



UNIVERSIDAD CENTRAL "MARTA ABREU" DE LAS VILLAS
VERITATE SOLA NOBIS IMPONETUR VIRILISTOGA. 1948

**FACULTAD DE MATEMÁTICA, FÍSICA Y
COMPUTACIÓN**
CENTRO DE ESTUDIOS DE INFORMÁTICA

Sistema de gestión para la evaluación pluviométrica de las máquinas de riego de pivote central

Tesis de Diploma

Licenciatura en Ciencia de la Computación

Autor: Lisandra Cardoso Morales

Tutor: MSc. Yoan Pacheco Cárdenas

Santa Clara, Cuba, 2017





Hago constar que el presente trabajo de diploma fue realizado en la Universidad Central "Marta Abreu" de Las Villas como parte de la culminación de estudios de la especialidad de Licenciatura en Ciencias de la Computación, autorizando a que el mismo sea utilizado por la Institución, para los fines que estime conveniente, tanto de forma parcial como total y que además no podrá ser presentado en eventos, ni publicados sin autorización de la Universidad.

Firma del Autor

Los abajo firmantes certificamos que el presente trabajo ha sido realizado según acuerdo de la dirección de nuestro centro y el mismo cumple con los requisitos que debe tener un trabajo de esta envergadura referido a la temática señalada.

Firma del Autor

Firma del Jefe de Departamento
donde se defiende el trabajo

Firma del Responsable de
Información Científico-Técnica

DEDICATORIA

Dedico esta tesis a mi mamá y mi papá por ayudarme y alentarme a ser quien soy hoy en día.

A mis abuelos maternos por sus consejos, amor y cuidados durante toda mi vida.

A mis abuelos paternos por quererme y ayudarme como lo han hecho.

A mi hermana por estar siempre a mi lado y quererme como lo hace.

A mi novio por ser la persona más maravillosa que he conocido y por ser mi apoyo incondicional en todo momento.

A mis suegros y cuñados por acogerme como uno más en su familia y hacerme sentir como en casa.

A Yunet por ser mi amiga incondicional y estar siempre para mí en todo momento, ayudándome en la realización de esta tesis.

A Alfredo por convertirse en mi gran amigo en estos 5 años de carrera.

AGRADECIMIENTOS

A toda mi familia por ayudarme y apoyarme en todas mis decisiones a lo largo de mi vida, y sobre todo por ser la maravillosa familia que somos no podría estar más agradecida en esta vida.

A mi novio por ser mi otra mitad, por todo su apoyo incondicional y por todas las carreras que ha tenido que dar para ayudarme, sin pensarlo dos veces, por resolverme todos mis problemas sin tener que pedirlo y sobre todo por amarme como lo hace.

A mis suegros por ser mi otro apoyo incondicional y tenerme en su familia como una más.

A mis dos cuñados por ayudarme y quererme.

A mi tutor Yoan Pacheco por ayudarme en la realización de esta tesis.

A mis amigos Alfredo, Yunet, Adonis, Carlos E. y Jorgito por estar siempre para mí como mis amigos incondicionales.

Gracias a todos por ayudarme a cerrar este hermoso ciclo de mi vida y hacerme sentir tan querida y amada.

Para todos mi eterno agradecimiento y amor.

PENSAMIENTO

Tu tiempo es limitado, de modo que no lo malgastes viviendo la vida de alguien distinto. No quedes atrapado en el dogma, que es vivir como otros piensan que deberías vivir. No dejes que los ruidos de las opiniones de los demás acallen tu propia voz interior. Y, lo que es más importante, ten el coraje para hacer lo que te dicen tu corazón y tu intuición.

Steve Jobs

RESUMEN

En la empresa agrícola “Valle del Yabú” se realiza el riego de cultivos con las máquinas de pivote central, por lo cual se realiza una evaluación pluviométrica de dichas máquinas para así validar en condiciones de campo los parámetros reales de su funcionamiento, dicha evaluación posibilita validar el funcionamiento hidráulico de la máquina. Actualmente se cuenta con una aplicación desktop que es la encargada de gestionar todo el proceso de evaluación, pero la misma supone diversas dificultades para sus trabajadores por lo que en el presente trabajo se realiza una aplicación web siguiendo los estándares de código abierto que permite gestionar todo el proceso de la evaluación pluviométrica de máquinas de riego de pivote central y además resuelve todas las dificultades presentadas por la aplicación actual.

Se realiza una revisión sistemática de la literatura sobre los mejores marcos de trabajo para el desarrollo web y como resultado de la misma después de comparar los diferentes marcos de trabajo encontrados en cuanto al rápido acceso a la base de datos, la integración con dispositivos móviles, la documentación entre otros parámetros se decide desarrollar el sitio con las herramientas que brinda el marco de trabajo MEAN, que son MongoDB para la gestión de la base de datos, Express que constituye un marco de aplicación web para Node.js, proporcionando una interfaz simple para redirigir URL entrantes, ayudando a Node.js en la creación de aplicaciones, Angular para la programación en el lado del cliente y Node.js que constituye una plataforma de software que permite crear un servidor web propio. Se realiza un diseño e implementación de una base de datos no relacional utilizando MongoDB para la gestión de la información. Se describe el análisis y diseño de los principales componentes de la aplicación y su arquitectura en general. Finalmente se valida el funcionamiento de la aplicación mediante distintos métodos de prueba.

ABSTRACT

In the "Valle del Yabú" agricultural enterprise, irrigation of crops is carried out with the central pivot machines, which is why a pluviometric evaluation of these machines is performed in order to validate in field conditions the actual parameters of its operation, Validate the hydraulic operation of the machine. At the moment there is a desktop application that is in charge of managing the entire evaluation process, but it involves various difficulties for its workers, so that in the present work a web application is performed following the open source standards that allow to manage everything The process of the pluviometric evaluation of central pivot machines and also solves all the difficulties presented by the current application.

A systematic review of the literature on the best frameworks for web development and as a result of this is done after comparing the different frameworks found for fast access to the database, integration with mobile devices, the documentation among other parameters is decided to develop the site following the MEAN philosophy. A design and implementation of a non-relational database using MongoDB for the management of the information is carried out. It describes the analysis and design of the main components of the application and its architecture in general. Finally, the operation of the application is validated using different test methods.

TABLA DE CONTENIDOS

INTRODUCCIÓN.....	1
CAPÍTULO 1. ESTUDIO DEL MARCO TEÓRICO DE LAS TECNOLOGÍAS, LENGUAJES, Y HERRAMIENTAS PARA EL DESARROLLO DE APLICACIONES WEB.....	5
1.1 Revisión Sistemática de la literatura: Diseño y ejecución.....	5
1.1.1 Necesidad de la revisión sistemática de la literatura	5
1.1.2 Protocolo de investigación	6
1.1.3 Ejecución de la revisión sistemática de la literatura.....	8
1.2 Resultados de la Revisión Sistemática de la Literatura	9
1.2.1 RQ1. ¿Cuáles son los principales marcos de trabajo para el desarrollo de aplicaciones web?.....	9
1.2.2 RQ2. ¿Qué marco de trabajo seleccionar para mejorar los servicios de la aplicación Pluviopivot?	13
1.3 Estudio de lenguajes, herramientas y entorno de desarrollo a utilizar.....	16
1.3.1 JavaScript	16
1.3.2 MEAN	16
Conclusiones parciales.....	19
CAPÍTULO 2. ANÁLISIS Y DISEÑO DEL SISTEMA PARA LA EVALUACIÓN DE LAS MÁQUINAS DE RIEGO DE PIVOTE CENTRAL	21
2.1 Proceso de recolección de datos y cálculo de los parámetros de uniformidad.	21
2.2 Propuesta de solución computacional.....	24
2.3 Diagrama de casos de uso del sistema	25
2.3.1 Descripción de los casos de uso	26
2.4 Modelado de datos en MongoDB	30
2.5 Diagrama de paquetes	32
2.6 Diagrama de clases	33
2.7 Diagrama de componentes.....	36
2.8 Diagrama de despliegue.....	38
2.9 Maqueta del sitio web	38
2.10 Diagrama de navegación.....	40

TABLA DE CONTENIDOS

2.11 Conclusiones parciales.....	41
CAPÍTULO 3. PRUEBAS SOBRE LA APLICACIÓN PARA LA EVALUACIÓN DE MÁQUINAS DE RIEGO DE PIVOTE CENTRAL.....	42
3.1 Pruebas de caja blanca.....	42
3.1.1 Pruebas unitarias.....	43
3.2 Pruebas de caja negra.....	45
3.2.1 Pruebas de sistema.....	46
Conclusiones parciales.....	54
CONCLUSIONES.....	55
RECOMENDACIONES.....	56
BIBLIOGRAFÍA.....	57
ANEXOS.....	59

LISTA DE FIGURAS

<i>Figura 1. Gráfico que muestra la frecuencia de aparición de los frameworks en la literatura científica</i>	10
<i>Figura 2. Esquema del área que riega una máquina con una línea de n pluviómetros y las áreas de las coronas circulares asignadas.</i>	22
<i>Figura 3. Diagrama de casos de uso del sistema</i>	26
<i>Figura 4 Diagrama de paquetes</i>	33
<i>Figura 5: MVC en Angular</i>	34
<i>Figura 6: Diagrama de clases del Cliente</i>	35
<i>Figura 7: Diagrama de clases del Servidor</i>	36
<i>Figura 8: Diagrama de componentes del sistema</i>	37
<i>Figura 9: Esquema de la arquitectura y proceso de la aplicación web</i>	37
<i>Figura 10: Diagrama de despliegue del sistema</i>	38
<i>Figura 11: Maqueta del sitio web</i>	40
<i>Figura 12: Diagrama de navegación para el sistema de evaluación de máquinas de riego de pivote central</i>	41
<i>Figura 13: Resultado de las pruebas de la aplicación</i>	43
<i>Figura 14: Validación de la función media_ponderada()</i>	44
<i>Figura 15: Validación de la función coeficiente_variacion_ponderado()</i>	44
<i>Figura 16: Informe generado con el reporter Coverage</i>	45
<i>Figura 17: Entrada "Autenticar Usuario"(prueba 1).</i>	47
<i>Figura 18: Salida "Autenticar Usuario"(prueba 1).</i>	48
<i>Figura 19: Entrada "Autenticar Usuario"(prueba 2).</i>	49
<i>Figura 20: Salida "Autenticar Usuario"(prueba 2).</i>	50
<i>Figura 21: Entrada "Añadir Evaluación"(prueba 1).</i>	51
<i>Figura 22: Salida "Añadir Evaluación"(prueba 1).</i>	52
<i>Figura 23: Entrada "Añadir Evaluación"(prueba 2).</i>	53
<i>Figura 24: Salida "Añadir Evaluación"(prueba 2).</i>	54
<i>Figura 25: Formulario de acceso a la aplicación</i>	59
<i>Figura 26: Página principal de la aplicación (Rol Administrador)</i>	60

LISTA DE FIGURAS

<i>Figura 27:Página principal de la aplicación (Rol Jefe)</i>	<i>61</i>
<i>Figura 28:Página principal de la aplicación (Rol Evaluador).....</i>	<i>62</i>
<i>Figura 29:Barra de navegación de la aplicación</i>	<i>64</i>
<i>Figura 30:Gestionar Usuarios</i>	<i>65</i>
<i>Figura 31:Formulario para adicionar un nuevo usuario</i>	<i>66</i>
<i>Figure 32:Gestionar evaluaciones planificadas</i>	<i>67</i>
<i>Figura 33: Formulario para filtrar evaluaciones por fecha</i>	<i>68</i>
<i>Figura 34:Formulario para agregar una evaluación planificada.....</i>	<i>69</i>
<i>Figura 35:Área de trabajo Evaluar.....</i>	<i>71</i>
<i>Figura 36:Formulario para la gestión de datos de la evaluación pluviométrica.....</i>	<i>72</i>
<i>Figura 37:Ejemplo de una evaluación terminada.....</i>	<i>74</i>
<i>Figura 38:Área de trabajo Gestionar Perfil</i>	<i>75</i>
<i>Figure 39:Aval emitido por la dirección de la empresa “Valle del Yabú”</i>	<i>77</i>

LISTA DE TABLA

<i>Tabla 1:Formulario para la extracción de datos</i>	8
<i>Tabla 2:Publicaciones extraídas de cada base de datos</i>	9
<i>Tabla 3:Principales características de los marcos de trabajo encontrados</i>	10
<i>Tabla 4:Requisitos de evaluación para cada framework</i>	14
<i>Tabla 5:Descripción del caso de uso "Gestionar usuarios"</i>	26
<i>Tabla 6:Descripción del caso de uso "Gestionar Perfil"</i>	27
<i>Tabla 7:Descripción del caso de uso "Gestionar evaluaciones planificadas"</i>	28
<i>Tabla 8: Descripción del caso de uso "Evaluar"</i>	28
<i>Tabla 9:Descripción del caso de uso "Ver resultados de la evaluación pluviométrica"</i>	29
<i>Tabla 10:Descripción del caso de uso "Configurar idioma"</i>	30
<i>Tabla 11:Descripción del caso de prueba "Autenticar Usuario"(prueba 1)</i>	46
<i>Tabla 12:Descripción del caso de prueba "Autenticar Usuario"(prueba 2)</i>	48
<i>Tabla 13:Descripción del caso de prueba "Añadir Evaluación"(prueba 1)</i>	50
<i>Tabla 14:Descripción del caso de prueba "Añadir Evaluación"(prueba 2)</i>	52
<i>Tabla 15:Relación de cada elemento del formulario de acceso a la aplicación y el número correspondiente</i>	59
<i>Tabla 16:Relación entre cada elemento de la página principal(Administrador) y el número correspondiente</i>	62
<i>Tabla 17:Relación entre cada elemento de la página principal(Jefe) y el número correspondiente</i>	63
<i>Tabla 18:Relación entre cada elemento de la página principal(Evaluador) y el número correspondiente</i>	63
<i>Tabla 19:Relación entre cada elemento de la barra de navegación y el número correspondiente</i>	64
<i>Tabla 20:Relación entre cada elemento del área de trabajo Gestionar Usuarios y el número correspondiente</i>	65
<i>Tabla 21:Relación entre cada elemento del formulario Insertar Usuario y su número correspondiente</i>	66
<i>Table 22: Relación entre cada elemento del área de trabajo Gestionar evaluaciones planificadas y el número correspondiente</i>	67

LISTA DE TABLAS

Tabla 23:Relación entre cada elemento del formulario Filtrar evaluaciones por fecha y el número correspondiente68

Tabla 24:Relación entre cada elemento del formulario Nueva Evaluación y el número correspondiente69

Tabla 25:Relación entre cada elemento del área de trabajo Evaluar y el número correspondiente71

Tabla 26:Relación entre cada elemento del formulario para gestionar los datos de la evaluación y el número correspondiente72

Tabla 27:Relación entre cada elemento del área de trabajo Gestionar Perfil y el número correspondiente75

Tabla 28:Datos que registra la aplicación Pluviopivot75

INTRODUCCIÓN

En la empresa agrícola “Valle del Yabú” está implementada la aplicación desktop Pluviopivot que es la encargada de la evaluación de las máquinas de riego de pivote central. Dicha aplicación presenta diversos problemas que dificultan el proceso de evaluación por parte de los trabajadores de la empresa. Entre ellos tenemos los siguientes:

- No existen roles
- No existe una forma rápida e eficiente de compartir la información entre todos los usuarios
- No existe una base de datos centralizada
- Dificultades para actualizar el software

Para resolver estas deficiencias se propone el desarrollo de una aplicación web que sea capaz de:

- Gestionar los distintos roles de usuario en el proceso de evaluación (Administrador, Jefe de Área, Evaluador)
- Permitir que los usuarios puedan gestionar la evaluación de máquinas de riego mediante una red de computadoras.
- Guardar la información del proceso en una base de datos centralizada.

Además, la aplicación debe:

- Ser fácilmente actualizable
- Permitir que sus datos sean accedidos desde dispositivos móviles a través de la red

Cuando se habla de aplicaciones web hay que remontarse a su historia para poder observar el gran desarrollo de las mismas y el por qué hoy en día es mucho más práctico realizar aplicaciones web y no aplicaciones de escritorio pues las aplicaciones web solucionan todos los inconvenientes que son propios de las aplicaciones desktop y que no fue hasta su aparición que lograron solucionarse y he aquí el motivo por el cual hoy en día son mucho más usadas. Pero las aplicaciones web no siempre fueron como lo son en la actualidad pues en los primeros días de la web, los usuarios no tenían grandes expectativas de los sitios web, no se le dio mucho énfasis a la presentación se trataba más acerca de lo que estaba pasando por detrás de las escenas. Por lo general, si alguien sabía algo como Perl y juntaba un poco de cadenas HTML, entonces era un desarrollador web.

Con el uso y la popularidad de Internet, las empresas comenzaron a preocuparse de cómo se reflejaban en la web lo cual propició el surgimiento de implementaciones frontales más complicadas como resultado de la combinación de las hojas de estilo en cascada (CSS) y JavaScript. Debido a esto las implementaciones ya no eran tan sencillas pues se necesitaba

tener conocimientos de JavaScript y CSS para asegurarse de que todo funcionaba correctamente.(Holmes 2013)

En este escenario es donde surge la distinción entre desarrollador front-end y desarrollador de back-end. Mientras que los desarrolladores de back-end se centraron en todo lo que era el código que corría detrás de la interfaz de usuario, los desarrolladores front-end se centraron en la construcción de una buena experiencia de usuario. Con el tiempo, se hicieron mayores las expectativas en ambos campos, lo que estimuló a que los desarrolladores tuvieran que elegir una experiencia y centrarse en ella.(Holmes 2013)

Durante los años 2000, las bibliotecas y marcos de trabajo comenzaron a ser populares y fueron implementados para los lenguajes más comunes, tanto en el back-end como en el front-end.

Los marcos de trabajo fueron diseñados para hacer la vida de un desarrollador más fácil, reduciendo las barreras de entrada. Una buena biblioteca o marco de trabajo abstrae algunas de las complejidades más frecuentes, lo que le permite codificar más rápido y requiere menos experiencia en profundidad. Esta tendencia hacia la simplificación ha dado lugar a un resurgimiento de los desarrolladores tanto de front-end como de back-end dando lugar así al concepto de desarrolladores full-stack. Ha habido, por supuesto, desarrolladores de full-stack a lo largo de toda la historia de la web, y es probable que algunos desarrolladores elijan especializarse en el desarrollo de front-end o back-end. Pero la intención del uso de marcos y herramientas es demostrar que mediante ellos ya no hay que elegir un lado u otro para ser un buen desarrollador web.(Holmes 2013)

Una gran ventaja de adoptar el enfoque de marco es que puede ser increíblemente productivo, ya que tendrá una visión general de la aplicación y de cómo interactúan juntas ambas partes.

Continuando con la tendencia de los marcos, en los últimos años se ha registrado una tendencia a mover la lógica de la aplicación lejos del servidor y llevarla hacia el front-end. Algunas de los frameworks más populares de JavaScript que hacen esto son AngularJS, Backbone y Ember.(Holmes 2013)

Al acoplar firmemente el código de la aplicación en el back-end empieza a difuminarse las líneas entre los desarrolladores front-end y back-end tradicionales. Una de las razones que la gente le gusta usar este enfoque es que reduce la carga en los servidores, por lo tanto, se reducen los costos. (Holmes 2013)

Las rutas de los desarrolladores front-end y back-end se están fusionando, y es totalmente posible ser plenamente competentes en ambas disciplinas. Poder desarrollar todo el alcance de un sitio web o una aplicación proporciona un mejor control general y puede ayudar a las diferentes partes a que trabajen perfectamente juntos, ya que no han sido construidos aisladamente por equipos separados.

