | sudo gpg -
-dearmor -o /etc/apt/keyrings/docker.gpg
```

```
$ echo \  
"deb [arch=$(dpkg --print-architecture) signed-  
by=/etc/apt/keyrings/docker.gpg]  
https://download.docker.com/linux/ubuntu \  
$(lsb_release -cs) stable" | sudo tee  
/etc/apt/sources.list.d/docker.list > /dev/null  
  
$ apt-get update  
  
$ sudo apt-get install docker-ce docker-ce-cli containerd.io  
docker-compose-plugin
```

Una vez ejecutado todos los comandos, se utilizará los siguientes comandos para verificar la versión de Docker y Docker Compose, esto con el fin de validar la correcta instalación.

```
$ sudo docker versión  
  
$ sudo docker compose
```

Una vez que los comandos se ejecuten correctamente, se habrá verificado que se ha instalado correctamente docker en el servidor.

Para mayor detalle sobre el proceso de instalación de docker ingresar al siguiente enlace:

[CLIC AQUÍ PARA ACCEDER AL RESPOSITORIO](#)

**Ficheros**

**de**

**Docker**

Tanto Back End como Font End, tienen un fichero Dockerfile, que poseen las configuraciones para desplegar cada aplicación.

**Front End**

El fichero Dockerfile del front end, cuenta con lo siguiente:

```
FROM node:14.16.1-alpine3.13  
  
WORKDIR /usr/app
```

```

COPY . .
RUN npm install -g @angular/cli@11.2.9
RUN npm install -g @ionic/cli@6.18.1
RUN npm i
RUN ng build --configuration production --base-href
/tablero/
COPY ./server-config ./www

```

Este archivo únicamente tiene como objetivo construir un contenedor que tome como base un contenedor existente de node en su versión “14.16.1”, para compilar el proyecto desarrollado en angular para el front end.

Debido a que todos los archivos del proyecto de angular son necesarios para el despliegue, se agregó un fichero “.dockerignore” con el fin de evitar que sean copiados al contenedor archivos innecesarios.

### **Back End**

El fichero Dockerfile del back end, cuenta con lo siguiente:

```

FROM python:3.10.4-alpine3.16

WORKDIR /usr/app
COPY requirements.txt .
COPY ./src .
RUN /usr/local/bin/python -m pip install --upgrade pip
RUN pip install --no-cache-dir -r requirements.txt
RUN apk add -U tzdata
RUN ln -s /usr/share/zoneinfo/America/Guayaquil
/etc/localtime

EXPOSE 3000

CMD [ "gunicorn", "--workers=4", "--
bind", "0.0.0.0:3000", "app:app" ]

```

Este archivo tiene como objetivo construir un contenedor que tome como base un contenedor existente de Python en su versión “3.10.4”, para desplegar la REST API mediante el servidor de GUNICOR.

Al igual que el front end, se cuenta con un fichero “.dockerignore”.

### **Docker Compose**

El fichero “docker-compose.yml”, contiene la estructura del despliegue, es decir, todos los comandos necesarios para levantar ambos contenedores.

```
version: '3.8'

services:
  restapi:
    container_name: REST_API_PATFA
    restart: always
    build: ./backend
    image: backend_patfa:1.0
    environment:
      - HOST=$HOST
      - PORT=$PORT
      - PG_DATABASE=$PG_DATABASE
      - PG_USER=$PG_USER
      - PG_PASSWORD=$PG_PASSWORD
      - PG_HOST=$PG_HOST
      - PG_PORT=$PG_PORT
    ports:
      - 3000:3000

  angular_app:
    container_name: ANGULAR_APP_PATFA
    build: ./frontend
    image: frontend_patfa:1.0
    depends_on:
```

