

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
VALPARAÍSO - CHILE



“ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE HERRAMIENTAS DE SOFTWARE PARA EL ESCALAMIENTO GLOBAL DE WISECONN”

CHRISTIAN DEMMYS REYES LEIVA

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN INFORMÁTICA

Profesor Guía: CECILIA REYES COVARRUBIAS
Profesor Correferente: JOSÉ ULLOA

XXXX - 2023

DEDICATORIA

Considerando la importancia de este trabajo para los alumnos, este apartado es para que el autor entregue palabras personales para dedicar este documento. La extensión puede ser de máximo una hoja y se deben mantener este formato, tipo y tamaño de letra.

AGRADECIMIENTOS

Considerando la importancia de este trabajo para los alumnos, este apartado se podrá incluir en el caso de que el autor desee agradecer a las personas que facilitaron alguna ayuda relevante en su trabajo para la realización de este documento. La extensión puede ser de máximo una hoja y se deben mantener este formato, tipo y tamaño de letra.

RESUMEN

Resumen— El resumen y las palabras clave no deben superar la mitad de la página, donde debe precisarse brevemente: 1) lo que el autor ha hecho, 2) cómo lo hizo (sólo si es importante detallarlo), 3) los resultados principales, 4) la relevancia de los resultados. El resumen es una representación abreviada, pero comprensiva de la memoria y debe informar sobre el objetivo, la metodología y los resultados del trabajo realizado.

Palabras Clave— Cinco es el máximo de palabras clave para describir los temas tratados en la memoria, ponerlas separadas por punto y comas.

ABSTRACT

Abstract— Corresponde a la traducción al idioma inglés del Resumen anterior. Sujeto a la misma regla de extensión del Resumen.

Keywords— Corresponde a la traducción al idioma inglés de Palabras Clave anteriores.

GLOSARIO

Aquí se deben colocar las siglas mencionadas en el trabajo y su explicación, por orden alfabético. Por ejemplo:

UTFSM: Universidad Técnica Federico Santa María.

SaaS: Software as a Service

IoT: Internet of Things

SPA: Sociedad por Acciones

AWS: Amazon Web Services

IDE: Integrated Development Environment

API: Application Programming Interface

REST: REpresentational State Transter

JS: JavaScript

JSF: Java Server Faces

ÍNDICE DE CONTENIDOS

| | |
|--|------|
| RESUMEN | IV |
| ABSTRACT | IV |
| GLOSARIO | V |
| ÍNDICE DE FIGURAS | VIII |
| ÍNDICE DE TABLAS | VIII |
| INTRODUCCIÓN | 1 |
| CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA | 2 |
| 1.1 DESCRIPCIÓN DE LA EMPRESA | 2 |
| 1.1.1 ORGANIGRAMA DE LA EMPRESA | 2 |
| 1.2 PRODUCTOS | 2 |
| 1.2.1 HARDWARE | 2 |
| 1.2.2 SOFTWARE | 3 |
| 1.3 SITUACIÓN ACTUAL | 4 |
| 1.3.1 DESCRIPCIÓN | 4 |
| 1.3.2 ACTORES | 4 |
| 1.3.3 ÁRBOL DEL PROBLEMA | 5 |
| 1.4 SOLUCIÓN | 6 |
| 1.4.1 OBJETIVO GENERAL | 6 |
| 1.4.2 OBJETIVOS ESPECÍFICOS | 6 |
| 1.4.3 ALCANCE DE LA SOLUCIÓN | 6 |
| CAPÍTULO 2: MARCO CONCEPTUAL | 7 |
| 2.1 CONCEPTOS TÉCNICOS | 7 |
| 2.1.1 SOFTWARE AS A SERVICE (<i>SaaS</i>) | 7 |
| 2.1.2 WEB FRAMEWORKS | 7 |
| 2.1.2.1 FRONT-END | 7 |
| 2.1.2.2 BACK-END | 8 |
| 2.1.3 APPLICATION PROGRAMING INTERFACE (API) | 8 |
| 2.1.3.1 API REST | 8 |
| 2.2 HERRAMIENTAS TECNOLÓGICAS | 10 |
| 2.2.1 JAVASCRIPT | 10 |
| 2.2.1.1 NODE.JS | 10 |
| 2.2.1.2 REACTJS | 11 |
| 2.2.2 JAVA | 11 |
| 2.2.2.1 JAVASERVER FACES | 11 |
| 2.3 COMPETENCIAS EN EL MERCADO | 12 |
| 2.3.1 IRROMETER | 12 |