La construcción de una aplicación completa es muy gratificante. Cada parte viene con sus propios desafíos y problemas para resolver, manteniendo las cosas interesantes. Las

tecnologías y herramientas disponibles hoy en día realzan esta experiencia y permiten construir aplicaciones web grandes con relativa rapidez y facilidad.

Hay muchos beneficios para el aprendizaje del desarrollo del full-stack. Para empezar, está el disfrute de aprender cosas nuevas y jugar con las nuevas tecnologías. También está la satisfacción de dominar algo diferente y la emoción de ser capaz de construir y lanzar una aplicación completa basada en datos.(Holmes 2013)

En general, hay mucho que decir para el desarrollo de full-stack. La comprensión general y la capacidad para ver el panorama más amplio de este tipo de programadores es un tremendo bono. Pero existen varias tecnologías para el desarrollo de aplicaciones web por lo que en nuestro trabajo nos proponemos el siguiente **problema de investigación**:

¿Qué tecnologías utilizar para desarrollar una versión web de la aplicación Pluviopivot de modo que se mejoren sus servicios?

Objetivo general de la investigación:

Desarrollar una versión web de la aplicación Pluviopivot que facilite la evaluación de máquinas de riego de pivote central en la empresa agrícola “Valle del Yabú”.

Para cumplir con el objetivo general se plantean los siguientes **objetivos específicos**:

1. Identificar los marcos de trabajo para el desarrollo de aplicaciones web existentes en la literatura científica.
2. Seleccionar el marco de trabajo apropiado para mejorar los servicios de la aplicación Pluviopivot.
3. Proponer una arquitectura que utilice el marco de trabajo seleccionado.
4. Implementar la arquitectura propuesta en una nueva versión de Pluviopivot

Para la solución del objetivo general y los específicos nos trazamos las siguientes **preguntas de investigación**:

- RQ1. ¿Cuáles son los principales marcos de trabajo para el desarrollo de aplicaciones web?
- RQ2. ¿Qué marco de trabajo seleccionar para mejorar los servicios de la aplicación Pluviopivot?
- RQ3. ¿Qué interfaz deben tener los servicios de la arquitectura propuesta?

Justificación de la investigación:

La justificación de esta investigación se considera que es de tipo práctica pues el desarrollo de un sitio web que facilite la gestión de la evaluación de máquinas de riego de pivote central tendrá un importante impacto en los trabajadores de la empresa “Valle del Yabú” pues se les facilitará su trabajo de manera significativa ya que se tendrá un mayor acceso a la información permitiendo así mejorar el proceso de evaluación de las máquinas de riego lo cual a su vez

ahorra una cantidad de agua considerable e influye en la producción pues evita la pérdida de productos agropecuarios contribuyendo así a mejorar la economía del país.

La estructura de la tesis es la siguiente:

Capítulo I. Estudio del marco teórico de las tecnologías que implementan la tendencia full-stack y su integración con dispositivos móviles. Se realiza una búsqueda en la literatura a partir de la técnica del SLR con el objetivo de profundizar en las principales tecnologías que utilizan la tendencia full-stack y que además permiten integrarse con dispositivos móviles, así como su arquitectura y los componentes que la integran, para de esta manera enfatizar en las herramientas, lenguajes y metodologías que se van a utilizar para desarrollar el caso de estudio de la aplicación Pluviopivot.

Capítulo II. Análisis y diseño del sistema para la evaluación de máquinas de riego de pivote central. Se propone la solución computacional hallada para resolver el problema en cuestión, se describen los casos de uso y explican las variables pluviométricas utilizadas. Se modela la base de datos con las consultas que gestionan las mismas. Se muestra la arquitectura del sistema y describen sus principales componentes.

Capítulo III. Pruebas sobre la aplicación para la evaluación de máquinas de riego de pivote central. Se analizan los principales aspectos sobre el funcionamiento de la herramienta creada. Se valoran las potencialidades de la herramienta con respecto a otras existentes en la literatura científica y su respuesta a diferentes casos de prueba.

CAPÍTULO 1. ESTUDIO DEL MARCO TEÓRICO DE LAS TECNOLOGÍAS, LENGUAJES, Y HERRAMIENTAS PARA EL DESARROLLO DE APLICACIONES WEB.

En el presente capítulo se hace una revisión sistemática de la literatura mediante la técnica del SLR para el estudio de los marcos de trabajo del desarrollo web y de esta forma determinar cuál de ellos es el que se utilizará en el software que se desea implementar. Con los resultados de dicha revisión se ofrecerá una caracterización de dichos marcos de trabajo, así como ventajas y desventajas de su uso. Se analizan y describen las tecnologías, lenguajes y herramientas que serán utilizadas para el desarrollo de la propuesta de solución.

1.1 Revisión Sistemática de la literatura: Diseño y ejecución

Las Revisiones Sistemáticas son un diseño de investigación observacional y retrospectivo, que sintetiza los resultados de múltiples investigaciones primarias. Son parte esencial de las Ciencias de la Computación basada en la evidencia por su rigurosa metodología, identificando los estudios relevantes para responder preguntas específicas de la práctica computacional (Kitchenham & Ebse 2007). El término metanálisis se reserva para la combinación numérica de los datos. No todas las revisiones sistemáticas lo incluyen. Las revisiones sistemáticas son tan populares que han tenido un crecimiento vertiginoso en los últimos años. Esta técnica está compuesta por 3 etapas fundamentales:

- 1) Planificación de la investigación: Se establece primeramente la necesidad de la revisión, luego se define un protocolo de búsqueda, un protocolo de investigación y después se hace una evaluación de la planificación.
- 2) Desarrollo de la investigación: Se establece una búsqueda de estudios primarios, se seleccionan dichos estudios, luego se hace una extracción y gestión de los datos y posteriormente se sintetizan los datos.
- 3) Publicación de los resultados: Se explica el uso dado a los resultados obtenidos de la revisión.

En este epígrafe se realizará una revisión sistemática de la literatura para investigar los principales marcos de trabajo para el desarrollo web y de esta manera determinar cuál de ellos es el más adecuado para implementar la nueva versión del software Pluviopivot.

1.1.1 Necesidad de la revisión sistemática de la literatura

La revisión sistemática de la literatura a realizar tiene como objetivo resumir la evidencia existente acerca de cuáles son los principales marcos de trabajo para el desarrollo de aplicaciones web y así determinar cuál de ellas utilizar para implementar la nueva versión de la aplicación Pluviopivot.

Preguntas de investigación:

- RQ1. ¿Cuáles son los principales marcos de trabajo para el desarrollo de aplicaciones web?
- RQ2. ¿Qué marco de trabajo seleccionar para mejorar los servicios de la aplicación Pluviopivot?

Recursos Disponibles:

- Internet
- IEEE Xplorer (<http://ieeexplore.ieee.org/Xplore/home.jsp>)
- ACM Digital Library (<http://dl.acm.org/>)
- Google Scholar (<http://scholar.google.com>)
- Science Direct (<http://www.sciencedirect.com>)
- Citeseer library (<http://citeseer.ist.psu.edu>).
- SCOPUS (https://dev.elsevier.com/sc_search_example.html)

Además de realizar búsquedas en las bibliotecas online previamente mencionadas se realizaron búsquedas en libros y artículos de autores importantes en la temática de investigación.

1.1.2 Protocolo de investigación

En el presente trabajo se pretende realizar una investigación en cuanto a los principales marcos de trabajo existentes para el desarrollo de aplicaciones web pues se desea implementar un sitio web tomando como objeto de estudio la aplicación pluviopivot implementada como aplicación desktop en la empresa Yabú que además también permita la conexión con dispositivos móviles por lo que para lograr dichos objetivos el protocolo de revisión debe estar encaminado a responder las preguntas de investigación definidas en el epígrafe 1.1.1.

Protocolo de búsqueda

Como estrategia de búsqueda de los artículos para dar respuesta a las preguntas de investigación nos proponemos un conjunto de términos y combinaciones de los mismos para realizar las búsquedas en las diferentes fuentes de datos.

Términos:

- full-stack
- mobile

- web
- components
- Android
- JavaScript
- Development
- Programming

Combinaciones de términos

- full stack JavaScript development
- full stack web development
- full stack web programming
- full stack web programming and Android
- full stack web programming and mobile

Estrategia de búsqueda

Se realizará una búsqueda a partir de las combinaciones de términos previamente mencionadas en las fuentes de datos enunciadas en el epígrafe 1.1.1 y se evaluará el volumen de estudios relevantes potenciales. También se consultará con expertos en el campo de la Computación.

Estudio del criterio de selección

Criterios de inclusión:

- Artículos que abarquen el tema de los marcos de trabajo para el desarrollo web
- Artículos que traten sobre las tecnologías web que permiten la interacción con dispositivos móviles.
- Artículos escritos en idioma inglés.

Criterios de exclusión

- Artículos que no tengan un lenguaje adecuado,
- Artículos que hayan sido publicados antes del 2006.
- Artículos que no contengan información completa sobre los temas de interés.

Estrategia para la extracción de datos

El investigador realizará una búsqueda primaria a partir de las combinaciones de términos mencionadas en el epígrafe 1.1.2.1 y luego realizará una segunda búsqueda para de esta manera evaluar la consistencia de los datos y chequear que se extraigan los mismos de manera correcta. A continuación, se muestra el formulario para la extracción de datos de los documentos (véase Tabla1).

Tabla 1:Formulario para la extracción de datos

Nombre del artículo	Autor	Biblioteca	Tecnologías Web que utiliza	Componentes que utiliza para permitir integración con dispositivos móviles	Gratis	Open Source	Antigüedad
A1							
A2							

1.1.3 Ejecución de la revisión sistemática de la literatura

En la ejecución de la SLR se procede a extraer las publicaciones dentro de las correspondientes bases de datos según las cadenas de búsquedas definidas anteriormente. Para ello se crea una carpeta llamada SLR que contiene varias carpetas con los nombres de cada base de datos y dentro de cada una de ellas se encuentra una lista de carpetas correspondiente a las cadenas de búsqueda. Cada carpeta de una cadena de búsqueda contiene un fichero llamado publicaciones.txt con los metadatos de los artículos encontrados que cumplen con los criterios de selección. Para incluir un artículo en el archivo publicaciones.txt se analiza su título y resumen, en caso de existir dudas se descarga y se revisan las conclusiones y el cuerpo de ser necesario.

Luego de crear todos los archivos publicaciones.txt se procede a crear el archivo de Microsoft Excel Artículos.xlsx que contiene las columnas del formulario para extracción de datos las cuales son: Autor, Biblioteca, Tecnologías Web que utiliza, Componentes, Gratis, Open Source y Antigüedad. Para cada publicación se crea una entrada en el Excel y se ingresan sus datos para cada una de las columnas. La ventaja de utilizar Microsoft Excel es que permite filtrar por la columna título al adicionar una publicación y así se evita adicionar trabajos

duplicados que pueden aparecer en distintas bases de datos. Para ingresar los datos primeramente se copian de los archivos publicaciones.txt el autor y el título del artículo completando así las dos primeras columnas. Para completar los valores de las columnas restantes se descargan los artículos de internet y se revisa su cuerpo. Si el cuerpo indica que la publicación es irrelevante al estudio se elimina de Artículos.xlsx. Finalmente se introducen las publicaciones de Artículos.xlsx en Mendeley, donde se crea una carpeta SLR y dentro de SLR una carpeta para cada base de datos que es donde se colocan las publicaciones.

A continuación, en la tabla 2 se muestra el número total de publicaciones.

Tabla 2: Publicaciones extraídas de cada base de datos

Base de datos	Cantidad de publicaciones
ACM Digital Library	2
Citeseer	2
IEEE Xplorer	1
Google Scholar	3
Science Direct	1
Scopus	2
Total	11

1.2 Resultados de la Revisión Sistemática de la Literatura

En el presente epígrafe se realizará el análisis de los resultados de la revisión sistemática de la literatura. Para ello se subdividirá en dos epígrafes cada uno de ellos correspondiente a las preguntas de investigación de la revisión sistemática. El epígrafe 1.2.1 corresponderá al análisis de los marcos de trabajo para el desarrollo de aplicaciones web y el epígrafe 1.2.2 a la selección de uno de los marcos de trabajo encontrados anteriormente es el más adecuado para realizar la versión web de Pluviopivot. Cada uno de estos epígrafes contendrá dos secciones:

- **Análisis y Resultados** sección que explica las posibles respuestas encontradas a la pregunta de investigación.
- **Discusión** sección que provee un resultado y discusión para la pregunta de investigación.

1.2.1 RQ1. ¿Cuáles son los principales marcos de trabajo para el desarrollo de aplicaciones web?

Análisis y Resultado

Para contestar esta pregunta de investigación se seleccionan todos los frameworks identificados en las publicaciones resultantes de la revisión que sigan la tendencia full stack web programming entre los cuales se identificaron MEAN, Php framework, Django, Ruby on Rails entre otras. En la Figura 1. se muestra la frecuencia con que aparecen estos marcos de trabajo en las publicaciones.

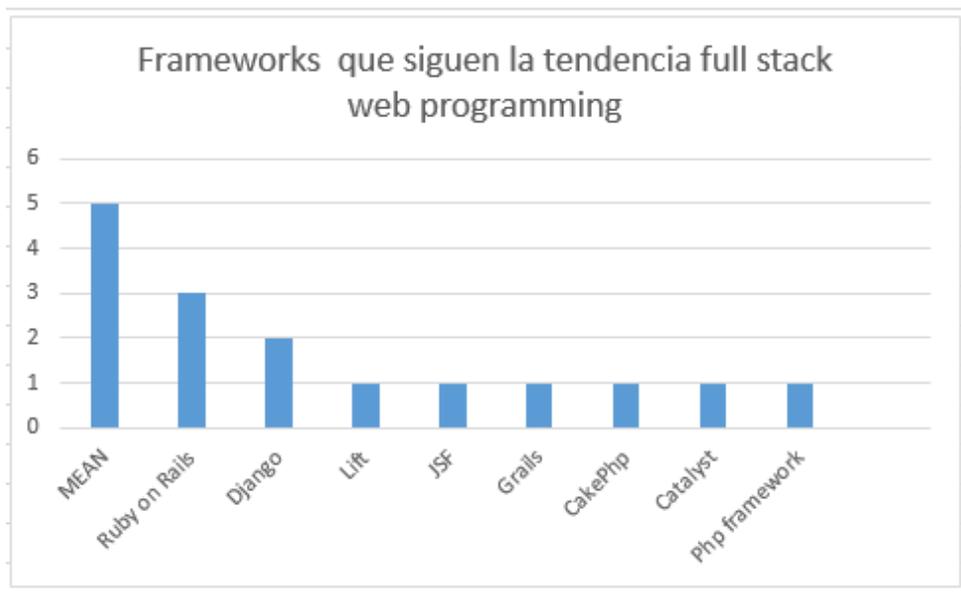


Figura 1. Gráfico que muestra la frecuencia de aparición de los frameworks en la literatura científica

Un total de 10 frameworks fueron identificados en las 11 publicaciones resultantes de la revisión, pero para complementar esta búsqueda se hicieron revisiones en internet para no descartar ningún otro framework existente. Al hacer esta búsqueda complementaria se encontraron otros marcos de trabajo como (Symfony y Meteor) todos estos quizás no aparezcan en las bibliotecas científicas puesto que son muy recientes. A continuación, se muestra en la Tabla 3 una breve caracterización de cada uno de los frameworks encontrados.

Tabla 3: Principales características de los marcos de trabajo encontrados

Frameworks	Principales Características
MEAN	<ul style="list-style-type: none"> • Utiliza JavaScript tanto en el back end como en el front end por lo que se puede codificar toda la aplicación en el mismo idioma. • Utiliza bases de datos no relacionales(MongoDB) • Utiliza Node.js para la creación del servidor web. Este constituye la base

CAPÍTULO 1. ESTUDIO DEL MARCO TEÓRICO DE LAS TECNOLOGÍAS QUE IMPLEMENTAN LA TENDENCIA FULL-STACK
Y SU INTEGRACIÓN CON DISPOSITIVOS MÓVILES

	<p>de MEAN.</p> <ul style="list-style-type: none"> • Permite la conexión con dispositivos móviles mediante una capa intermedia creada por Express.
JSF	<ul style="list-style-type: none"> • Framework web basado en el lenguaje Java • Soporta AJAX • Soporta framework JavaScript • Como formas de validación posee el JSF standard validators y el Bean validation • Soporte de HTML5 • Soporte de ORM(Object Relational Mapping) • Posee JavaDoc como documentación
Grails	<ul style="list-style-type: none"> • Framework basado en el lenguaje Groovy • Soporta AJAX • Posee como documentación Grails doc. • Como formas de validación tiene el Spring's validator • Soporta HTML5 (Modernizr)
CakePhp	<ul style="list-style-type: none"> • Framework basado en el lenguaje Php • Soporta AJAX • Formas de validación (Form helper) • No tiene soporte de HTML 5 • Como documentación cuenta con (phpdomain)
Php Frameworks	<ul style="list-style-type: none"> • Todas las aplicaciones desarrolladas siguen el modelo vista controlador (MVC). • Ausencia de mapeo de objetos relacionales(ORM) • Hace que el código sea más fácil y entendible para los programadores
Catalyst	<ul style="list-style-type: none"> • Soporta AJAX • Como documentación posee a Plain Old Documentation • Formas de validación (HTML: Form

CAPÍTULO 1. ESTUDIO DEL MARCO TEÓRICO DE LAS TECNOLOGÍAS QUE IMPLEMENTAN LA TENDENCIA FULL-STACK
Y SU INTEGRACIÓN CON DISPOSITIVOS MÓVILES

	<p>Handler y HTML: Form Validator)</p> <ul style="list-style-type: none"> • Soporte de HTML
Lift	<ul style="list-style-type: none"> • Framework basado en el lenguaje Scala • Posee las ventajas de la programación funcional y es que su código es fácil de probar y reusable. • Su código es simple de paralelizar • Posee una alta escalabilidad • Soporta AJAX
Ruby on Rails	<ul style="list-style-type: none"> • Todas las aplicaciones desarrolladas siguen el modelo vista controlador (MVC). • Este framework hace más fácil las pruebas a la aplicación. • Las aplicaciones Rails son escritas en Ruby. • Incorpora a su código AJAX e interfaces REST.
Django	<ul style="list-style-type: none"> • Las aplicaciones web son escritas en Python. • Todas las aplicaciones desarrolladas siguen el modelo vista controlador (MVC). • Es open source • Soporta AJAX • Documentación (Sphinx) • Soporte de HTML5 (HTML5 Boilerplate, H5BP) • Forma de validación (Django Form)
Meteor	<ul style="list-style-type: none"> • Las aplicaciones web son escritas con JavaScript • Sirve para móviles y web • Cuenta con una plataforma online • Usa menos código • Es integrado y flexible • Crea interfaces de usuario eficientes • Trabaja con el gestor de bases de datos MongoDB
Symphony	<ul style="list-style-type: none"> • Fácil de instalar y configurar en la mayoría de plataformas (y con la garantía de que funciona correctamente en los sistemas

	<p>Windows y Unix estándares)</p> <ul style="list-style-type: none">• Independiente del sistema gestor de bases de datos• Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos• Sigue la mayoría de las <i>mejores prácticas</i> y patrones de diseño para la web• Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo• Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros
--	--

Discusión

De los resultados expuestos anteriormente se puede observar que estos frameworks se separan en 3 grandes grupos uno es el que sigue las tecnologías que utilizan XAMP. Dentro de XAMP se encuentran los framework Lift, JSF, Grails, CakePhp, Symphony, Catalyst y Php framework los cuales permiten y hacen más fácil la programación con el lenguaje Php del lado del servidor mientras que el grupo de las tecnologías MEAN utilizan el lenguaje JavaScript tanto del lado del servidor como del lado del cliente y dentro de este grupo tenemos los frameworks MEAN y Meteor. Por su parte en el tercer grupo se agrupó a Ruby on Rails y Django pues estos soportan varios tipos de base de datos y además se pueden integrar con diversos servidores como Tomcat, Apache entre otros y los lenguajes utilizados para la programación en el servidor son Ruby y Python respectivamente. Para decidir que tecnología es la más apropiada para desarrollar la nueva versión de la aplicación pluviopivot se responderá la segunda pregunta de investigación.