```
- restapi
volumes:
  - server_apache:/usr/app/www

volumes:
  server_apache:
    driver: local
    driver_opts:
      type: 'none'
      o: 'bind'
      device: '/var/www/html/tablero'

networks:
  default:
    name: patfa_net
```

En el archivo se declaran dos servicios, uno enfocado desplegar el contenedor del Back End y el otro enfocado a compilar el proyecto del Font End.

El contenedor del back end se quedará en ejecución, es decir, levantará el servidor de GUNICORN en el puerto 3000 y lo mantendrá a la escucha.

El contenedor del front end, únicamente compilará el proyecto y se detendrá inmediatamente. Cabe recalcar que se está generando un volumen denominado “server\_apache”, este volumen esta sincronizado con la ruta “/var/www/html/tablero” del host, esto con la finalidad que los ficheros de la compilación del proyecto de angular sean movidos al servidor de apache que se está ejecutando en el host.

Las variables de entorno del servicio “restapi”, se cargan desde otras variables de entorno declaradas en el HOST, no se recomienda colocar el valor de manera directa.

### **Configuración de Apache**

Un último aspecto a configurar es servidor de apache que se encuentra ejecutándose en el HOST, debido a que el front end es una SPA, se debe deshabilitar el enrutador por defecto de apache, para que utilice el enrutador de angular, esta configuración esta realizada en

el fichero “.htaccess” del directorio “server-config” en el front end, para que la configuración de “.htaccess” haga efecto se debe habilitar el modo “rewrite” en el servidor, para ello inicialmente debemos acceder a la ruta: “/etc/apache2/apache2.conf” y modificar la siguiente sección:

```
<Directory /var/www/>
    Options Indexes FollowSymLinks
    AllowOverride All #modificacion
    Require all granted
</Directory>
```

En esta sección únicamente se debe modificar “AllowOverride None” por “AllowOverride All”.

Finalmente queda cargar el módulo de “rewrite” para ello ejecutamos el siguiente comando:

```
$ sudo ln -s /etc/apache2/mods-available/rewrite.load
/etc/apache2/mods-enabled/rewrite.load
```

Finalmente queda reiniciar apache para que los cambios carguen, para ello se ejecutara lo siguiente:

```
$ sudo systemctl restart apache2

$ sudo systemctl status apache2
```

Si todo ha ido correctamente, el “status” de apache será en ejecución.

## **Bases de Datos**

Para la configuración de las bases de datos, se debe realizar lo siguiente:

- Crear las bases de datos “dw\_patfa” y “bd\_staging” en POSTGRESQL.
- Ejecutar el script “DW.sql” en la base de datos “dw\_patfa” y el script “Staging.sql” en la base de datos “bd\_staging”.

### **3.9 Mantenimiento**

No se realizará mantenimiento a la aplicación, ya que se dio soporte a la aplicación durante el proceso de desarrollo, hasta la etapa de despliegue de esta, hasta su funcionamiento en producción.

Se probaron las características incorporadas, de modo que el sistema esté funcionando.

## Referencias

- Astera. (2023). *Conceptos de Data Warehouse: enfoque de Kimball vs. Inmon*. Recuperado de: <https://www.astera.com/es/type/blog/data-warehouse-concepts/>
- Becker, B. (2009). *Kimball Group*. Recuperado de Six Key Decisions for ETL Architectures: <https://www.kimballgroup.com/2009/10/six-key-decisions-for-etl-architectures/>
- Bernabeu, R. D. (2009). Data Warehousing: Investigación y sistematización de conceptos. *Libro Electrónico* <http://www.dataprix.com/es/data-warehousing-hefesto> Cordova, Argentina
- Brizuela, E. I. L., & Blanco, Y. C. (2013). Metodologías para desarrollar Almacén de Datos. *Revista de arquitectura e Ingeniería*, 7(3), 1-12.
- Bustos, S., & Mosquera, V. (2013). *Análisis, diseño e implementación de una solución BUSINESS INTELLIGENCE para la generación de indicadores y control de desempeño, en la empresa OTECEL SA, utilizando la metodología HEFESTO V2.0*. Tesis, Escuela Politécnica del Ejército, Ciencias de la Computación, Sangolquí. Obtenido de <http://repositorio.espe.edu.ec/bitstream/21000/6305/1/T-ESPE-047033.pdf>
- Cáceres, C., & Cevallos, M. (2016). *Business Intelligence para el área de seguridad ciudadana en el distrito de Villa El Salvador mediante la metodología de Ralph Kimball*. Tesis, Universidad Mayor de San Marcos, Ingeniería de Sistemas e Informática, Lima. Obtenido de <https://cybertesis.unmsm.edu.pe/handle/20.500.12672/5810>
- Caicedo, H. R. O., & Tello, A. (2018). Metodología para el diseño de proyectos de Data Warehouse para el sector minero. Caso de Uso: Camilo Ponce Enríquez, Grupo Empresarial Campoverde. *Alternativas*, 19(2), 24-35.
- Castro Jara, B. R. (2022). *Diseñar e implementar la metodología Hefesto para un Data Warehouse y Data Mining en un sistema ERP* (Bachelor's thesis).
- Conesa Caralt, J. (2012). Introducción al business intelligence. *Introducción al business intelligence*, 0-0
- Curto Díaz, J. (2017). Introducción al business intelligence. *Introducción al business intelligence*, 1-315

- Duran, E. (2017). Data mart como parte de una solución de inteligencia de negocios, para el soporte de la toma de decisiones de la gestión académica de la facultad de ciencias en la UNASAM. (*Tesis para optar por el título de Ingeniero de Sistemas e Informática*). Universidad Nacional de Santiago Antunez de Mayolo, Huaraz. Recuperado el 28 de Octubre de 2022, de <http://repositorio.unasam.edu.pe/handle/UNASAM/1926>
- Esteso, M. P. (2021). Qué es una ETL y cómo funciona. Recuperado: <https://geekytheory.com/que-es-una-etl-y-como-funciona/>.
- Gardner, S. R. (1998). Building the data warehouse. *Communications of the ACM*, 41(9), 52-60.
- Guerrero García, M. J., & Rodas Silva, J. (2022). Análisis comparativo de metodologías y herramientas tecnológicas para procesos de Business Intelligence orientado a la toma de decisiones. *Informática y Sistemas: Revista de Tecnologías de la Informática y las Comunicaciones*, 6(1), 38-44. DOI: <https://doi.org/10.33936/isrtic.v6i1.4522>
- Gutiérrez Retamal, G. (2013). Propuesta de implementación de un Data Warehouse para el área de soporte de información, Rabie SA.
- IBM. (2021). *InfoSphere Data Architect*, 9.1.2. Recuperado de: Esquemas de constelación: <https://www.ibm.com/docs/es/ida/9.1.2?topic=schemas-starflake>
- Inmon, W. (2002). *Building the data warehouse* (Tercera ed.). New York: John Wiley & Sons, Inc. Recuperado el 13 de Noviembre de 2022, de <https://fit.hcmute.edu.vn/Resources/Docs/SubDomain/fit/ThayTuan/DataWH/Buiding%20the%20Data%20Warehouse%204%20Edition.pdf>