| | |
|---|-----------|
| CAPÍTULO 3: PROPUESTA DE SOLUCIÓN | 13 |
| 3.1 DISEÑO DE LA SOLUCIÓN | 13 |
| 3.1.1 CONFIGURADOR DE MAPA | 13 |
| 3.1.2 ALINEADOR DE IMÁGENES | 13 |
| 3.2 EJEMPLO DE COMO CITAR FIGURAS E ILUSTRACIONES | 14 |
| CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN | 15 |
| 4.1 EJEMPLO DE COMO CITAR TABLAS | 15 |
| CAPÍTULO 5: CONCLUSIONES | 16 |
| ANEXOS | 17 |
| REFERENCIAS BIBLIOGRÁFICAS | 18 |

ÍNDICE DE FIGURAS

| | | |
|---|--|----|
| 1 | Organigrama de Wiseconn | 3 |
| 2 | Árbol del problema | 5 |
| 3 | Malla Curricular Ingeniería Civil Informática. | 14 |

ÍNDICE DE TABLAS

| | | |
|---|---|----|
| 1 | Coloquios del Ciclo de Charlas Informática. | 15 |
|---|---|----|

INTRODUCCIÓN

Debe proporcionar a un lector los antecedentes suficientes para poder contextualizar en general la situación tratada, a través de una descripción breve del área de trabajo y del tema particular abordado, siendo bueno especificar la naturaleza y alcance del problema; así como describir el tipo de propuesta de solución que se realiza, esbozar la metodología a ser empleada e introducir a la estructura del documento mismo de la memoria.

En el fondo, que el lector al leer la Introducción pueda tener una síntesis de cómo fue desarrollada la memoria, a diferencia del Resumen dónde se explicita más qué se hizo, no cómo se hizo.

CAPÍTULO 1

DEFINICIÓN DEL PROBLEMA

1.1. DESCRIPCIÓN DE LA EMPRESA

La industria agrícola es una de las actividades donde más se consume agua, según cifras del Banco Mundial [Banco Mundial, 2017], el 70 % del agua que se extrae en el mundo es destinado a la agricultura. Este recurso natural es uno de los más importante y, a la vez, más escasos del planeta. Por esto, el uso ineficiente e irresponsable de este elemento afecta negativamente nuestro futuro. La empresa Wiseconn SPA, conscientes de esta realidad, desarrolla tecnologías para optimizar la capacidad de gestión de riego, mejorando así el consumo del agua y el nivel productivo de los agricultores. Para esto, Wiseconn ofrece hardware y software, el primero es para el control y monitoreo de riego, mientras que, el segundo es para almacenar y visualizar los datos recolectados en terreno y para la configuración de componentes.

1.1.1. ORGANIGRAMA DE LA EMPRESA

La estructura de Wiseconn se divide en 4 gerencias:

- **Comercial:** Encargada del marketing y ventas de productos.
- **Operaciones:** Encargada de la producción y logística de la empresa, además, del área de soporte y de servicios en terreno.
- **Tecnología:** Enfocada en generar servicios y soluciones para la empresa y clientes basándose en innovación y tecnologías, apoya en los procesos de departamentos internos y de negocio. Además, mantiene la infraestructura informática y los sistemas.
- **Finanzas:** Encargada de las finanzas de la empresa.

1.2. PRODUCTOS

1.2.1. HARDWARE

El hardware ha sido especialmente diseñado por la empresa para el monitoreo y control de riego, los componentes principales son los nodos, dispositivos de monitoreo y control IoT que operan en terreno y se comunican mediante radio y celular para conectarse a la

- **Admin de DropControl:** Servicio enfocado a los usuarios administradores de campo para realizar modificaciones de sectores de riego, usuarios, servicios API, entre otros.