1.2.2 RQ2. ¿Qué marco de trabajo seleccionar para mejorar los servicios de la aplicación Pluviopivot?

Análisis y Resultado

Para contestar esta pregunta de investigación se seleccionan todos los marcos de trabajo identificados en la pregunta de investigación RQ1 entre los cuales se identificaron MEAN, Php, Django, Ruby on Rails entre otras y se evaluarán una serie de requisitos que se necesitan para la implementación de la nueva versión de pluviopivot y en función de estos se hará la

selección del mejor marco de trabajo para desarrollar dicha aplicación. En la Tabla 4. se muestran los framework y los criterios de evaluación requeridos que cumplen cada uno de ellos.

Tabla 4: Requisitos de evaluación para cada framework

Framework s\Criterios	Permite integración con dispositivos móviles	Permite realizar llamadas asíncronas	Mismo lenguaje en la programación del front end y el back end	Velocidad de acceso a la base de datos	Calidad y cantidad de la documentación	Nivel de integración entre todas las tecnologías
MEAN	X	X	X	alto	alta	Alto
Meteor	X	X	X	alto	medio	Alto
Symfony	X	X		medio	alta	medio
Ruby on Rails		X		medio	media	medio
Catalyst	X	X		medio	alta	medio
Lift		X		medio	media	medio
CakePhp	X	X		medio	media	medio
JSF		X		medio	media	Medio
Grails		X		medio	media	Medio
Php framework	X	X		medio	baja	Medio
Django	X	X		medio	alta	Medio

Como se puede observar en la tabla anterior de las 11 tecnologías identificadas en RQ1 solamente 7 de ellas cuentan con un componente que permite la integración con dispositivos móviles entre ellas están MEAN y Meteor que cuenta con el módulo EXPRESS el cual permite crear una capa intermedia a la que se conecta el dispositivo móvil y utiliza el lenguaje JavaScript, otra es Django la cual cuenta con un plugin llamado Django-movi el cual es el encargado de esta tarea también y las otras son Php framework, CakePhp, Catalyst y Symphony que al integrarse a la tecnología XAMP permite la integración con dispositivos móviles

mediante un módulo AMP que utiliza código Php. En cuanto a las llamadas asíncronas todas cumplen este requisito pues todas tienen un soporte AJAX que es quien permite que ocurran llamadas asíncronas y en el caso de MEAN Node.js junto al código JavaScript es el encargado de que sean posibles las llamadas asíncronas.

En cuanto al mismo lenguaje de programación en el front end y en el back end las únicas que hacen esto posible son MEAN y Meteor pues utilizan JavaScript tanto en el front end como en el back end trayendo muchos beneficios y facilidades a la hora de hacer la aplicación, como por ejemplo, el fácil acceso a la base de datos pues al ser una base de datos con MongoDB y precisamente este gestor de bases de datos guardar sus datos con BSON (JSON binario) esta integración de código JavaScript en todas partes de la aplicación hace que sea mucho más fácil la conexión, el acceso y la interpretación de los datos, además, MongoDB garantiza la rapidez y la escalabilidad. La rapidez de acceso a la base de datos de las demás aplicaciones es medio debido a que ninguna tiene ningún elemento notorio en este aspecto todas se quedan en lo básico con respecto a este aspecto. En cuanto a la documentación vemos que están bien documentados los frameworks MEAN, Symphony y Catalyst pues además de que cuentan con más de 2 manuales propios también existe mucha documentación en Internet y en las bibliotecas científicas.

Por su parte Php Framework no posee un manual propio como todas las demás por lo que su documentación es baja y la del resto es media pues solamente cuentan con un manual para su documentación. En cuanto al nivel de integración de las tecnologías resalta MEAN entre todas pues se va más allá de lo que ofertan todos los demás frameworks pues cada una de las tecnologías que forman parte de ella tienen un propósito específico para estar ahí pues como ya explicamos Node.js es el encargado del lado del servidor y todo lo hace a través del lenguaje JavaScript ,Express por su parte brinda a Node.js todas las ventajas posibles para hacer más fácil al programador el trabajo con Node.js, MongoDB por su parte codifica los datos con BSON lo cual al estar tratando con una aplicación donde todo el lenguaje es JavaScript trae muchas ventajas y rapidez a la hora de la interpretación de estos datos y Angular por su parte constituye uno de los frameworks más potentes para el trabajo con JavaScript. Todo esto hace que la integración de estas tecnologías sea muy buena y sobresalga por encima de todas las demás.

Discusión

Por lo observado y explicado anteriormente se puede observar que existen dos framework que sobresalen por encima de los demás y que están muy parejos en cuanto a los requisitos planteados pues ambos siguen la misma filosofía ellos son MEAN y Meteor. Pero en el presente trabajo el framework seleccionado para desarrollar la aplicación Pluviopivot es MEAN pues no solo cumple con todos los requisitos con el máximo nivel al igual que Meteor, sino que tiene un mayor terreno experimentado ya que existen diversas publicaciones en las

diferentes librerías científicas mientras que de Meteor no se encontró ninguna solamente en la búsqueda complementaria realizada en Internet arrojó resultados sobre él, lo cual demuestra que MEAN ha sido más utilizado y probado por diversos autores de numerosos artículos brindando más seguridad para utilizar este marco de trabajo.

1.3 Estudio de lenguajes, herramientas y entorno de desarrollo a utilizar

1.3.1 JavaScript

JavaScript es un sencillo lenguaje de programación, constituido por programas, llamados comúnmente script. Estos scripts normalmente consisten en unas funciones que son llamadas desde el propio HTML cuando algún evento sucede. De ese modo, podemos añadir efectos como que un botón cambie de forma al pasar el ratón por encima, o abrir una ventana nueva al pulsar un enlace. JavaScript fue desarrollado por Netscape, a partir del lenguaje Java, el cual sigue una filosofía similar, aunque va más allá. Java es un lenguaje de programación por sí mismo, como lo puedan ser C, Pascal o VisualBasic. Esto quiere decir, que se puede ejecutar un programa Java fuera de un navegador. Pero, repetimos, que la diferencia fundamental es que Java es un lenguaje completo, que puede ser utilizado para crear aplicaciones de todo tipo, mientras que JavaScript sólo “funciona” dentro de una página HTML.

JavaScript fue declarado como estándar del European Computer Manufacturers' Association (ECMA) en 1997, y poco después también fue estandarizado por ISO. Sin embargo, la estructura de objetos que implementaban los diferentes navegadores (Netscape y Explorer en aquellos momentos) no se ajustaba al estándar, lo que provocaba numerosos problemas de compatibilidad. Para solventarlos el W3C publicó un nuevo modelo de objetos, DOM (Document Object Model), que incorporan la mayoría de navegadores actuales como Explorer o Firefox.(Holmes 2013)

1.3.2 MEAN

La filosofía MEAN reúne algunas de las tecnologías web modernas "mejores de su tipo" la misma es muy potente y flexible. Una de las grandes ventajas de esta filosofía es que no sólo utiliza JavaScript en el navegador, sino que lo utiliza en todas partes. Con la misma se puede codificar tanto el back end como el front end en el mismo idioma. La tecnología principal que permite que esto suceda es Node.js, permitiendo traer JavaScript al back end (Holmes 2013).MEAN está constituida por cuatro tecnologías **M**ongoDB, **E**xpress, **A**ngular y **N**ode.js.A continuación se explicará de manera independiente en qué consisten cada una de estas cuatro potentes tecnologías.

MongoDB

La capacidad de almacenar y utilizar datos es vital para la mayoría de las aplicaciones. En la filosofía MEAN, la base de datos de elección es MongoDB, la cual representa la M en MEAN. MongoDB encaja perfectamente en MEAN pues al igual que Node.js, es famoso por ser rápido y escalable.

Esta no es como las bases de datos relacionales que contienen en sí los conceptos de filas y columnas, sino que es una base de datos de documentos. El concepto de filas todavía existe, pero las columnas desaparecen. En lugar de una columna definir lo que debe estar en la fila, cada fila es un documento, y este documento define y contiene los datos en sí.

Este enfoque menos estructurado significa que una colección de documentos podría tener una amplia variedad de datos dentro (Holmes 2013).

MongoDB almacena los documentos como BSON, que es JSON binario (JavaScript Serialized Objects Notation). JSON es una forma JavaScript de almacenar datos, por lo que MongoDB se ajusta muy bien a al JavaScript que está centrado en la pila MEAN.

MongoDB se distingue de muchas otras bases de datos de documentos con su indexación secundaria y sus fáciles consultas. Esto significa que puede crear índices en otros campos además del campo identificador, siendo la consulta de campos indexados mucho más rápida. También se pueden crear algunas consultas bastante complejas contra una base de datos MongoDB, no con un gran nivel de comandos SQL con los que se une todo, pero lo suficientemente potente para la mayoría de los casos de uso (Holmes 2013).

MongoDB es una gran opción de base de datos para la mayoría de las aplicaciones web, ya que proporciona un equilibrio entre la velocidad de las bases de datos de documentos puros y el poder de las bases de datos relacionales. El hecho de que los datos se almacenen eficazmente en JSON es lo que la convierten en el almacén perfecto para la pila MEAN. (Holmes 2013)

Express

Express representa la E en MEAN. Como Node.js es una plataforma, no prescribe cómo debería ser configurado o utilizado lo cual constituye una de sus grandes fortalezas, pero al crear sitios web y aplicaciones web hay bastantes tareas comunes que se necesitan hacer cada vez. Express es un marco de aplicación web para Node.js que ha sido diseñado para hacer este de una manera bien probada y repetible (Holmes 2013).

Node.js es una plataforma no un servidor. Esto permite ser creativo con la configuración del servidor y hacer cosas que otros servidores web no pueden hacer. También lo hace más difícil para obtener el funcionamiento básico de un sitio web.

Express resuelve esta dificultad mediante la creación de un servidor web para escuchar las solicitudes entrantes y devolver las respuestas pertinentes. Además, también define una estructura de directorio donde una de estas carpetas está configurada para servir archivos

estáticos sin ningún tipo de interrupciones, de esta manera la aplicación no tiene que esperar cuando alguien más solicita un archivo CSS(Holmes 2013).

Una de las grandes características de Express es que proporciona una interfaz realmente simple para redirigir una URL entrante a una cierta parte de código ya sea la página HTML estática, la lectura de una base de datos, o escribir en una base de datos. La interfaz es simple y consistente. Lo que Express ha hecho aquí es muy simple comparado con la complejidad de hacer esto en Node.js nativo, lo que hace que el código sea más rápido de escribir y más fácil de mantener(Holmes 2013).

Otra de las ventajas que proporciona Express es la capacidad de utilizar las sesiones para que se puedan identificar visitantes individuales a través de múltiples solicitudes y páginas.

Como se puede apreciar Express le da una gran mano de ayuda y un buen punto de partida a Node.js para la construcción de aplicaciones web. Con su ayuda se abstraen una serie de complejidades y tareas repetibles de las cuales el programador no tiene que preocuparse. (Holmes 2013)

Angular

AngularJS es el A en MEAN. En términos simples, AngularJS es un framework JavaScript para trabajar directamente con datos en el front end.

La forma tradicional de hacer una aplicación web es tener todo el procesamiento de datos y la lógica de la aplicación en el servidor, que luego pasa el HTML hacia el navegador. AngularJS le permite mover todo o parte de este procesamiento y la lógica hacia el navegador, a veces dejando el servidor sólo pasar los datos de la base de datos.

AngularJS es lo que se conoce como un marco de opinión. Esto significa que fuerza su opinión sobre la del programador en cuanto a cómo debe ser utilizado. AngularJS ayuda a poner el HTML en función de los datos proporcionados, pero hace más que esto pues actualiza inmediatamente el HTML si los datos cambian, y también actualiza los datos si el HTML cambia. Esto se conoce como vinculación de datos bidireccional(Holmes 2013).

Algo para lo que AngularJS ha sido específicamente diseñado es para la aplicación de una sola página(SPA). En términos reales, un SPA ejecuta todo dentro del navegador y nunca hace una recarga de una página completa. Esto significa que toda la lógica de la aplicación, el procesamiento de datos, el flujo del usuario y la entrega de plantillas se pueden gestionar en el navegador. Este enfoque puede realmente reducir la cantidad de recursos que se necesita en el servidor. El navegador es quien realiza todo el trabajo de la aplicación, y el servidor básicamente sólo sirve archivos estáticos y datos bajo pedido. La experiencia del usuario también puede ser mejor cuando se utiliza este enfoque. Una vez cargada la aplicación hay menos llamadas a realizar en el servidor, reduciendo el potencial de latencia(Holmes 2013).

Node.js

Node.js constituye la N en MEAN. Pero no por ser el último significa que sea lo menos importante, En realidad es la base de MEAN. En pocas palabras, Node.js es una plataforma de software que permite crear su propio servidor web y construir aplicaciones web en la parte superior de la misma. Node.js no es un servidor web, ni es un lenguaje. Contiene una biblioteca de servidores HTTP integrada, por lo que no es necesario que ejecute un programa de servidor web independiente, como Apache o Internet Information Services (IIS). Esto finalmente proporciona un mayor control sobre cómo funciona un servidor web, pero también aumenta la complejidad de ponerlo en funcionamiento, especialmente en un entorno en vivo. Con PHP, por ejemplo, se puede encontrar fácilmente un host de servidor compartido en ejecución. Apache envía algunos archivos a través de FTP, y el sitio se está ejecutando. Esta funciona porque Apache ya lo ha configurado para que sea utilizado por los usuarios (Holmes 2013).

Con Node.js esto no ocurre, ya que configura el servidor Node.js al crear su aplicación. Muchos de los hosts web tradicionales están detrás de Node.js en este aspecto, pero una serie de nuevas plataformas como un servicio (PaaS) hosts están saltando a abordar esta necesidad, incluyendo Heroku, Nodejitsu y OpenShift. El enfoque de desplegar sitios en vivo en estos hosts PaaS es diferente del antiguo modelo FTP, pero es bastante fácil cuando se consigue entenderlo (Holmes 2013).

Node.js es un entorno de programación en la capa del servidor basado en el lenguaje de programación ECMAScript, asíncrono, con I/O de datos en una arquitectura orientada a eventos y basado en el motor V8 de Google. Fue creado con el enfoque de ser útil en la creación de programas de red altamente escalables, como, por ejemplo, servidores web (Holmes 2013).

Node.js es similar en su propósito a Twisted o Tornado de Python, Perl Object Environment de Perl, React de PHP, libevent o libev de C, EventMachine de Ruby y de Java existe Apache MINA, Netty, Akka, Vert.x, Grizzly o Xsocket. Al contrario que la mayoría el código JavaScript, no se ejecuta en un navegador, sino en el servidor. (Holmes 2013)

Conclusiones parciales

En el presente capítulo se realizó una revisión sistemática de la literatura con el objetivo de identificar los principales marcos de trabajo que siguen la tendencia full stack web programming resultando encontrados 11 marcos de trabajo y finalmente escogido para dar solución al problema planteado el framework MEAN, para dicha selección también se tuvieron en cuenta varios criterios por los cual fueron evaluados todos los marcos de trabajo encontrados y finalmente se realizó un estudio de esta herramienta seleccionada y de los lenguajes que utiliza. En el siguiente capítulo se analizará el proceso de evaluación de

**CAPÍTULO 1. ESTUDIO DEL MARCO TEÓRICO DE LAS TECNOLOGÍAS QUE IMPLEMENTAN LA TENDENCIA FULL-STACK
Y SU INTEGRACIÓN CON DISPOSITIVOS MÓVILES**

máquinas de riego de pivote central y se realizará el diseño de la propuesta de solución computacional.

CAPÍTULO 2. ANÁLISIS Y DISEÑO DEL SISTEMA PARA LA EVALUACIÓN DE LAS MÁQUINAS DE RIEGO DE PIVOTE CENTRAL

En el presente capítulo se explica el proceso necesario para la evaluación de máquinas de riego de pivote central mediante la recolección de datos y el cálculo de los parámetros de uniformidad. También se describe el análisis y diseño de la solución computacional dada al problema planteado mediante el diagrama de casos de uso del sistema y la descripción de cada uno de ellos. Se explica el proceso de modelado de datos con MongoDB y se presentan la estructura de la base de datos no relacional de la aplicación. Además, se presenta el diagrama de clases que permite dar solución a cada caso de uso, el diagrama de componentes y el diagrama de despliegue. Por último, se ofrece una vista del diseño de la aplicación mediante la maqueta del sitio web y el diagrama de navegación.

2.1 Proceso de recolección de datos y cálculo de los parámetros de uniformidad.

El estudio de la pluviometría en las máquinas de riego de pivote central resulta más complejo que en otros sistemas de aspersión, porque la corona circular representada por cada pluviómetro es mayor en la medida que su posición se aleja del pivote. Luego, el peso relativo de los diferentes pluviómetros en el cálculo de los parámetros de uniformidad no puede ser igual, lo que da lugar a que se introduzca un factor de ponderación que se corresponde con la superficie que representa cada pluviómetro y por consiguiente todos los coeficientes de uniformidad calculados resultan ponderados por la superficie (Cárdenas et al. n.d.).

Materiales y métodos

Para la ponderación de las alturas de agua recogidas en los pluviómetros, se asignó a cada uno de ellos, un área igual a la corona circular situada entre las circunferencias de radios $R_i + a/2$ y $R_i - a/2$, siendo " R_i " la distancia del centro del pivote al pluviómetro que ocupa la posición " i " y " a " la equidistancia entre pluviómetros según la Figura 2. Las áreas de las coronas circulares $A_1, A_2, A_3 \dots A_n$ señaladas en la Figura 2, vienen dadas por las expresiones siguientes:

$$\begin{aligned}
 A_1 &= \pi (z + a/2)^2 - \pi (z - a/2)^2 = 2 \pi a z \\
 A_2 &= \pi (z + 3 a/2)^2 - \pi (z + a/2)^2 = 2 \pi a (z + a) \\
 A_3 &= \pi (z + 5 a/2)^2 - \pi (z + 3 a/2)^2 = 2 \pi a (z + 2a) \\
 A_n &= \pi (z + (2n-1) a/2)^2 - \pi (z + (2n-3) a/2)^2 = 2 \pi a [z + (n-1) a]
 \end{aligned}
 \tag{1}$$

Cuando la distancia desde el centro del pivote al primer pluviómetro (z) es igual a la equidistancia entre pluviómetros (a), las áreas asignadas a los pluviómetros en el sistema de ecuaciones (1), toman los valores siguientes:

$$\begin{aligned} A_1 &= 2\pi a^2 \times 1 & A_3 &= 2\pi a^2 \times 3 \\ A_2 &= 2\pi a^2 \times 2 & A_n &= 2\pi a^2 \times n \end{aligned} \quad (2)$$

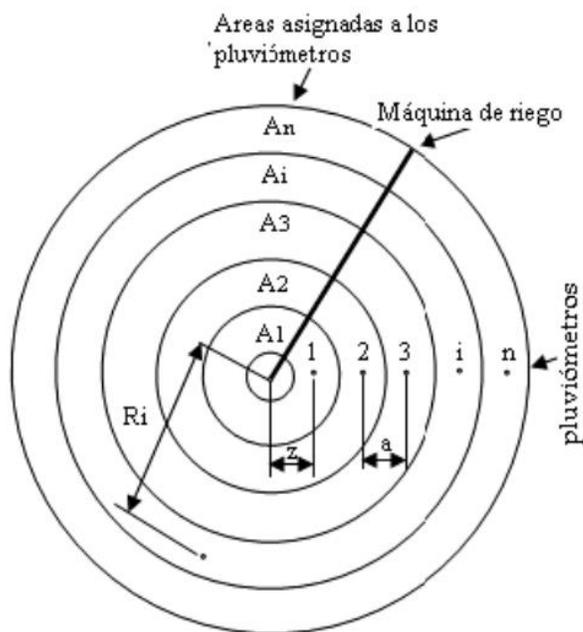


Figura 2. Esquema del área que riega una máquina con una línea de n pluviómetros y las áreas de las coronas circulares asignadas.