- Jaramillo, F. (2016). Implementación de un Datawarehouse para la Toma de Decisiones en el área Logística de la Compañía Pronaca. *Proyecto examen complejo previo a la obtención del título de ingeniero en sistemas e informática*. Universidad Nacional Autónoma de los Andes, Santo Domingo. Recuperado de: <https://dspace.uniandes.edu.ec/bitstream/123456789/5254/1/TUAEXCOMSIS02-2-2016.pdf>
- Jarrín, C., & Belalcázar, P. (2014). Diseño e implementación de un data warehouse del Sistema Financiero Ecuatoriano para la integración y consulta de la información.

- Trabajo de graduación previo a la obtención del título de Ingeniero Informático.*  
Universidad Central del Ecuador, Quito, Quito, Ecuador. Recuperado de:  
<http://www.dspace.uce.edu.ec/handle/25000/2829>
- Leonard, I., & Castro, Y. (2013). Metodologías para desarrollar Almacén de Datos. *Revista de Arquitectura e Ingeniería.*, VII(3), 13. doi:ISSN 1990-8830 / RNPS 2125
- Luján-Mora, S., & Trujillo, J. (2005). A data warehouse engineering process. In *Advances in Information Systems: Third International Conference, ADVIS 2004, Izmir, Turkey, October 20-22, 2004. Proceedings 3* (pp. 14-23). Springer Berlin Heidelberg.
- Méndez, G., & Regalado, S. (2009). Modelo de herramienta de inteligencia de negocios aplicada al área de dirección de una importadora. *Tesis de graduación previo a la obtención del título de Ingeniera de Sistemas.* Universidad del Azuay, Cuenca, Cuenca, Ecuador. Recuperado de <https://dspace.uazuay.edu.ec/bitstream/datos/2407/1/07441.pdf>
- Mendoza, A. (2022). *Metodologías de Data Warehouse.* Recuperado de: <https://gravitar.biz/datawarehouse/metodologias-data-warehouse/>
- Kimball, R., & Ross, M. (2002). *The Data Warehouse Toolkit: the complete guide to dimensional modeling* (Segunda ed.). New York: Wiley Computer Publishing. Recuperado de: [https://www.academia.edu/9156395/The\\_Data\\_Warehouse\\_Toolkit\\_The\\_Complete\\_Guide\\_to\\_Dimensional\\_Modeling\\_2nd\\_Ed\\_2002\\_Wiley\\_Kimball\\_and\\_Ross](https://www.academia.edu/9156395/The_Data_Warehouse_Toolkit_The_Complete_Guide_to_Dimensional_Modeling_2nd_Ed_2002_Wiley_Kimball_and_Ross)
- Mbaabu, O. (2021). *Section.* Recuperado: MOLAP vs ROLAP vs HOLAP in Online Analytical Processing (OLAP): <https://www.section.io/engineering-education/molap-vs-rolap-vs-holap/>
- Ochoa, H. (2017). *Sistema para soporte a la toma de decisiones en la comercialización de insumos para el sector minero; caso de uso: grupo empresarial CAMPOVERDE del cantón Camilo Ponce Enríquez de la Provincia del Azuay.* Tesis, Universidad de Cuenca, Facultad de Ingeniería, Cuenca. Obtenido de <http://dspace.ucuenca.edu.ec/bitstream/123456789/28474/1/Trabajo%20de%20Titulaci%C3%B3n.pdf>

- Peñañiel, X. L., Tamayo, H. C., Argudo, W. P., & de la Torre Díaz, A. (2013). ANÁLISIS, DISEÑO, CONSTRUCCIÓN E IMPLEMENTACIÓN DE UN DATA WAREHOUSE PARA TOMA DE DECISIONES Y CONSTRUCCIÓN DE LOS KPI, PARA LA EMPRESA KRONOSCONSULTING CIA LTDA. Recuperado de: <http://repositorio.espe.edu.ec/bitstream/21000/7942/1/AC-SI-ESPE-047750.pdf>
- Romero, F. (2014). *Soporte a la toma de decisión en el Área de Análisis & Mejoramiento Mina, utilizando Data Mart, en la Mina Spence, Bhp Billiton*. Tesis, Universidad del Bio-Bio, Departamento de sistemas de información, Concepción. Obtenido de [http://repobib.ubiobio.cl/jspui/bitstream/123456789/1687/1/Romero\\_Pedrerros\\_Felipe\\_Andres.pdf](http://repobib.ubiobio.cl/jspui/bitstream/123456789/1687/1/Romero_Pedrerros_Felipe_Andres.pdf)
- Rayón, Á. (2015). *Herramientas ETL y su relevancia en la cadena de valor del dato*. Recuperado: <https://blogs.deusto.es/bigdata/herramientas-etl-y-su-relevancia-en-la-cadena-de-valor-del-dato/>.
- Rolón, M. (2021). Metodología para el desarrollo de Datamarts en Universidades Nacionales del Consorcio SIU. Caso de Estudio: Área Graduados de la FCEQyN. *Tesis de Maestría en Tecnologías de la Información*. Universidad Nacional de Misiones, Posadas, Argentina. Recuperado de : [https://www.fceqyn.unam.edu.ar/mti/wp-content/uploads/TFM\\_RolonDaniela.pdf](https://www.fceqyn.unam.edu.ar/mti/wp-content/uploads/TFM_RolonDaniela.pdf)