De acuerdo a lo descrito en el sistema de ecuaciones (2), puede deducirse que la ponderación de la altura de agua en los pluviómetros respecto a la superficie que representan, es equivalente a una ponderación respecto a sus posiciones, asignando el valor 1 al más cercano al centro del pivote, 2 al siguiente y así hasta un valor “n” para el que ocupa la posición más alejada del centro de pivote. Los parámetros de uniformidad ponderados se calculan teniendo en cuenta las expresiones anteriores y por las fórmulas siguientes:

Media ponderada

$$m_p = \frac{\sum_{i=1}^n m_i \times n_i}{\sum_{i=1}^n n_i} \quad (3)$$

Donde:

m_p = media ponderada de las alturas de agua recogidas.

m_i = altura de agua recogida en el pluviómetro “i”.

n = número de pluviómetros.

n_i = posición ocupada por el pluviómetro “i” con valor de 1 para el más cercano al pivote, 2 para el siguiente y así sucesivamente.

Coefficiente de Variación Ponderado

$$CV_p = \frac{S_p}{m_p} \times 100 \quad (4)$$

Donde:

CV_p = Coeficiente de variación ponderado de las alturas de agua recogidas %.

S_p = desviación típica ponderada de las alturas de agua recogidas.

Desviación Típica Ponderada

$$S_p = \sqrt{\frac{\sum_{i=1}^n n_i (m_i - m_p)^2}{\sum_{i=1}^n n_i - 1}} \quad (5)$$

Donde:

S_p = Desviación típica ponderada de las alturas de agua recogidas.

Coeficiente de Uniformidad Ponderado

$$CU_p = \left[1 - \sqrt{\frac{\sum_{i=1}^n n_i (m_i - m_p)}{m_p \times \sum_{i=1}^n n_i}} \right] \quad (6)$$

Donde:

CU_p = coeficiente de uniformidad ponderado de las alturas de agua recogidas.

Los valores de CU_p pueden clasificarse según SIAR (2003) de la forma siguiente:

< 80 % Inaceptable

80-85 % Aceptable

85-90 % Bueno

> 90 % Muy Bueno

Media Ponderada del 25% menos regado

Para el cálculo de $m_{p25\%}$ se realiza una ordenación de menor a mayor de las alturas de agua recogidas. Esos valores se multiplican por su posición ocupada en el radio del pivote y se suman hasta que el número de posiciones de esa suma alcance un valor igual o inmediatamente superior al 25% de la suma de posiciones totales.

$$m_{25p} = \frac{\sum_{j=1}^b m_j n_j}{\sum_{j=1}^b n_j} \quad (7)$$

Donde:

m_{25p} = altura media ponderada recogida en el 25% del área menos regada.

J = subíndice de ordenación de menor a mayor de las alturas de agua recogidas en los pluviómetros.

b = es un valor tal que se cumpla que: $\sum_{j=1}^b n_j \cong \frac{\sum_{i=1}^n n_i}{4}$

Uniformidad de Distribución Ponderada

$$UD_p = m_{25p} / m_p$$

Donde:

UD_p = uniformidad de distribución ponderada de las alturas de agua recogidas.

2.2 Propuesta de solución computacional

Atendiendo a las necesidades de la empresa agrícola “Valle del Yabú” y a las deficiencias presentadas por el sistema Pluviopivot actual, se propone desarrollar una aplicación que facilite el proceso de evaluación de máquinas de riego de pivote central generado diariamente en dicho centro y permita gestionar los distintos roles de usuarios existentes en el proceso de evaluación y que actualmente no se encuentran implementados. El sistema también ofrecerá la posibilidad de consultar la información relativa a este proceso a través de una red de computadoras de modo que se divulgará la información de una manera más fácil y rápida y dicha información será guardada en una base de datos centralizada. La aplicación permitirá además visualizar tablas y gráficos con el comportamiento de las distintas métricas presentes en cada evaluación y que son de gran utilidad a la hora de tomar decisiones en cuanto a la eficiencia de las máquinas.

Se ha decidido desarrollar este sistema como una aplicación web siguiendo una arquitectura de tres capas basado en el patrón MVC con una base de datos no relacional que permita el acceso a la información de forma segura y eficiente. El sistema a desarrollar ofrecerá además servicios web a través de una API a la cual podrán acceder dispositivos móviles para que de esta manera el evaluador pueda gestionar los datos de la evaluación desde una red wifi. Para el desarrollo de esta aplicación se pretende utilizar la filosofía MEAN que utiliza Node.js como servidor, Express para mejorar y añadir funcionalidades a Node.js entre ellas la conexión a dispositivos móviles de una manera rápida y segura, MongoDB como gestor de Bases de Datos y AngularJS que es un completo framework diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web basado en el patrón MVC. Para empezar, requiere escribir menos código que con otros frameworks pues hay muchas mejoras que son realmente críticas como el "doble bindign" que permite que los distintos componentes de la aplicación estén al tanto de los cambios para modificar su estado automáticamente, sin necesidad de suscribirse a eventos y realizar otro tipo de acciones por medio de líneas de código, mejora el HTML existente, facilitando el desarrollo de aplicaciones. En este punto cabe recalcar la palabra "aplicaciones" puesto que este tipo de herramientas son adecuadas para realizar las llamadas "aplicaciones de gestión" o "aplicaciones de negocio", además utiliza y recrea el concepto de un SPA (Single Page Application) liberando de esta manera al servidor de la mayor parte del trabajo realizándolo casi

todo el propio navegador. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación.

AngularJS es JavaScript. Es un proyecto de código abierto, realizado en JavaScript que contiene un conjunto de bibliotecas útiles para el desarrollo de aplicaciones web y propone una serie de patrones de diseño para llevarlas a cabo. En pocas palabras, es lo que se conoce como un framework para el desarrollo, en este caso sobre el lenguaje JavaScript con programación del lado del cliente.

El desarrollo de una aplicación web solucionará las dificultades de la aplicación desktop actualmente utilizada como la ausencia de roles del usuario, los problemas para la actualización, la no existencia de una base de datos centralizada que permita compartir la información de manera rápida y eficiente entre todos los usuarios, así como la necesidad de hacer las evaluaciones desde las distintas áreas de trabajo.

2.3 Diagrama de casos de uso del sistema

Los casos de uso han sido adoptados casi universalmente para la captura de requisitos de sistemas de software en general, y de sistemas basados en componentes en particular, pero los casos de uso son mucho más que una herramienta para capturar requisitos. Dirigen el proceso de desarrollo en su totalidad. Los casos de uso son la entrada fundamental cuando se identifican y especifican clases, subsistemas e interfaces, cuando se identifican y especifican casos de prueba, y cuando se planifican las iteraciones del desarrollo y la integración del sistema. Para cada iteración, nos guían a través del conjunto completo de flujos de trabajo, desde la captura de requisitos, pasando por el análisis, diseño e implementación, hasta la prueba, enlazando estos diferentes flujos de trabajo (I. Jacobson, G.Booch 2000).

A continuación, se muestra el diagrama de Casos de Uso y actores del sistema a implementar (véase la Figura 3).

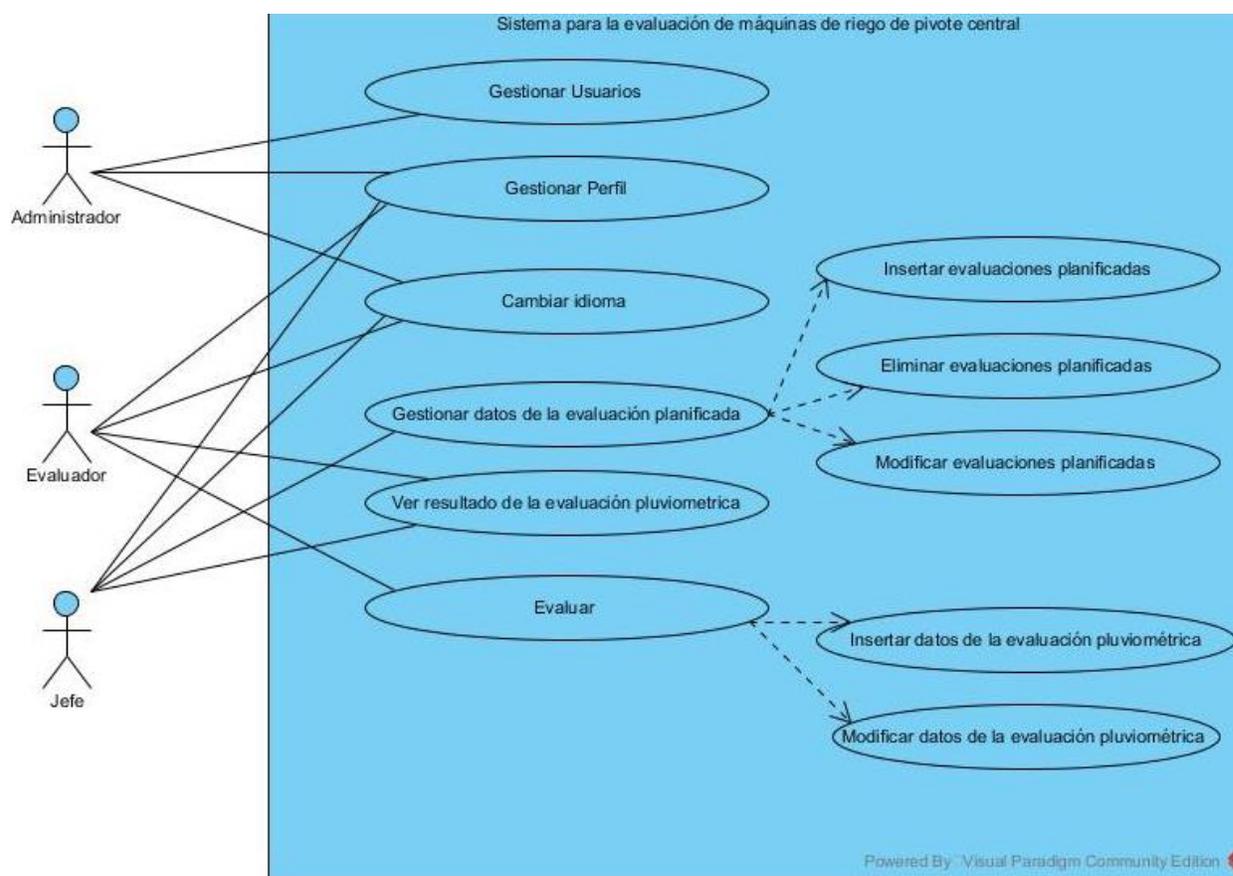


Figura 3. Diagrama de casos de uso del sistema

2.3.1 Descripción de los casos de uso

En el presente epígrafe se describirán mediante las siguientes tablas en qué consisten cada uno de los casos de uso del sistema.

En la tabla 5 se muestra la explicación del caso de uso “Gestionar usuarios”.

Tabla 5: Descripción del caso de uso "Gestionar usuarios"

Caso de uso:	Gestionar usuarios
Propósito	Permite al administrador insertar un usuario que no se encuentre en la base de datos de la aplicación así como modificar, eliminar o filtrar datos de uno existente.
Actor	Administrador
Precondiciones	Autenticarse en el sistema
Resumen	El caso de uso inicia cuando el actor accede al

	menú de la aplicación “Usuarios” donde se muestra toda la información de los usuarios almacenada en la base de datos permitiendo insertar un nuevo usuario o modificar y eliminar uno existente. El sistema valida todos los datos antes de completar cualquier operación.
--	--

La Tabla 6. muestra la descripción del caso de uso “Gestionar Perfil” donde el usuario logueado puede modificar los datos de su perfil.

Tabla 6: Descripción del caso de uso "Gestionar Perfil"

Caso de uso:	Gestionar perfil
Propósito	Permite al usuario logueado gestionar datos específicos de su perfil como cambiar su contraseña y otros datos personales.
Actor	Administrador, Jefe, Evaluador
Precondiciones	Autenticarse en el sistema
Resumen	El caso de uso inicia cuando el actor accede al menú de la aplicación “Gestionar Perfil” donde se muestra toda la información del actor logueado en ese momento y permite modificar la información personal de este actor como su contraseña u otros datos personales.

Un servicio fundamental de la aplicación Pluviopivot es poder gestionar toda la información relativa a las evaluaciones planificadas pues de esta manera el jefe de área planifica las evaluaciones para una máquina determinada en una fecha puesta por él para luego el evaluador poder completar estas evaluaciones en tiempo pasando estas de ser planificadas a terminadas y de esta manera dar cumplimiento a este proceso de evaluación de máquinas de riego de pivote central. Es importante mencionar que el jefe debe poder visualizar los resultados de cada evaluación por lo que existe el caso de uso “Ver resultados de la evaluación pluviométrica” Las Tablas 7,8 y 9 muestran las descripciones de los casos de uso “Gestionar evaluaciones planificadas”, “Evaluar” y “Ver resultados de la evaluación pluviométrica” los cuales están implicados en este proceso.

Tabla 7: Descripción del caso de uso "Gestionar evaluaciones planificadas"

Caso de uso:	Gestionar evaluaciones planificadas
Propósito	Permite al actor planificar las evaluaciones que son requeridas para una máquina dada en una determinada fecha.
Actor	Jefe
Precondiciones	Autenticarse en el sistema
Resumen	El caso de uso inicia cuando el actor accede al menú "Evaluaciones" donde se muestra la máquina, la fecha y el evaluador para la que fue programada dicha revisión y el estado en que está que puede ser <i>Planificada</i> si todavía no se ha realizado, <i>Iniciada</i> si ya comenzó y <i>Terminada</i> si ya se realizó. Si el estado de la evaluación es <i>Planificada</i> el actor puede modificar los datos relativos a una evaluación o eliminarla. Además, también puede agregar nuevas evaluaciones. El sistema valida todos los datos antes de realizar cualquier operación.

Tabla 8: Descripción del caso de uso "Evaluar"

Caso de uso:	Evaluar
Propósito	Permite al actor completar la información requerida para las evaluaciones planificadas y de esta manera poder cumplir con el proceso de evaluación de máquinas de riego de pivote central.
Actor	Evaluador
Precondiciones	Autenticarse en el sistema
Resumen	El caso de uso inicia cuando el actor accede al menú "Evaluar" mostrándose una lista de todas las evaluaciones a realizar y al dar click sobre una de ellas automáticamente se muestra un formulario con los campos

	<p>necesarios para realizar la evaluación. Al llenar este formulario la evaluación pasa al estado de <i>Iniciada</i> y por tanto deja de ser una evaluación planificada. Si la evaluación está en el estado de <i>Iniciada</i> pueden modificarse sus datos. Al entrar el diámetro y los volúmenes de la evaluación se lleva a cabo el proceso de evaluación y se muestran una serie de medidas calculadas, así como un gráfico de las mismas. Una vez terminada la evaluación se marca como <i>Terminada</i>, después del cual no podrá ser modificada. Todos los datos son validados antes de completar la operación.</p>
--	---

Tabla 9: Descripción del caso de uso "Ver resultados de la evaluación pluviométrica"

Caso de uso:	Ver resultados de la evaluación pluviométrica
Propósito	Permite al actor visualizar toda la información de una determinada evaluación planificada una vez que esta esté cumplida.
Actor	Jefe, Evaluador
Precondiciones	Autenticarse en el sistema
Resumen	El caso de uso inicia cuando el actor accede al menú "Evaluaciones" mostrándose una tabla con todas las evaluaciones planificadas por máquina, fecha, evaluador y el estado en que se encuentran. Al presionar sobre una fila de la tabla se muestran los resultados de la evaluación correspondiente incluyendo el gráfico de la evaluación permitiendo así visualizar la información recopilada y por tanto los resultados de la evaluación pluviométrica.

En la Tabla 10 se recoge toda la información del caso de uso "Configurar idioma".

Tabla 10: Descripción del caso de uso "Configurar idioma"

Caso de uso:	Configurar idioma
Propósito	Permite al actor cambiar el idioma del sistema
Actor	Jefe, Evaluador, Administrador
Precondiciones	Autenticarse en el sistema
Resumen	El caso de uso inicia cuando presiona uno de los botones disponibles en la barra de navegación para cambiar el idioma a Español o Inglés, cambiando así todo el idioma del sistema.

2.4 Modelado de datos en MongoDB

MongoDB es una base de datos orientada a documentos no una base de datos relacional. La principal razón para abandonar el esquema relacional es la fácil escalabilidad. Una base de datos orientada a documentos reemplaza el concepto de filas por un modelo más flexible, los "documentos". Este modelo hace posible representar relaciones jerárquicas complejas con un solo registro. Además, no existen esquemas de datos predefinidos, las llaves o valores de un documento no son de tipos o tamaños fijos. Sin un esquema fijo agregar o quitar los campos es mucho más fácil. Por lo que hace el desarrollo mucho más rápido (Chodorow 2013).

El corazón de MongoDB está en los documentos. La representación de un documento varía por el lenguaje de programación. En JavaScript, por ejemplo, los documentos son representados por la combinación llave: valor, por ejemplo:

```
{"greeting": "Hello, world!"}
```

Otro término importante dentro de MongoDB son las colecciones las cuales están constituidas por un grupo de documentos. De esta manera si un documento es equivalente a una fila en las bases de datos relacionales entonces una colección es equivalente a una tabla. Las colecciones tienen esquemas dinámicos lo que quiere decir que dentro de una misma colección pueden existir documentos con distintos formatos. Es importante mencionar que una colección es identificada por un nombre (Chodorow 2013).

MongoDB agrupa documentos por colecciones y también agrupa colecciones por bases de datos. Cada base de datos tiene sus propios permisos y está almacenada en archivos separados en el disco. Una buena práctica a seguir es almacenar todos los archivos de una aplicación en una misma base de datos. Cada base de datos es identificada por un nombre (Chodorow 2013).

Para la aplicación Pluviopivot se creó una base de datos llamada PluviopivotDB la cual posee 3 colecciones:

- Evaluaciones (en esta colección se almacenan todos los documentos que corresponden a las evaluaciones de las máquinas de riego de pivote central)
- Usuarios (en esta colección se almacenan todos los datos de los usuarios del sistema)
- Roles (en esta colección se almacenan todos los roles existentes en el sistema)

A continuación, se mostrará el esquema de cada colección.