- Tamayo, M., & Moreno, F. (2006). Análisis del modelo de almacenamiento molap frente al modelo de almacenamiento rolap. *Ingeniería e Investigación*, 26(3), 135-142. Recuperado de: <https://repositorio.unal.edu.co/handle/unal/28782>
- Zambrano Alarcón, J. A. (2011). *Análisis, diseño e implementación de un datamart para el área de mantenimiento y logística de una empresa de transporte público de pasajeros*.

## DE LOS AUTORES

**Pamela Alexandra Buñay Guisñan.**



Correo electrónico: [pbunay@unach.edu.ec](mailto:pbunay@unach.edu.ec). Nacionalidad Ecuatoriana. Máster Universitario en Ingeniería de Software y Sistemas Informáticos. Magíster Interconectividad de Redes. Ingeniera en Sistemas Informáticos. Grupo de investigación ModSim. Semillero: Aprendizaje inmersivo y Servicios TI. Docente Universitaria de la Escuela Superior Politécnica de Chimborazo y la Universidad Nacional de Chimborazo. Participante en proyectos de investigación y vinculación. Experiencia profesional en desarrollo de software. Facilitadora de cursos. Tutora de la comunidad académica D'Soft.Net. Líder del semillero Aprendizaje inmersivo y servicios TI. Directora de proyectos de investigación de pregrado y miembro de tesis de posgrado. Artículos publicados en revistas con factor de impacto, en congresos nacionales e internacionales. Actualmente, Docente de la Universidad Nacional de Chimborazo.

## **Guido Javier Mazón Fierro**



Me llamo Guido Javier Mazón Fierro, y me caracterizo por ser una mente inquieta y apasionada por el conocimiento. Con raíces ecuatorianas, he forjado una sólida trayectoria académica que incluye un Máster Universitario en Ingeniería Matemática y Computación, una Maestría en Desarrollo de la Inteligencia y Educación, y una formación como Ingeniero Mecánico. En la actualidad, desempeño un papel esencial como docente investigador en la Escuela Superior Politécnica de Chimborazo (ESPOCH), específicamente en la Facultad de Administración de Empresas, donde comparto con entusiasmo mi conocimiento y experiencia. Mi participación en el Grupo de Investigación Movilidad ECO es un testimonio de mi compromiso con el avance científico y tecnológico en mi campo. Además, respaldo mi labor investigativa con numerosas publicaciones en revistas científicas regionales y de alto impacto, aportando así a la comunidad académica y científica con descubrimientos y reflexiones significativas.

## René David Carranza Guijarro



Soy René David Carranza Guijarro ([rene.carranza@unach.edu.ec](mailto:rene.carranza@unach.edu.ec)), Ingeniero en Sistemas y Computación de la Universidad Nacional de Chimborazo. Nací en Riobamba, Ecuador el 1 de febrero de 1999.

Participo en el Semillero de Investigación Ciencia de los Datos e Inteligencia Artificial y Aprendizaje Inmersivo y Servicios TI, en el Desarrollo de una Aplicación Web para el registro de los resultados de pruebas rápidas COVID-19, como Coordinador Estudiante y Scrum Master. Forme parte del desarrollo del Proyecto de Investigación Metodología DWEP aplicada a la implementación de un Data Warehouse para la toma de decisiones fitosanitarias agrícolas en la provincia de orellana. Pertenezco al Semillero de Investigación Aprendizaje inmersivo y servicios TI.

Soy un apasionado desarrollador de Software Full Stack con inclinación por el Back End, mejoro constantemente mis habilidades y conocimientos para poderlos aplicar en el desarrollo de nuevos proyectos.

## Danny Joel Bucay Shucad



Soy Danny Joel Bucay Shucad ([danny.bucay@unach.edu.ec](mailto:danny.bucay@unach.edu.ec)), Ingeniero en Sistemas y Computación, de la Universidad Nacional de Chimborazo. Actualmente me encuentro cursando un posgrado. Soy un desarrollador FullStack, con inclinación por el lado del Frontend. En mi trayecto estudiantil tuve la oportunidad de formar parte de proyectos multidisciplinario a través de la participación en los semilleros de investigación tanto el de “Aprendizaje inmersivo y servicios TI” y como el de “Ciencia de los Datos e Inteligencia Artificial”. Por otra parte también, forme para de la investigación denominada “Metodología DWEP aplicada a la implementación de un Data Warehouse para la toma de decisiones fitosanitarias agrícolas en la provincia de Orellana”.





PUERTO MADERO  
EDITORIAL

ISBN 978-631-6557-45-2



9 786316 557452