Colección Evaluaciones

```
var evaluacionesSchema = new mongoose.Schema ({
maquina: {type: String, required: true},
fecha: {type: Date, required: true},
estado: {type: String, required: true},
horaI: {type: String},
horaF: {type: String},
diametro: {type: Number, required: true},
presion: {type: Number},
velocidadUT: {type: Number},
cultivoFase: {type: String},
equidistanciaP: {type: Number},
laminaR: {type: Number},
velocidadV: {type: Number},
regulacion: {type: String},
direccionV: {type: String},
direccionP: {type: String},
volumenes: [Number],
media ponderada: {type: Number},
media_ponderada25: {type: Number},
coeficiente_variacionP: {type: Number},
coeficiente_uniformidadP: {type: Number},
uniformidad_distP: {type: Number},
evaluador: {type: String, required: true}
});
```

Colección Usuarios

```
var userSchema = new mongoose.Schema ({
login: {
type: String,
unique: true,
required: true
},
name: {
type: String,
required: true
},
hash: String,
salt: String,
primer_apellido: {type: String, required: true},
seg_apellido: {type: String, required: true},
rol: String
});
```

Colección Roles

```
var rolesSchema = new mongoose.Schema ({
name_rol: {type: String, required: true},
usuarios: [String]
});
```

2.5 Diagrama de paquetes

Debido a la complejidad del sistema, teniendo en cuenta que se realizó una API la cual tiene implementada sus propios modelos y controladores y para un desarrollo organizado de la aplicación esta se divide en dos paquetes: el cliente y el servidor. En el paquete del cliente se encuentran varios paquetes que componen el framework Angular JS y que se utilizaron en el proyecto y además contiene el paquete llamado pluviopivotApp que es el que contiene a su vez las funcionalidades de la aplicación. Por su parte el paquete del servidor está compuesto por el paquete Express (framework que permite crear los servicios web) y el paquete que agrupa la API de la aplicación, es decir, donde se encuentran los diferentes servicios para

interactuar con el cliente. A continuación, se presenta la Figura 4 donde se representa gráficamente lo explicado anteriormente.

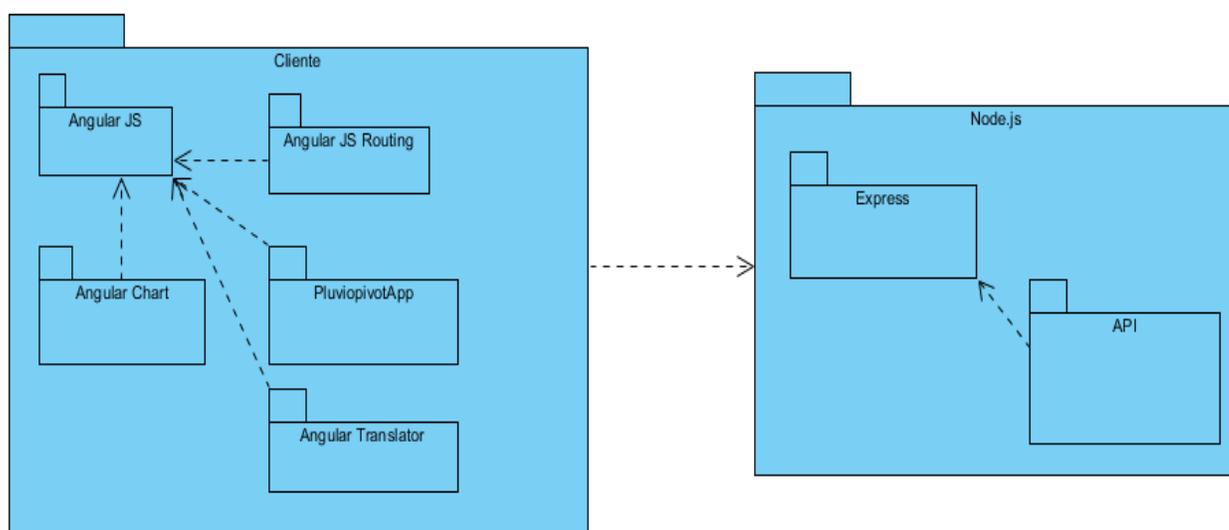


Figura 4 Diagrama de paquetes

2.6 Diagrama de clases

Para comprender el diagrama de clases de la aplicación primeramente tenemos que hablar sobre el patrón que se usa en Angular, el conocido Modelo, Vista, Controlador.

- **Vistas:** Será el HTML y todo lo que represente datos o información.
- **Controladores:** Se encargarán de la lógica de la aplicación y sobre todo de las llamadas "Factorías" y "Servicios" para mover datos contra servidores o memoria local en HTML5.
- **Modelo de la vista:** En Angular el "Modelo" es algo más de aquello que se entiende habitualmente cuando se habla del MVC tradicional, o sea, las vistas son algo más que el modelo de datos. A modo de ejemplo, en aplicaciones de negocio donde se tiene que manejar la contabilidad de una empresa, el modelo serían los movimientos contables. Pero en una pantalla concreta de la aplicación es posible que se tenga que ver otras cosas, además del movimiento contable, como el nombre de los usuarios, los permisos que tienen, si pueden ver los datos, editarlos, etc. Toda esa información, que es útil para el programador pero que no forma parte del modelo del negocio, es a lo que se le llama el "Scope" que es el modelo en Angular.

Debido a que en Angular se tienen unos modelos un poco diferentes, algunos autores dicen que el patrón que utiliza el framework es el **MVVM** (Model-View-View-Model).

En la siguiente figura se muestra una descripción general de como se trata el MVC en Angular (véase la Figura 5).



Figura 5: MVC en Angular

Una vez explicadas las particularidades del MVC para el framework Angular JS en esta sección se detalla de manera más específica el funcionamiento interno de la aplicación mostrando la relación existente entre las distintas clases que forman los componentes de la aplicación mediante el diagrama de clases.

Como se presentó en el epígrafe anterior la aplicación está compuesta por dos paquetes principales el del cliente y el del servidor. Con el fin de lograr un mayor entendimiento del diagrama de clases se mostrará el diagrama de clases contenidas en cada uno de los paquetes por separado.

Los diagramas de clases se utilizan para modelar la vista estática de diseño de un sistema. Esta vista soporta principalmente los requisitos funcionales de un sistema, los servicios que el sistema debe proporcionar a sus usuarios finales (Grady Booch, 2005).

La Figura 6 muestra las clases existentes en el paquete Cliente del diagrama de paquetes y la relación existente entre ellas.

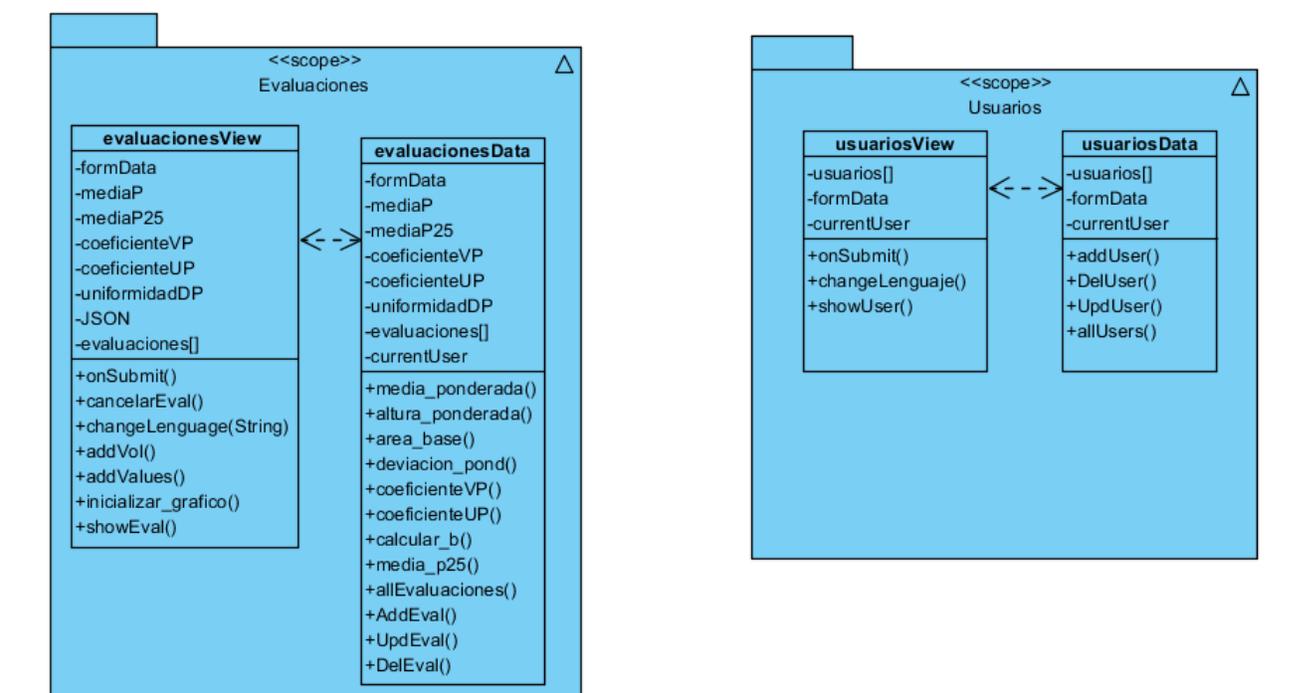


Figura 6: Diagrama de clases del Cliente

El paquete cliente sigue el patrón MVC usado en Angular JS y explicado anteriormente y además dicha aplicación está construida sobre la base de un SPA (Single Page Application). Como se explicó anteriormente en Angular cada vista está asociada a un controlador y viceversa, estas se comunican a través de un <<Scope>> que es el modelo en Angular. Es importante mencionar que en JavaScript no existen clases por lo que lo más parecido a una es cada Scope asociado a las vistas y controladores existentes. En el diagrama anterior se representa el Scope de cada vista el cual se subdividió en dos partes para un mejor entendimiento del contenido del mismo. Las clases EvaluacionesView, UsuariosView y RolesView representan los datos y operaciones de las vistas y las restantes clases representan los datos y operaciones del negocio

La Figura 7 muestra las clases relacionadas con la API, contenidas en el paquete Servidor del diagrama de paquetes.

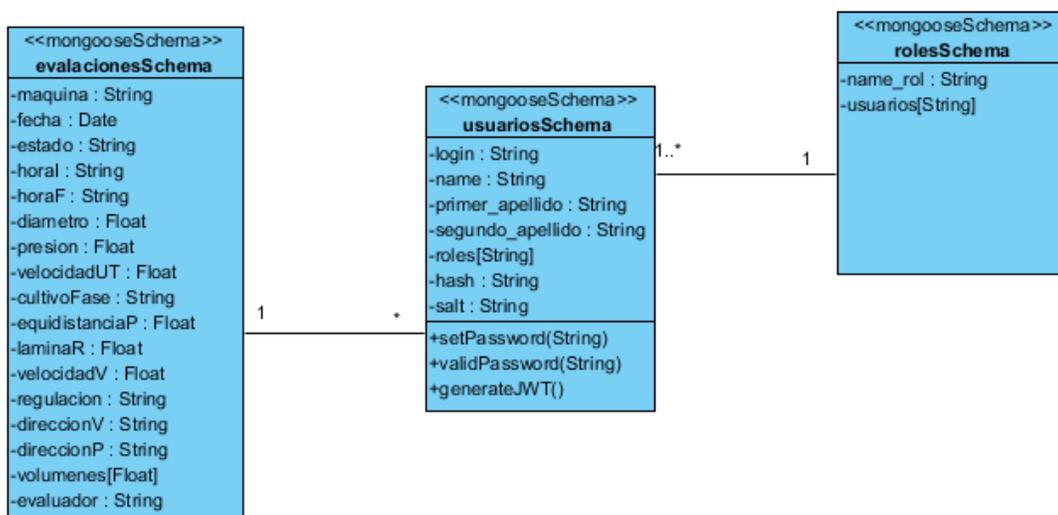


Figura 7: Diagrama de clases del Servidor

La API REST está compuesta por las clases evaluacionesSchema, usuariosSchema y rolesSchema que constituyen los modelos mongoose de las respectivas colecciones de la base de datos. En ellas se define la estructura de los documentos de dichas colecciones así como los tipos de datos y las operaciones definidas para el modelo.

2.7 Diagrama de componentes

Un diagrama de componente muestra la organización y las dependencias entre un conjunto de componentes. Los diagramas de componentes se utilizan para modelar la vista de implementación estática de un sistema (Grady Booch, 2005).

A continuación, se muestra la arquitectura de la aplicación a través del diagrama de componentes del sistema (véase la Figura 8).

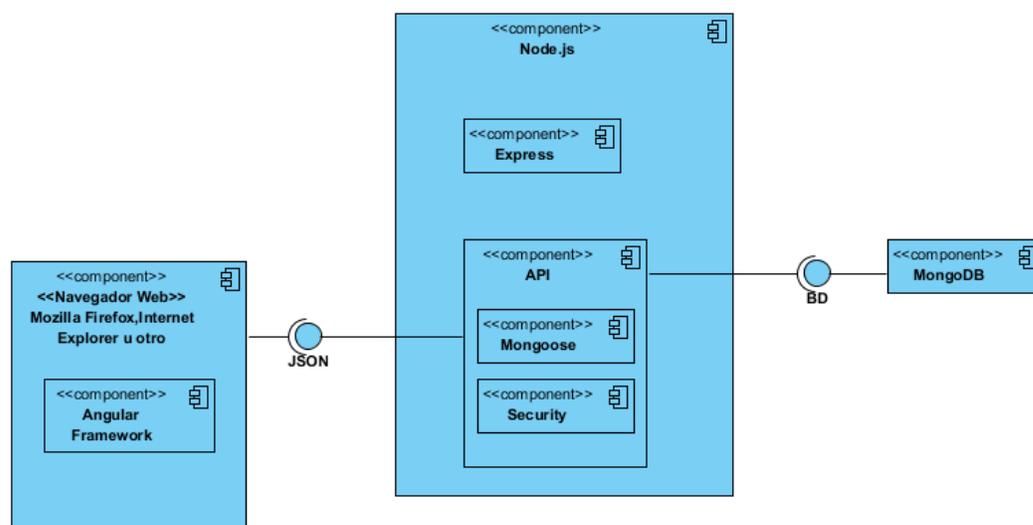


Figura 8: Diagrama de componentes del sistema

Inicialmente el usuario solicita a través del navegador visitar el sitio Pluviopivot enviando el navegador la solicitud al servidor, posteriormente el servidor envía el sitio hacia el navegador y la aplicación procesa la solicitud y renderiza la página pudiendo el usuario visualizar la página correcta. Luego de iniciado el sitio si el jefe solicita insertar una nueva evaluación en el sistema esta solicitud es procesada por el navegador a través del módulo de angular ngRoute el cual enruta la petición hacia el controlador de la aplicación quien a su vez a través del servicio angular correspondiente realiza una llamada AJAX a la API la cual a su vez realiza llamadas GET o POST a la Base de datos devolviendo un json al servidor Node.js con Express y este a su vez envía este JSON al navegador el cual Angular procesa y finalmente enruta la página html correspondiente hacia el SPA (véase Figura 9).

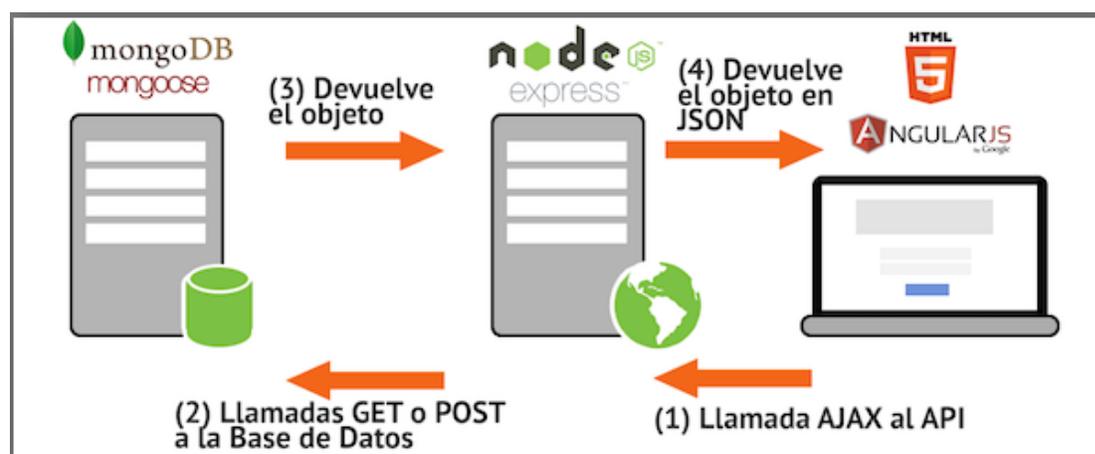


Figura 9: Esquema de la arquitectura y proceso de la aplicación web

2.8 Diagrama de despliegue

Un diagrama de despliegue muestra la configuración de nodos que participan en la ejecución y de los componentes que residen en ellos. Los diagramas de despliegue se utilizan para modelar la vista de despliegue estática de un sistema. La mayoría de las veces, esto implica modelar la topología del hardware sobre el que se ejecuta el sistema. Los diagramas de despliegue fundamentalmente diagramas de clases que se ocupan de modelar los nodos de un sistema (Grady Booch, 2005).

A continuación, se muestra el diagrama de despliegue del sistema (véase la Figura 10).

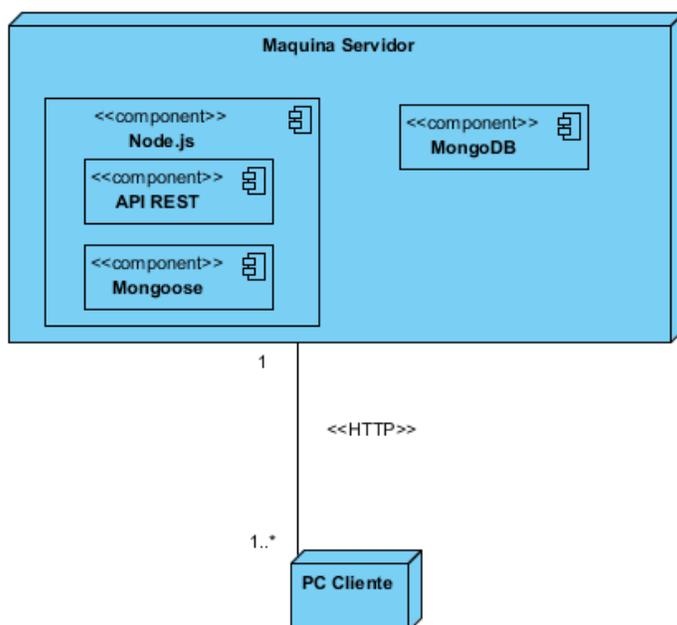


Figura 10: Diagrama de despliegue del sistema

Como se puede apreciar en el diagrama anterior mediante el protocolo <<HTTP>> se establece una conexión de uno o varios clientes hacia una PC que actúa como servidor donde corren el servidor Node.js junto a Express, la API que brinda los servicios web y el gestor de bases de datos MongoDB.

2.9 Maqueta del sitio web

Una maqueta de sitio web es un grupo de imágenes y bloques que definen la estructura visual de las principales páginas que conforma el sitio web. Es realizada generalmente por un diseñador web. Los desarrolladores de software utilizan dicha maqueta como pauta para implementar el sitio web mediante HTML, CSS, JavaScript, PHP o cualquier otro lenguaje de programación web (Beaird 2007).

Dentro de un sitio web existen varios componentes y bloques que se incluyen dentro de la maqueta del sitio y son utilizados en dependencia de las funcionalidades y propósitos del sitio. Algunos de estos componentes se describen a continuación:

- **Bloque Contenedor:** Es el componente que contiene todos los elementos de la página tales como la navegación, tablas, párrafos etc sin él no tendríamos lugar donde poner el contenido de la página. El ancho puede ser líquido, es decir, se expande para llenar la ventana del navegador, o fijo, de modo que el contenido es del mismo ancho no importa el tamaño de la ventana.
- **Logo:** Cuando los diseñadores se refieren a una identidad, se refieren al logotipo y colores que existen a través de las diversas formas de comercialización de una empresa, tales como tarjetas de visita, membretes, folletos, etc. El bloque de identidad que aparece en el sitio web debe contener el logotipo de la empresa o nombre y ubicarse en la parte superior de cada página del sitio web. El bloque de identidad aumenta la marca y permite a los usuarios saber que las páginas que están viendo son parte de un solo sitio.
- **Navegación:** Debe ser fácil de encontrar y usar. Se espera que aparezca en la parte superior de la página. Debe ser visible en la página sin hacer scroll.
- **Contenido:** Es lo principal de un sitio web. Debe ser fácil de encontrar y de entender, puede ser líquido o fijo. Debe ser lo suficientemente atractivo y entendible para que los visitantes no abandonen la página.
- **Pie de página:** Se ubica en la parte inferior de la página. Contiene derechos de autor, información legal, así como algunos enlaces a las secciones principales del sitio. Debe indicar a los usuarios que están en la parte inferior de la página.
- **Espacio en blanco:** Se refiere literalmente a cualquier área de una página que no está cubierta por el tipo o las ilustraciones. Tener espacio vacío en una página es tan importante como tener contenido pues ayuda a un diseño a "respirar" guiando el ojo del usuario alrededor de una página, ayuda a crear el equilibrio y la unidad.

En la siguiente figura se muestra la maqueta del sitio web a desarrollar, donde están presentes algunos de los componentes y bloques mencionados anteriormente (véase la Figura 11).

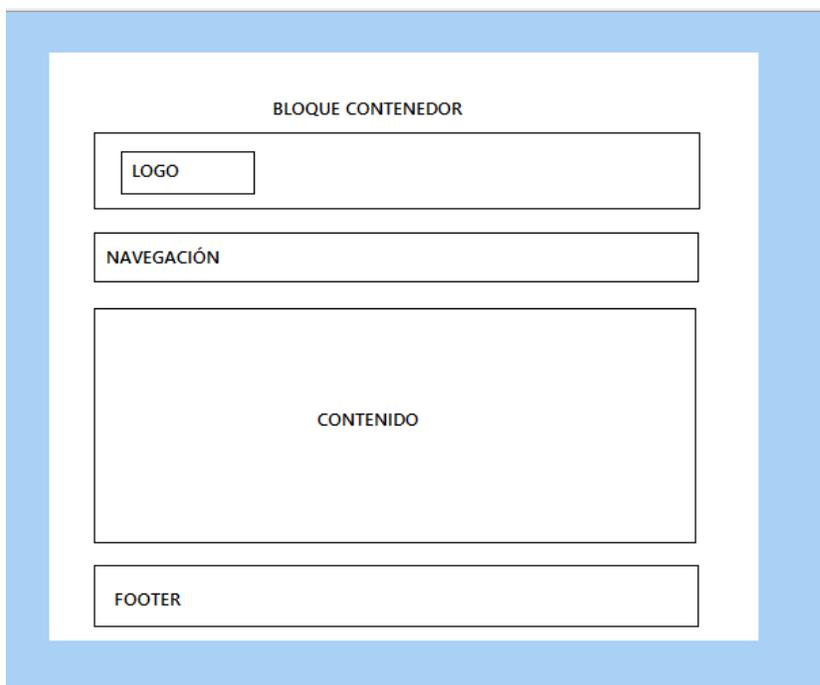


Figura 11:Maqueta del sitio web

2.10 Diagrama de navegación

Los mapas de navegación proporcionan una representación esquemática de la estructura del hipertexto, indicando los principales conceptos incluidos en el espacio de la información y las interrelaciones que existen entre ellos. Un mapa es, por ejemplo, una representación completa (o resumida) del sitio *web* para orientar al lector/usuario durante el recorrido o para facilitarle un acceso directo al lugar que le interese. Reflejará la estructura del *web* por medio de enlaces a los nodos principales, y éstos también pueden desarrollarse para mostrar los subnodos.

El mapa de navegación puede representarse bien en forma textual, bien en forma gráfica, o una combinación de ambas. De manera similar a los sumarios, índices y tablas de contenido que sirven para "navegar" por los documentos en papel, estas herramientas también son útiles para navegar por un hipertexto. En la Web, estos mecanismos tradicionales pueden tomar la forma de menús (Kenguan 2015).

A continuación, se muestra el mapa de navegación del sitio web para la evaluación de las máquinas de riego de pivote central (véase la Figura 12).

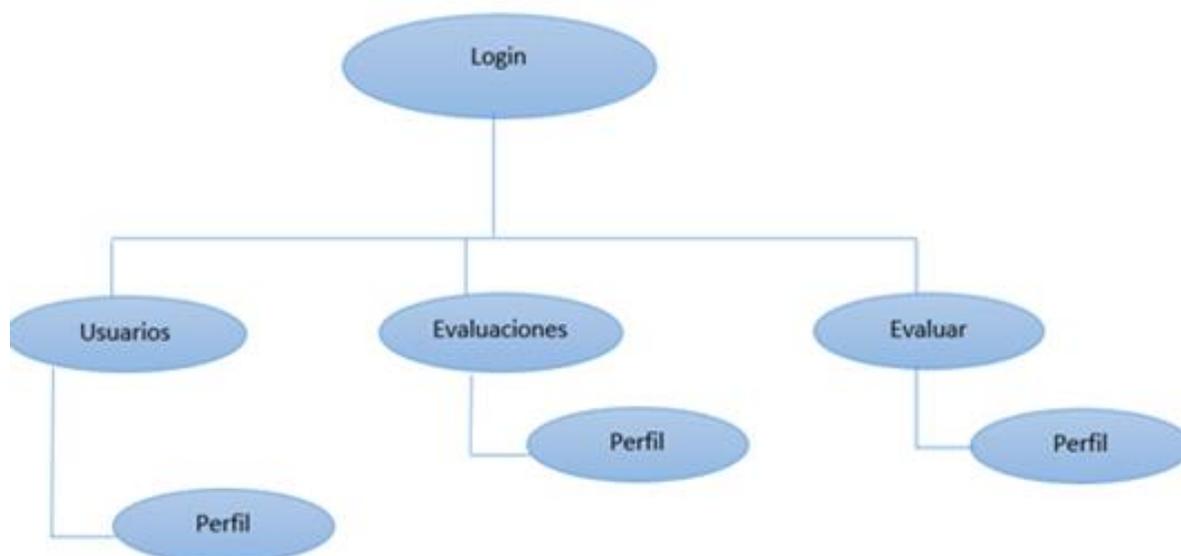


Figura 12:Diagrama de navegación para el sistema de evaluación de máquinas de riego de pivote central

2.11 Conclusiones parciales

El presente capítulo estuvo encaminado hacia el análisis de los principales componentes del sistema y el diseño de cada una de sus partes a través de diagramas y esquemas que permitieron una mejor comprensión de la solución computacional. Se realizó la descripción del proceso de evaluación de máquinas de riego de pivote central, así como la modelación del diagrama de casos de uso, la estructura de la base de datos no relacional del sistema, el diagrama de paquetes, de clases, de componentes y de despliegue, así como la maqueta y el diagrama de navegación del sistema.

CAPÍTULO 3. PRUEBAS SOBRE LA APLICACIÓN PARA LA EVALUACIÓN DE MÁQUINAS DE RIEGO DE PIVOTE CENTRAL.

Las pruebas de software (Software Testing) comprenden el conjunto de actividades que se realizan para identificar posibles fallos de funcionamiento, configuración o usabilidad de un programa o aplicación, por medio de pruebas sobre el comportamiento del mismo. Los sistemas informáticos, programas y aplicaciones han crecido a niveles inimaginables en complejidad e interoperabilidad, con lo cual también se han incrementado las posibilidades de defectos (bugs), a simple vista insignificantes, pero que pudieran adquirir proporciones catastróficas. Además, factores como el uso de software de terceros desde aplicaciones móviles han añadido niveles adicionales de complejidad y por ende incrementado los posibles puntos de fallas.

Las pruebas en el software garantizan la calidad del producto validando que cumpla las especificaciones para las que fue diseñado. Existen diferentes tipos de pruebas, cada una con objetivos específicos y cuya responsabilidad recae sobre diferentes roles. Entre estos tipos de pruebas se encuentran las pruebas de caja negra que, a través de la entrada de datos, tratan de encontrar fallas. Otro de los métodos de prueba que existe es el de caja blanca que se centra en los detalles de la implementación del software, por lo que su diseño está fuertemente ligado al código fuente de la aplicación.

En el presente capítulo se describen los tipos de prueba mencionados anteriormente, así como su aplicación sobre el sistema desarrollado. Se muestra además la descripción de las pruebas, los posibles errores y los resultados obtenidos.

3.1 Pruebas de caja blanca

Las pruebas de caja blanca se basan en el conocimiento de la lógica interna del código del sistema. Las pruebas completan los distintos caminos que se pueden generar gracias a las estructuras condicionales, a los distintos estados del mismo, etc. Al estar basadas en una implementación concreta, si ésta se modifica, por regla general las pruebas también deberán rediseñarse (Pressman 2008).

Entre los tipos de pruebas de caja blanca se encuentran las pruebas unitarias, las cuales son diseñadas para validar que una unidad de código, ya sea una clase, un método o función, realice el trabajo para el que fue diseñado correctamente.

Para realizar las pruebas unitarias con AngularJS y Node.js se utiliza la herramienta KarmaJS. KarmaJS es un lanzador de test, desarrollado por el equipo de angular, que nos permite automatizar algunas tareas de los frameworks de tests, como Jasmine. Al igual que ocurre con AngularJS, se parte de una instalación básica muy sencilla que luego se va complicando según las necesidades particulares con plugins que cubren necesidades específicas.

En el presente epígrafe se describirá el funcionamiento y utilización de las pruebas unitarias en AngularJS específicamente en el sitio web Pluviopivot.

3.1.1 Pruebas unitarias

Las pruebas unitarias consisten en la ejecución de actividades que le permitan verificar al desarrollador que los componentes unitarios están codificados bajo condiciones de robustez, esto es, soportando el ingreso de datos erróneos o inesperados y demostrando así la capacidad de tratar errores de manera controlada. Adicionalmente, las pruebas sobre componentes unitarios, suelen denominarse pruebas de módulos o pruebas de clases, siendo la convención definida por el lenguaje de programación la que influye en el término a utilizar (Sánchez 2013).

Por convención en el presente trabajo cada prueba unitaria en AngularJS se define en un documento JavaScript cuyo nombre termina en `_test` y se encuentra en la raíz del proyecto dentro de una carpeta llamada **tests**.

Cuando se lanzan los test si se produce un error, karmaJS emite una salida por la consola donde muestra el texto **FAILED** como resumen de la ejecución, indicando para cada error el archivo y el método erróneo, el mensaje propio que se incluyó en la prueba e información adicional como el valor esperado y el valor obtenido.

Por el contrario, si la prueba pasa satisfactoriamente se muestra en la consola el texto **SUCCESS** indicando de esta manera que no existió ningún error en el lanzamiento de los tests (véase Figura 13).

```
D:\uclv\Tesis_Project\pluviopivot_web>npm test
> project@0.0.0 test D:\uclv\Tesis_Project\pluviopivot_web
> karma start karma.conf.js --reporters progress,junit,coverage
11 06 2017 14:23:16.028:INFO [karma]: Karma v1.7.0 server started at http://0.0.0.0:9876/
11 06 2017 14:23:16.039:INFO [launcher]: Launching browser Firefox with unlimited concurrency
11 06 2017 14:23:16.076:INFO [launcher]: Starting browser Firefox
11 06 2017 14:23:22.557:INFO [Firefox 40.0.0 (Windows 10 0.0.0)]: Connected on socket PrQX8JWGRUjvlvSMAAAA with id 7373094
Firefox 40.0.0 (Windows 10 0.0.0): Executed 9 of 9 SUCCESS (0.466 secs / 0.149 secs)
D:\uclv\Tesis_Project\pluviopivot_web>
```

Figura 13: Resultado de las pruebas de la aplicación

Las pruebas unitarias permiten comprobar el correcto funcionamiento de las reglas de validación de las entidades de la aplicación. A continuación, se muestra parte del código que comprueba la validación de la función `media ponderada ()` (véase Figura 14) y de la función `coeficiente_variacion_ponderado ()` (véase la Figura 15).

```
describe("probando funcion media_ponderada()", function()
{
  beforeEach(module("app"));

  describe("app", function()
  {
    var scope, ctrl;

    beforeEach(inject(function($rootScope, $controller)
    {
      scope = $rootScope.$new();
      ctrl = $controller("mediaPCtrl", {$scope:scope});
    }));

    it("debe estar definida la variable radio", function()
    {
      expect(scope.radio).toBeDefined();
    })
    it("deben estar definidos los volúmenes", function()
    {
      expect(scope.volumenes).toBeDefined();
    })
    it("deben estar definidas las alturas ponderadas", function()
    {
      expect(scope.alturasP).toBeDefined();
    })
    it("el valor de la funcion debe ser 62", function()
    {
      scope.volumenes=[100,130,140];
      scope.radio=7.5;
      scope.diametro=15;
    })
  })
});
```

Figura 14: Validación de la función *media_ponderada()*

```
describe("probando funcion coeficiente_variacion_ponderado", function()
{
  beforeEach(module("app1"));

  describe("app1", function()
  {
    var scope, ctrl;

    beforeEach(inject(function($rootScope, $controller)
    {
      scope = $rootScope.$new();
      ctrl = $controller("coefDPCtrl", {$scope:scope});
    }));

    it("debe estar definida la variable radio", function()
    {
      expect(scope.radio).toBeDefined();
    })
    it("deben estar definidos los volúmenes", function()
    {
      expect(scope.volumenes).toBeDefined();
    })
    it("deben estar definidas las alturas ponderadas", function()
    {
      expect(scope.alturasP).toBeDefined();
    })
    it("deben estar calculada la media ponderada y la desviacion_tipica_ponderada", function()
    {
      scope.volumenes=[100,130,140];
      scope.radio=7.5;
      scope.diametro=15;
    })
  })
});
```

Figura 15: Validación de la función *coeficiente_variacion_ponderado()*

Luego de probar los test de la aplicación gracias al soporte que brinda karma se pueden generar informes con la información relativa a la ejecución de los test. Por defecto Karma nos proporciona un reporter al que llama **progress**, que no es más que la salida por consola que nos indica qué ha sucedido con los tests. En el proyecto se añadieron dos más:

- **junit**: Generará un fichero .xml en formato junit. Este informe está especialmente indicado cuando se quiere que sea consumido por otras herramientas como **Jenkins o Bamboo**.
- **coverage**: Genera informes visuales (html) mostrando el porcentaje de código cubierto por los tests. Además, genera un fichero **lcov** especialmente indicado si se quiere exportar el informe a alguna herramienta tipo **Sonar**.

En la siguiente figura se muestra el informe generado por el reporter coverage luego de lanzar los test.

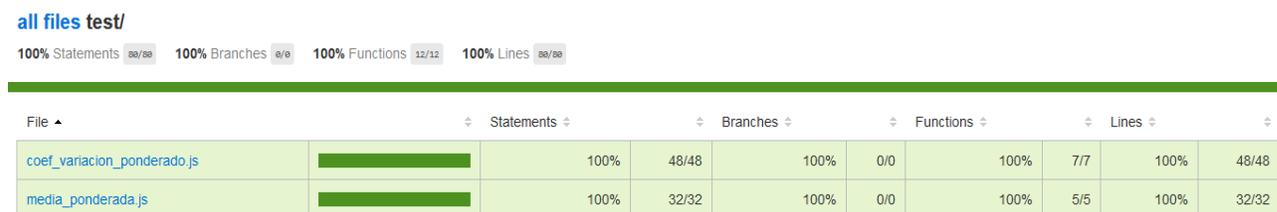


Figura 16: Informe generado con el reporter Coverage

Como resultado de las pruebas unitarias realizadas a la aplicación se comprobó el correcto funcionamiento de la misma pues los test fueron realizados reportando una salida satisfactoria y como se puede apreciar en la Figura 16 en ambos test realizados se comprobó el 100% de las sentencias y funciones, garantizando de esta manera el correcto funcionamiento de las funciones chequeadas.

3.2 Pruebas de caja negra

Las pruebas de caja negra, también denominadas pruebas de comportamiento, se centran en los requisitos funcionales del software. O sea, la prueba de caja negra permite al ingeniero del software obtener conjuntos de condiciones de entrada que comprueben completamente todos los requisitos funcionales de un sistema (Pressman 2008).

La prueba de caja negra no es una alternativa a las técnicas de prueba de caja blanca. Más bien se trata de un enfoque complementario que intenta descubrir diferentes tipos de errores que los métodos de caja blanca. La prueba de caja negra intenta encontrar errores de las siguientes categorías:

- 1) funciones incorrectas o ausentes
- 2) errores de interfaz

- 3) errores en estructuras de datos o en accesos a bases de datos externas
- 4) errores de rendimiento
- 5) errores de inicialización y de terminación.

A diferencia de la prueba de caja blanca, que se lleva a cabo previamente en el proceso de prueba, la prueba de caja negra tiende a aplicarse durante fases posteriores de la prueba. Ya que la prueba de caja negra ignora intencionadamente la estructura de control, centra su atención en el campo de la información (Pressman 2002).

3.2.1 Pruebas de sistema

En este apartado se describirán las pruebas para comprobar las funcionalidades del sistema implementado. Se mostrará la descripción, pasos, resultado esperado y el resultado obtenido de cada caso de prueba. Se comprobará el funcionamiento de dos de los casos de uso de la aplicación, en particular autenticar usuario y añadir una nueva evaluación (véase Tablas 11,12,13 y 14).

Tabla 11: Descripción del caso de prueba "Autenticar Usuario" (prueba 1)

Caso de prueba:	Autenticar Usuario
ID Caso de prueba	2
Descripción	Se comprobará que un usuario se puede autenticar en el sistema
Pasos	<ol style="list-style-type: none"> 1. Acceder a la página de inicio de la aplicación. 2. Rellenar los campos del formulario de acceso con nombre de usuario y contraseña. 3. Pulsar sobre el botón "Acceder" del formulario. 4. Permitir acceso a la aplicación
Resultado esperado	El usuario accede a la página principal de la aplicación según su rol donde puede comenzar a trabajar
Estado caso de prueba	Ejecutado: ➤ Exitoso
Resultado obtenido	Luego de la ejecución del caso de prueba y de acuerdo a la reacción de la aplicación, se observa que accede a la página de la aplicación según el rol del usuario logueado.

Errores asociados	Ninguno
-------------------	---------

Las figuras 17 y 18 muestran las entradas y salidas del caso de prueba Autenticar Usuario (prueba 1).



Figura 17: Entrada "Autenticar Usuario"(prueba 1).



Figura 18: Salida "Autenticar Usuario"(prueba 1).

Tabla 12: Descripción del caso de prueba "Autenticar Usuario"(prueba 2)

Caso de prueba:	Autenticar Usuario
ID Caso de prueba	2
Descripción	Se comprobará la respuesta del sistema cuando un usuario quiere acceder al sitio pero ingresa el usuario o la contraseña incorrecta.
Pasos	<ol style="list-style-type: none"> 1. Acceder a la página de inicio de la aplicación. 2. Rellenar los campos del formulario de acceso con nombre de usuario y contraseña. 3. Pulsar sobre el botón "Acceder" del formulario. 4. Permitir acceso a la aplicación
Resultado esperado	Se muestre un error diciendo que el usuario o la contraseña son incorrectos
Estado caso de prueba	Ejecutado: ➤ Exitoso
Resultado obtenido	Luego de la ejecución del caso de prueba y de

	acuerdo a la reacción de la aplicación, se observa que se muestra el error esperado.
Errores asociados	Ninguno

Las figuras 17 y 18 muestran las entradas y salidas del caso de prueba Autenticar Usuario (prueba 2).



Figura 19:Entrada "Autenticar Usuario"(prueba 2).



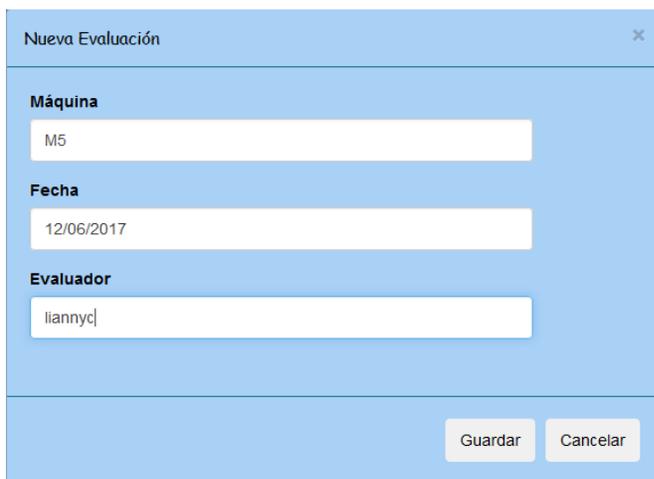
Figura 20: Salida "Autenticar Usuario"(prueba 2).

Tabla 13: Descripción del caso de prueba "Añadir Evaluación"(prueba 1)

Caso de prueba:	Añadir Evaluación
ID Caso de prueba	1
Descripción	Se comprobará que al añadir una nueva evaluación se registre en la base de datos
Pasos	<ol style="list-style-type: none"> 1. Acceder a la página de inicio de la aplicación. 2. Rellenar los campos del formulario de acceso con un nombre de usuario y contraseña que tenga permiso de Jefe. 3. Pulsar sobre el botón "Acceder" del formulario. 4. Pulsar sobre el botón Nuevo y rellenar el formulario
Resultado esperado	Se agregue la evaluación a la base de datos
Estado caso de prueba	Ejecutado: ➤ Exitoso

Resultado obtenido	Luego de la ejecución del caso de prueba y de acuerdo a la reacción de la aplicación, se observa que se agrega la evaluación a la base de datos.
Errores asociados	Ninguno

En las figuras 21 y 22 se muestran las entradas y salidas del caso de prueba Añadir Evaluaciones (prueba 1).



The image shows a web form titled "Nueva Evaluación" with a close button (X) in the top right corner. The form has a light blue background and contains three input fields: "Máquina" with the value "M5", "Fecha" with the value "12/06/2017", and "Evaluador" with the value "lianny". At the bottom right of the form, there are two buttons: "Guardar" and "Cancelar".

Figura 21:Entrada "Añadir Evaluación"(prueba 1).

```

C:\Windows\system32\cmd.exe - mongo.exe

"volumenes" : [
  100,
  130,
  140
],
"_v" : 1,
"coeficiente_uniformidadP" : 92,
"coeficiente_variacionP" : 11.9,
"equidistanciaP" : null,
"laminar" : null,
"media_ponderada" : 7.36,
"media_ponderada25" : 6.79,
"presion" : null,
"uniformidad_distP" : 92.3,
"velocidadUT" : null,
"velocidadV" : null,
"horaF" : "11:00",
"horaI" : "10:00"

" id" : ObjectId("593d9c791d5e73181887d2d2"),
"maquina" : "M5",
"fecha" : ISODate("2017-06-12T04:00:00Z"),
"estado" : "Planificada",
"diametro" : 0,
"evaluador" : "liannyc",
"volumenes" : [ ],
"_v" : 0
    
```

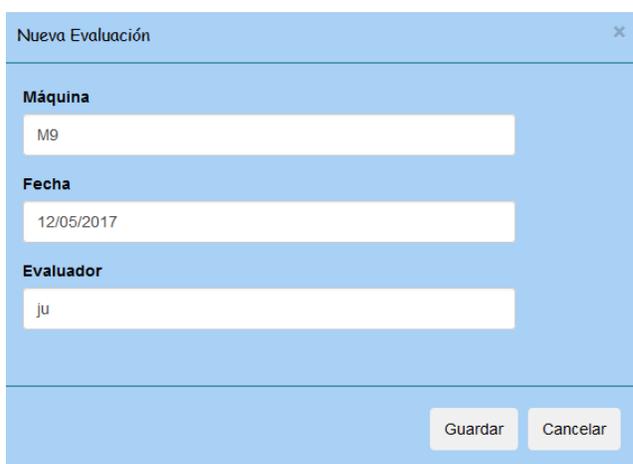
Figura 22:Salida "Añadir Evaluación"(prueba 1).

Tabla 14:Descripción del caso de prueba "Añadir Evaluación"(prueba 2)

Caso de prueba:	Añadir Evaluación
ID Caso de prueba	2
Descripción	Se comprobará que al añadir una nueva evaluación si a esta se le asocia un evaluador que no está registrado en la base de datos no se inserta la evaluación.
Pasos	<ol style="list-style-type: none"> 1. Acceder a la página de inicio de la aplicación. 2. Rellenar los campos del formulario de acceso con un nombre de usuario y contraseña que tenga permiso de Jefe. 3. Pulsar sobre el botón "Acceder" del formulario. 4. Pulsar sobre el botón Nuevo y rellenar el formulario
Resultado esperado	Se muestra un error diciendo que el evaluador

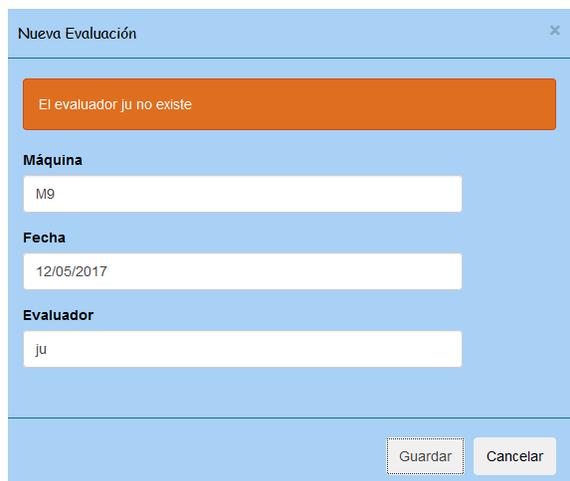
	no está registrado.
Estado caso de prueba	Ejecutado: ➤ Exitoso
Resultado obtenido	Luego de la ejecución del caso de prueba y de acuerdo a la reacción de la aplicación, se observa que se muestra el error esperado
Errores asociados	Ninguno

En las figuras 23 y 24 se muestran las entradas y salidas del caso de prueba Añadir Evaluaciones (prueba 2).



The image shows a web form titled "Nueva Evaluación" with a close button (X) in the top right corner. The form has a light blue background and contains three input fields: "Máquina" with the value "M9", "Fecha" with the value "12/05/2017", and "Evaluador" with the value "ju". At the bottom right of the form, there are two buttons: "Guardar" and "Cancelar".

Figura 23:Entrada "Añadir Evaluación"(prueba 2).



The screenshot shows a web form titled "Nueva Evaluación" with a close button (x) in the top right corner. At the top, there is an orange error message box that reads "El evaluador ju no existe". Below this, there are three input fields: "Máquina" with the value "M9", "Fecha" with the value "12/05/2017", and "Evaluador" with the value "ju". At the bottom of the form, there are two buttons: "Guardar" and "Cancelar".

Figura 24: Salida "Añadir Evaluación"(prueba 2).

Conclusiones parciales

El capítulo estuvo dirigido a la validación de la aplicación mediante los métodos de prueba de caja blanca y caja negra. Se explicaron las características de las pruebas unitarias y de sistema en AngularJS y su aplicación al sistema desarrollado. Se comprobó el correcto funcionamiento de dos de los casos de uso de la aplicación.

CONCLUSIONES

- ❖ Se identificaron mediante una revisión sistemática de la literatura aplicando la técnica del SLR siguiendo la metodología de Kitchenham 11 marcos de trabajo para el desarrollo de aplicaciones web existentes en la literatura científica.
- ❖ Se seleccionó el marco de trabajo MEAN, teniendo en cuenta criterios como la integración con dispositivos móviles, la documentación de la tecnología utilizada, la rapidez del acceso a la base de datos entre otros.
- ❖ Se propuso una arquitectura basada en Angular SPA que utilizó el marco de trabajo seleccionado compuesta por: el gestor de bases de datos MongoDB, la API intermedia construida sobre el servidor Node.js y el framework Express y AngularJS en el cliente.
- ❖ Se implementó la arquitectura propuesta en una nueva versión de Pluviopivot la cual permite la integración con dispositivos móviles.
- ❖ Se analizaron varias funcionalidades de la aplicación web mediante los métodos de caja blanca y de caja negra, obteniéndose para ambos tipos de prueba resultados satisfactorios.

RECOMENDACIONES

- ❖ Analizar la inclusión de un módulo a la aplicación que permita generar informes sobre el proceso de evaluación de máquinas de riego de pivote central.
- ❖ Añadir funcionalidades a la API realizada.

BIBLIOGRAFÍA

- Alimadadi, S., 2016a. Understanding Asynchronous Interactions in Full-Stack JavaScript.
- BANKER, K., VERCH, P.B.S. & HAWKINS, D.G.T., 2016. *MongoDB in action* Second Edi., Available at: <http://dl.acm.org/citation.cfm?id=2207997>.
- Beaird, J., 2007. *The Principles of Beautiful web design*, SitePoint Pty.Ltd.
- Cárdenas, Y.P., Segui, J.P. & Agropecuarias, C., CONTENIDO Introducción Materiales y Métodos Manual de Usuario Bibliografía Introducción Materiales y métodos.
- Caro, D.A. & Bío-bío, U., 2008. Revisiones Sistemáticas de la Literatura.
- Chodorow, K., *MongoDB The Definitive Guide*,
- Chodorow, K., 2013. *MongoDB: the definitive guide* Second Edi.,
- Chugh, R. & Meister, J.A., Staged Information Flow for JavaScript *, pp.50–62.
- Development, W. & Right, D., *The Definitive Guide to Web Development Done Right* Adrian Holovaty,
- German, A. et al., 2016. MEAN Web Development: A Tutorial for Educators., pp.128–129.
- Haviv, A.Q., *MEAN Web Development*,
- Holmes, S., 2013. *Getting MEAN with Mongo, Express, Angular, and Node*.
- I. Jacobson, G. Booch, J.R., 2000. *El Proceso Unificado de Desarrollo de Software*,
- Kenguan, 2015. ¿Qué es mapa de navegación?
- Kitchenham, B. & Ebse, C., 2007. Guidelines for performing Systematic Literature Reviews in Software Engineering Executive summary.
- Poulter, A.J., Johnston, S.J. & Cox, S.J., 2015. Using the MEAN Stack to Implement a RESTful Service for an Internet of Things Application.
- Pressman, R.S., 2008. *Ingeniería del Software. Un enfoque práctico* 6ta Edición.,

- Pressman, R.S., 2002. *Ingeniería del Software. Un enfoque práctico* 5ta Edició. C. F. Madrid, ed., Madrid, España.
- Sánchez, J.Z., 2013. PRUEBAS DE SOFTWARE _ Ingeniería de Software con énfasis en pruebas. Available at: <https://pruebasdelsoftware.wordpress.com/>
- Štajcer, M., 2016. Using MEAN stack for development of GUI in real-time big data architecture., pp.524–529.
- Tilkov, S. & Verivue, S.V., 2010. Node. js: Using JavaScript to Build High-Performance Network Programs.
- Vepsäläinen, J., 2011. Overview of JavaScript Application Development., pp.1–35.

ANEXOS

Anexo I Manual de usuario para el sistema informático para la evaluación de máquinas de riego de pivote central.

El sistema informático para la evaluación de las máquinas de riego es una aplicación Web que permite gestionar la información sobre la evaluación pluviométrica, permitiendo visualizar ciertos parámetros requeridos para observar la eficiencia de las máquinas de riego, como la media ponderada, el coeficiente de uniformidad ponderado, entre otros. Además, también grafica el comportamiento de las mismas. El sistema permite al usuario administrador gestionar los diferentes usuarios que pueden acceder a la aplicación y sus privilegios dentro del sistema. Este anexo está dedicado a explicar detalladamente el uso de esta herramienta para lograr una mayor familiarización de los usuarios con el sistema.

Formulario de acceso

Después de acceder a través del navegador web a la dirección donde se encuentra la aplicación, se activa el mecanismo de seguridad apareciendo un formulario para la autenticación del usuario (véase Figura 25).

Figura 25: Formulario de acceso a la aplicación

La siguiente tabla muestra la relación de cada elemento del formulario y el número correspondiente (Tabla 15).

Tabla 15: Relación de cada elemento del formulario de acceso a la aplicación y el número correspondiente

Número	Elemento
1	Campo texto para introducir el nombre de usuario
2	Campo texto para introducir la contraseña

3	Botón de envío del formulario que permite el acceso a la aplicación
---	---

Página principal de la aplicación (Rol Administrador).

Al autenticarse a través del formulario de acceso con el permiso de Administrador se muestra la página principal de la aplicación donde se puede comenzar a navegar por los diferentes menús de navegación permitidos para este Rol, así como gestionar el perfil del usuario logueado, lo mismo sucede con los roles de Jefe y Evaluador (véase Figura 26,27,28).

Pluviopivot
Sistema para la evaluación de máquinas de riego de pivote central

Usuarios Perfil Español Inglés Salir 1

Usuarios del Sistema 2

Nombre y Apellidos	
<input type="checkbox"/> Ernesto Rodriguez Rodriguez	
<input type="checkbox"/> Lisandra Cardoso Morales	
<input type="checkbox"/> Lianny Cardoso Morales	
<input type="checkbox"/> Yunet Luis Ruiz	

Nuevo 4 Eliminar 5

Datos del Usuario

Nombre: Ernesto 3
Primer Apellido: Rodriguez
Segundo Apellido: Rodriguez
Usuario: errodriguez
Roles:
• Jefe

Restablecer Contraseña 6
Introduce tu nueva contraseña

Cambiar 7

Sobre Nosotros
La empresa agricola Valle del Yabú se encuentra ubicada al norte de la ciudad de Santa Clara, provisto de suelos pardos de categoría uno y dos.
La misma se dedica a la producción de cultivos varios y la ganadería, donde se integran todas las formas productivas que funcionan en la agricultura cubana actualmente (unidades empresariales de base (UEB), unidades de prestación de servicios, cooperativa de producción agropecuaria (CPA), unidades básicas de producción cooperativa (Ubpc) y cooperativas de créditos y servicios (CCS) y productores independientes).

Contáctenos
Dirección:
Teléfono: +42 205635 8
Correo: lcmorales@nauta.cu

Figura 26: Página principal de la aplicación (Rol Administrador)



Pluviopivot

Sistema para la evaluación de máquinas de riego de pivote central

🇪🇸 Español 🇬🇧 Inglés [Salir](#)

Evaluaciones

Perfil

1

Evaluaciones

	Máquina	Fecha	Estado	Evaluador	
<input type="checkbox"/>	M4	07/06/2017	Planificada	ylruiz	
<input type="checkbox"/>	M6	04/06/2017	Planificada	ylruiz	
<input type="checkbox"/>	M12	03/06/2017	Iniciada	liannyc	
<input type="checkbox"/>	Fregat 1	17/06/2017	Planificada	liannyc	
<input type="checkbox"/>	Cpo 41 Pirey	15/06/2017	Terminada	liannyc	

Nuevo

Eliminar

Datos de las evaluaciones

Volúmenes(cm3)

Máquina
M4

Hora Inicio

Diámetro

Velocidad de la última torre

Dirección de pluviómetros

Hora Fin

Presión

Lámina de riego requerida

Velocidad del viento

Fecha
07/06/2017

Cultivo y Fase

Equidistancia entre pluviómetros

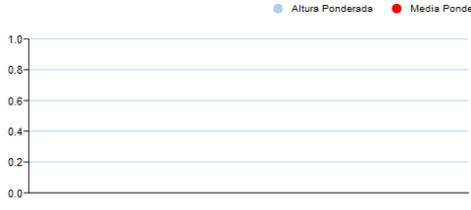
Media Ponderada

Media Ponderada del 25% menos regado

Coefficiente de variación ponderado

Coefficiente de uniformidad ponderado

Uniformidad de la distribución ponderada



● Altura Ponderada
 ● Media Ponderada

Sobre Nosotros

La empresa agrícola Valle del Yabú se encuentra ubicada al norte de la ciudad de Santa Clara, provisto de suelos pardos de categoría uno y dos.

La misma se dedica a la producción de cultivos varios y la ganadería, donde se integran todas las formas productivas que funcionan en la agricultura cubana actualmente (unidades empresariales de base (UEB), unidades de prestación de servicios, cooperativa de producción agropecuaria (CPA), unidades básicas de

Contáctenos

Dirección:

Teléfono: +42 205635

Correo: lcmorales@nauta.cu

Figura 27: Página principal de la aplicación (Rol Jefe)

Pluviopivot
Sistema para la evaluación de máquinas de riego de pivote central

Evaluar Perfil Español Inglés Salir

Evaluar

Máquina	Fecha	Estado
M12	03/06/2017	Iniciada
Fregat 1	17/06/2017	Planificada
Cpo 41 Pirey	15/06/2017	Terminada

Datos de las evaluaciones

Máquina: M12
Fecha: 03/06/2017
Cultivo y Fase:

Hora Inicio: 10:00 Hora Fin: 11:00

Diámetro: 15 Presión: 5 Equidistancia entre pluviómetros: 9

Velocidad de la última torre: m/h Lámina de riego requerida: mm

Dirección de pluviómetros: Velocidad del viento: m/s

Coefficiente de variación ponderado 57.7 %
Coefficiente de uniformidad ponderado 56 %
Uniformidad de la distribución ponderada 66.5 %

Altura Ponderada (blue line) vs Media Ponderada (red line) graph.

Evaluación Terminada

Guardar Cancelar

Sobre Nosotros Contáctenos

La empresa agrícola Valle del Yabú se encuentra ubicada al norte de la ciudad de Santa Clara, provisto de suelos pardos de categoría uno y dos.
La misma se dedica a la producción de cultivos varios y la ganadería, donde se integran todas las formas productivas que funcionan en la agricultura cubana.

Dirección:
Teléfono: +42 205635
Correo: lemorales@nauta.cu

Figura 28: Página principal de la aplicación (Rol Evaluador)

A continuación, se muestran las tablas con la relación entre cada elemento de las páginas principales y el número correspondiente (Tabla 16, 17 y 18).

Tabla 16: Relación entre cada elemento de la página principal (Administrador) y el número correspondiente

Número	Elemento
--------	----------

1	Barra de navegación
2	Tabla que lista los usuarios del sistema
3	Panel con los datos del usuario seleccionado
4	Botón para agregar un Nuevo usuario
5	Botón para eliminar los usuarios seleccionados
6	Input para introducir la nueva contraseña
7	Botón para cambiar la contraseña
8	Pie de página

Tabla 17: Relación entre cada elemento de la página principal(Jefe) y el número correspondiente

Número	Elemento
1	Barra de navegación
2	Tabla que lista las evaluaciones del mes
3	Formulario que muestra los datos de la evaluación seleccionada
4	Botón para agregar una nueva evaluación
5	Botón para eliminar las evaluaciones seleccionadas
6	Gráfico que muestra el comportamiento de la media ponderada y las alturas ponderadas en una evaluación.
7	Pie de página

Tabla 18: Relación entre cada elemento de la página principal(Evaluador) y el número correspondiente

Número	Elemento
1	Barra de navegación
2	Tabla que lista las evaluaciones del mes a realizar por el evaluador
3	Formulario que muestra los datos de la evaluación seleccionada

4	Gráfico que muestra el comportamiento de la media ponderada y las alturas ponderadas en una evaluación.
5	Ccheckbox para indicar que una evaluación ha sido terminada
6	Botón para guardar los cambios realizados en la evaluación
7	Botón para cancelar los cambios realizados en la evaluación
8	Pie de página

Barra de navegación

Varía en dependencia del usuario logueado. Pero de manera general está compuesta por un menú de navegación (parte variable) situado a la izquierda de la barra y un menú de selección de idioma junto con el enlace para salir del sitio situado a la derecha de la barra (véase Figura 29).

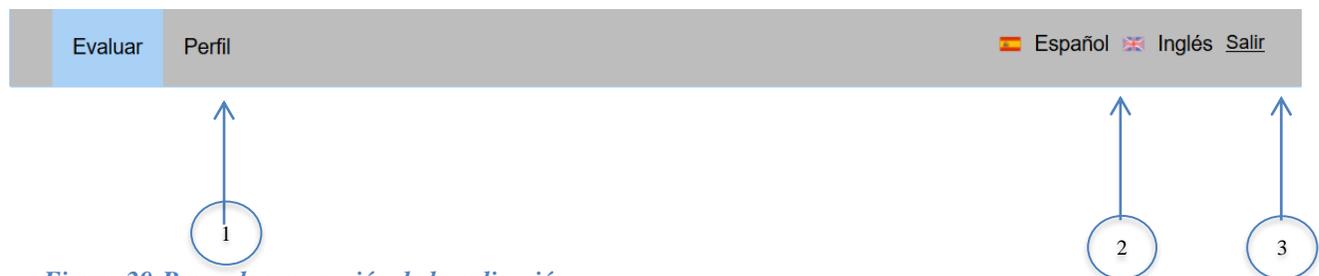


Figura 29: Barra de navegación de la aplicación

La siguiente tabla muestra la relación entre cada elemento del menú de navegación y el número correspondiente (Tabla 19).

Tabla 19: Relación entre cada elemento de la barra de navegación y el número correspondiente

Número	Elemento
1	Menú de navegación
2	Botones para seleccionar idioma
3	Enlace para cerrar la sesión

Gestionar usuarios

Gestionar usuarios constituye una de las áreas de trabajo de la aplicación y se encuentra ubicada en la página principal del rol Administrador. En esta área se pueden introducir nuevos usuarios en la base de datos, editar o eliminar uno existente, así como otras opciones de restablecer contraseña o mostrar todos los datos de uno seleccionado. (véase Figura 30).

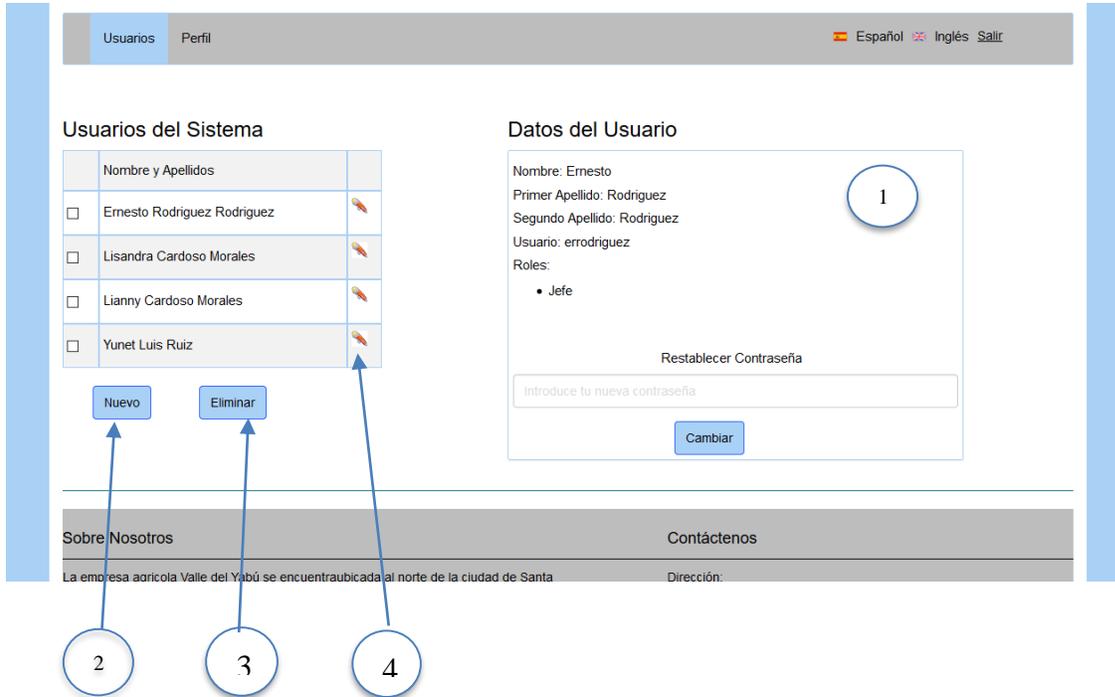


Figura 30: Gestionar Usuarios

La siguiente tabla muestra la relación entre cada elemento del área de trabajo gestionar usuarios y el número correspondiente (Tabla 20).

Tabla 20: Relación entre cada elemento del área de trabajo Gestionar Usuarios y el número correspondiente

Número	Elemento
1	Formulario donde se muestran los datos del usuario seleccionado
2	Botón para agregar un nuevo usuario
3	Botón para eliminar los usuarios seleccionados
4	Botón para actualizar el usuario seleccionado

Formulario de inserción de nuevo usuario

El formulario para insertar un nuevo usuario se muestra después de pulsar en el botón Nuevo de la página principal. Este formulario antes de insertar en la base de datos valida cada uno de

sus campos. No pueden quedar campos en blanco. Si el tipo de dato introducido en alguno de los campos no es correcto, el sistema muestra un mensaje de error indicando el problema (véase Figura 31).

Figura 31:Formulario para adicionar un nuevo usuario

La siguiente tabla muestra la relación entre cada elemento del formulario para adicionar un nuevo usuario y su número (véase Tabla 21).

Tabla 21:Relación entre cada elemento del formulario Insertar Usuario y su número correspondiente

Número	Elemento
1	Campo texto para introducir el nombre
2	Campo texto para introducir el primer apellido
3	Campo texto para introducir el segundo apellido
4	Campo texto para introducir el usuario
5	Campo texto para introducir la contraseña

6	Ccheckbox para seleccionar el rol del usuario
7	Botón para guardar los datos
8	Botón para cancelar la operación de inserción

Gestionar evaluaciones pluviométricas

Gestionar evaluaciones pluviométricas constituye una de las áreas de trabajo de la aplicación y se encuentra ubicada en la página principal del rol Jefe (véase Figura 32). En esta área se pueden introducir nuevas evaluaciones en la base de datos, editar o eliminar una existente si se encuentra en el estado de planificada, así como otras opciones de filtrar evaluaciones por fecha o mostrar todos los datos de una seleccionada.



Figure 32:Gestionar evaluaciones planificadas

La siguiente tabla muestra la relación entre cada elemento del área de trabajo Gestionar evaluaciones planificadas y el número correspondiente (Tabla 22).

Table 22: Relación entre cada elemento del área de trabajo Gestionar evaluaciones planificadas y el número correspondiente

Número	Elemento
1	Formulario donde se muestran los datos de la evaluación seleccionada
2	Filtro para ver evaluaciones por fecha

3	Botón para agregar una nueva evaluación planificada
4	Botón para eliminar evaluaciones planificadas
5	Botón para actualizar una evaluación planificada

Formulario para filtrar evaluaciones por fecha

El formulario para filtrar evaluaciones por fecha se muestra después de pulsar en el botón de la tabla (véase Figura 33). Este formulario antes de actualizar en la base de datos, valida cada uno de sus campos. No pueden quedar campos en blanco. Si el formato de fecha introducido en alguno de los campos no es correcto (dd/mm/yy), el sistema muestra un mensaje de error indicando el problema.

Figure 33: Formulario para filtrar evaluaciones por fecha

La siguiente tabla muestra la relación entre cada elemento formulario Filtrar evaluaciones por fecha y el número correspondiente (Tabla 23).

Tabla 23: Relación entre cada elemento del formulario Filtrar evaluaciones por fecha y el número correspondiente

Número	Elemento
1	Campo texto para introducir la fecha inicial del rango de búsqueda
2	Campo texto para introducir la fecha final del rango de búsqueda

3	Botón para guardar los datos del formulario
4	Botón para cancelar la operación de filtrado

Formulario para insertar una nueva evaluación planificada

El formulario para insertar una nueva evaluación planificada se muestra después de pulsar en el botón Nueva (véase Figura 34). Este formulario antes de actualizar en la base de datos, valida cada uno de sus campos. No pueden quedar campos en blanco. Si el tipo de dato introducido en alguno de los campos no es correcto, el sistema muestra un mensaje de error indicando el problema.

Figura 34: Formulario para agregar una evaluación planificada

La siguiente tabla muestra la relación entre cada elemento del formulario Nueva Evaluación y el número correspondiente (Tabla 24).

Tabla 24: Relación entre cada elemento del formulario Nueva Evaluación y el número correspondiente

Número	Elemento
1	Campo texto para introducir la máquina a la que se le va a realizar la evaluación
2	Campo texto para introducir la fecha para la cual está programada la evaluación
3	Campo texto para introducir el login del

	evaluador que realizará la evaluación
4	Botón para guardar los datos del formulario
5	Botón para cancelar la operación de inserción

Evaluar

Evaluar constituye el área de trabajo principal de la aplicación y se encuentra ubicada en la página principal del rol Evaluador (véase Figura 35). En esta área se pueden introducir o modificar los datos de cada evaluación y además se calculan las variables necesarias para medir la eficiencia de las máquinas y se grafica el comportamiento de las mismas. Cada vez que se inserta un volumen se recalculan todas las variables y se actualiza el gráfico. Para poder realizar este proceso tiene que estar definido el diámetro, de lo contrario el sistema muestra un mensaje de error.

Importante:

Para eliminar un volumen que ya está adicionado hay que seleccionarlo y presionar la tecla **Del** y para actualizarlo hay que realizar la misma operación, pero con la tecla **F2**.

Pluviopivot
Sistema para la evaluación de máquinas de riego de pivote central

1

Evaluar Perfil Español Inglés Salir

Evaluar

Máquina	Fecha	Estado
M12	03/06/2017	Iniciada
Fregat 1	17/06/2017	Planificada
Cpo 41 Pirey	15/06/2017	Terminada

Datos de las evaluaciones

Volúmenes en c

1-100
2-40

Máquina
M12

Fecha
03/06/2017

Hora
Cultivo y Fase

Inicio Fin

10:00 11:00

Diámetro
15

Presión
5

Equidistancia entre pluviómetros
9

Velocidad de la última torre
m/h

Lámina de riego requerida
mm

Dirección de pluviómetros

Velocidad del viento
m/s

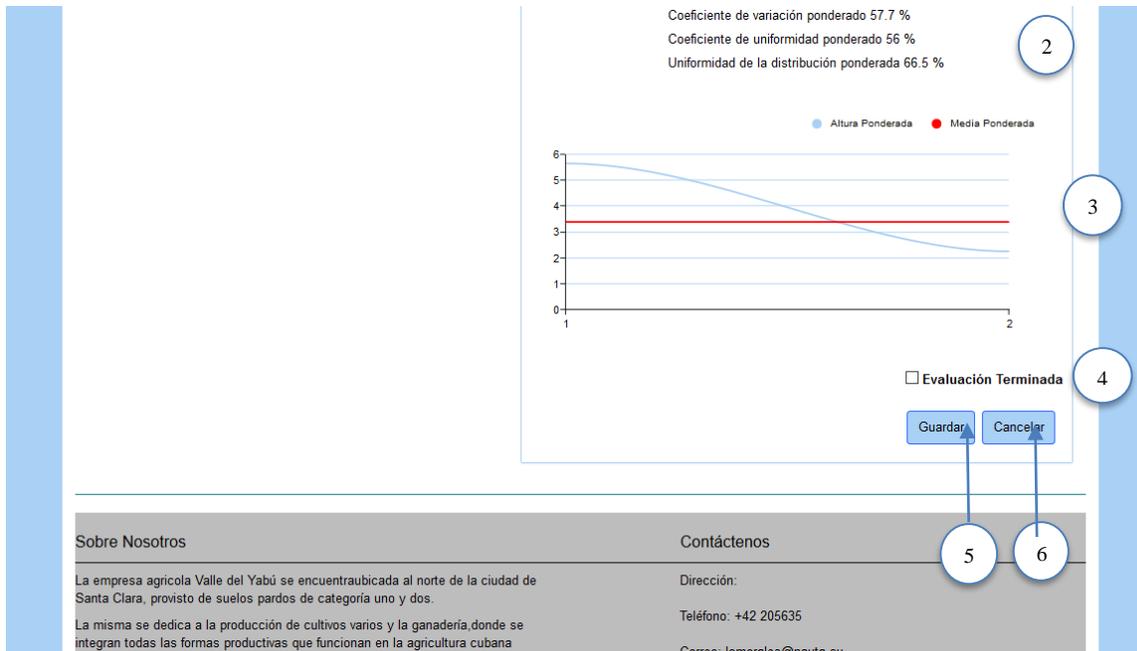


Figura 35:Área de trabajo Evaluar

La siguiente tabla muestra la relación entre cada elemento del área de trabajo Evaluar y el número correspondiente (Tabla 25).

Tabla 25:Relación entre cada elemento del área de trabajo Evaluar y el número correspondiente

Número	Elemento
1	Formulario para introducir los datos de la evaluación
2	Área para mostrar los resultados de las variables calculadas
3	Área donde se grafica el comportamiento de las variables calculadas
4	Ccheckbox para indicar si la evaluación está terminada
5	Botón para guardar los datos del formulario
6	Botón para cancelar la operación de actualización

Formulario para gestionar datos de la evaluación pluviométrica

El formulario para gestionar los datos de la evaluación pluviométrica se muestra en la página principal del rol Evaluador (véase Figura 36). Este formulario antes de actualizar en la base de datos, valida cada uno de sus campos. No pueden quedar algunos campos en blanco por lo que si esto sucede el sistema muestra un mensaje de error. Si el tipo de dato introducido en alguno de los campos no es correcto, el sistema muestra un mensaje de error indicando el problema.

Figura 36:Formulario para la gestión de datos de la evaluación pluviométrica

La siguiente tabla muestra la relación entre cada elemento del formulario para la gestión de datos de la evaluación pluviométrica y el número correspondiente (Tabla 26).

Tabla 26:Relación entre cada elemento del formulario para gestionar los datos de la evaluación y el número correspondiente

Número	Elemento
1	Input para introducir los volúmenes de los pluviómetros
2	Select para mostrar y seleccionar los pluviómetros y los volúmenes de la evaluación
3	Input donde se muestra la máquina a la cual

	está dirigida la evaluación
4	Input donde se muestra la fecha para la cual está programada la evaluación
5	Input donde se inserta la hora de inicio de la evaluación
6	Input donde se inserta la hora de fin de la evaluación
7	Input donde se inserta el cultivo y la fase de la evaluación
8	Input donde se inserta el diámetro
9	Input donde se inserta la presión del agua
10	Input donde se inserta la equidistancia entre los pluviómetros
11	Input donde se inserta la velocidad de la última torre
12	Input donde se inserta la lámina de riego requerida
13	Select para seleccionar la dirección de los pluviómetros
14	Input para insertar la dirección del viento
15	Select para seleccionar la dirección del viento
16	Input para insertar la regulación

Ejemplo de una evaluación terminada (Figura 37).

Evaluar

Máquina	Fecha	Estado
M12	03/06/2017	Iniciada
Fregat 1	17/06/2017	Planificada
Cpo 41 Pirey	15/06/2017	Terminada

Datos de las evaluaciones

Volúmenes en cr

- 1-5
- 2-10
- 3-15

Máquina
Cpo 41 Pirey

Hora Inicio **Hora Fin**

Diámetro **Presión**

Velocidad de la última torre

Dirección de pluviómetros

Dirección del viento

Fecha
15/06/2017

Cultivo y Fase

Equidistancia entre pluviómetros

Lámina de riego requerida

Velocidad del viento

Regulación

Media Ponderada 1.49 mm
Media Ponderada del 25% menos regado 1.06 mm

Coefficiente de variación ponderado 34.9 %
Coefficiente de uniformidad ponderado 72 %
Uniformidad de la distribución ponderada 71.1 %

● Altura Ponderada ● Media Ponderada

Evaluación Terminada

[Sobre Nosotros](#)

[Contáctenos](#)

Figura 37: Ejemplo de una evaluación terminada

Gestionar Perfil

Gestionar Perfil constituye una de las áreas de trabajo de la aplicación y se puede acceder a ella en cualquiera de las páginas principales en la barra de navegación con el menú Perfil (véase Figura 38). En esta área se puede visualizar los datos del usuario logueado y el usuario puede cambiar su contraseña si así lo desea.

Figura 38:Área de trabajo Gestionar Perfil

La siguiente tabla muestra la relación entre cada elemento del área de trabajo Gestionar Perfil y el número correspondiente (Tabla 29).

Tabla 27:Relación entre cada elemento del área de trabajo Gestionar Perfil y el número correspondiente

Número	Elemento
1	Área donde se muestra el nombre y los apellidos del usuario logueado
2	Input tipo password donde se entra la contraseña actual del usuario
3	Input tipo password donde se entra la nueva contraseña del usuario
4	Botón para cambiar la contraseña

Anexo II Datos que puede registrar la aplicación Pluviopivot

En la Tabla 28 se muestra información detallada sobre los datos que se registran en la evaluación, aunque varios de ellos no son requeridos para el cálculo y son guardados en la base de datos de la aplicación.

Tabla 28:Datos que registra la aplicación Pluviopivot

Datos	Requerido	Significado
-------	-----------	-------------

ANEXOS

Volúmenes (cm ³)	Si	Volúmenes recogidos en cada uno de los pluviómetros
Diámetro(cm)	Si	Diámetro de los pluviómetros utilizados
Máquina	No	Identificación de la máquina que se está evaluando
Cultivo y fase	No	Cultivo que se está regando y fase de desarrollo en que se encuentra
Fecha	No	Fecha de la evaluación
Hora de inicio	No	Hora de inicio de la evaluación
Hora de fin	No	Hora de fin de la evaluación
Equidistancia entre pluviómetros(metros)	No	Distancia entre el pluviómetro x y el x+1, que debe ser igual a la distancia del pivote al primer pluviómetro.
Lámina de riego requerida (mm)	No	Lámina de riego que requiere el cultivo en ese momento.
Velocidad del viento (m/s)	No	Velocidad media del viento durante la evaluación
Velocidad de la última torre (m/min)	No	Velocidad de avance de la última torre de la máquina de riego
Regulación No Regulación de la máquina para aplicar la lámina de riego requerida según la escala.	No	Regulación de la máquina para aplicar la lámina de riego requerida según la escala
Dirección del viento (m/s)	No	Dirección del viento según los puntos cardinales
Dirección del radio de	No	Dirección de la línea de

pluviómetros.		pluviómetros en relación con los puntos cardinales.
Presión (Kg./cm ²)	No	Presión existente en el pivote.

Anexo III Aval emitido por la dirección de la empresa “Valle del Yabú” certificando la aplicación Pluviopivot (véase Figura 39).



Figure 39: Aval emitido por la dirección de la empresa “Valle del Yabú”