Para estos servicios se ofrecen planes, cuyos factores diferenciales son esencialmente herramientas especializadas, el acceso a la aplicación web y/o móvil, cantidad de usuarios permitidos, almacenamiento, acceso a funcionalidades, entre otros. Wiseconn también tiene software interno utilizado por trabajadores de distintas áreas de la empresa como soporte y producción, para hacer configuraciones avanzadas de campos o temas productivos como manejo de inventarios, lotes, despachos, etc. Dentro de estos softwares están:

- **Operations:** Plataforma para la gestión de lotes y despacho de productos.
- **Setup:** Plataforma de configuración paso a paso. Provee herramientas de gestión para cuentas, campos y usuarios con el fin de crear y organizar los permisos a usar en DropControl o en el mismo Setup.

1.3. SITUACIÓN ACTUAL

1.3.1. DESCRIPCIÓN

En la actualidad, el usuario cliente no tiene completo acceso para realizar ciertas configuraciones en los campos debe recurrir al área de soporte mandando un ticket con la configuración a realizar. Algunas de estas configuraciones son procesos que consumen mucho tiempo y esfuerzo para los trabajadores de soporte. Esto puede significar una carga innecesaria y/o un aumento en el personal para poder llevar a cabo estos procesos en un tiempo razonable. Por el lado del usuario cliente, genera un descontento debido a la alta espera por la configuración. En los procesos productivos, las herramientas de softwares son limitadas y los trabajadores deben hacer pasos extras innecesarios para poder realizar la tarea correspondiente. Debido a esto, se pueden generar algunos errores y retrasos en producción. Como se explicó anteriormente, Wiseconn ofrece planes gratuitos y de pago. En el caso del plan gratuito, este habilita un punto de entrada para aquellos usuarios que no están dispuestos a realizar pagos recurrentes. Sin embargo, existen limitaciones para el usuario cliente debido a la baja cantidad de funcionalidades y/o herramientas disponibles, lo que impide poder entregar condiciones mínimas al usuario que cumplan con las especificaciones comerciales. Además, puede influir en la poca retención de usuarios y/o que estos no se cambien a un plan de pago.

1.3.2. ACTORES

Dentro de los actores está, primero, el usuario cliente que utiliza y/o paga los servicios que ofrece Wiseconn para la gestión de riego de sus campos. Segundo, los trabajadores de Wise-

conn del área de soporte y producción que utilizan los softwares internos para la realización de sus respectivas tareas.

1.3.3. ÁRBOL DEL PROBLEMA

En la figura 2 se encuentra representado el árbol del problema, que identifica como problema principal las funcionalidades limitadas del software de Wiseconn y alta intervención humana en el proceso. En la parte inferior se presentan las causas representadas en que el usuario depende del área de soporte para la realización de algunas configuraciones, el plan gratuito no cumple con las especificaciones comerciales mínimas y, por último, procesos que consumen mucho tiempo y esfuerzo para los trabajadores. En la parte superior, se presentan los efectos que puede generar este problema como el aumento en la posibilidad de errores, retrasos en producción, usuario final descontento por la espera de las configuraciones, el usuario no cambia al plan de pago o cancela sus servicios y mayor carga en el área de soporte.

1.4. SOLUCIÓN

1.4.1. OBJETIVO GENERAL

Análisis, diseño e implementación de herramientas de software que faciliten tareas a usuarios y mejoren procesos para un escalamiento global de servicios de WiseConn, para así, poder reducir tiempo y esfuerzo en los trabajadores de distintas áreas de la empresa, como también, reducir la brecha asociadas a los cobros de servicios SaaS.

1.4.2. OBJETIVOS ESPECÍFICOS

- Realizar un análisis de los procesos de producción que más cargan y esfuerzo generan, como también de funcionalidades/herramientas necesarias para el plan gratuito.
- A partir del análisis previo, diseñar las herramientas de software correspondientes para los servicios de WiseConn.
- Implementación de las herramientas de software en los servicios de WiseConn correspondientes.

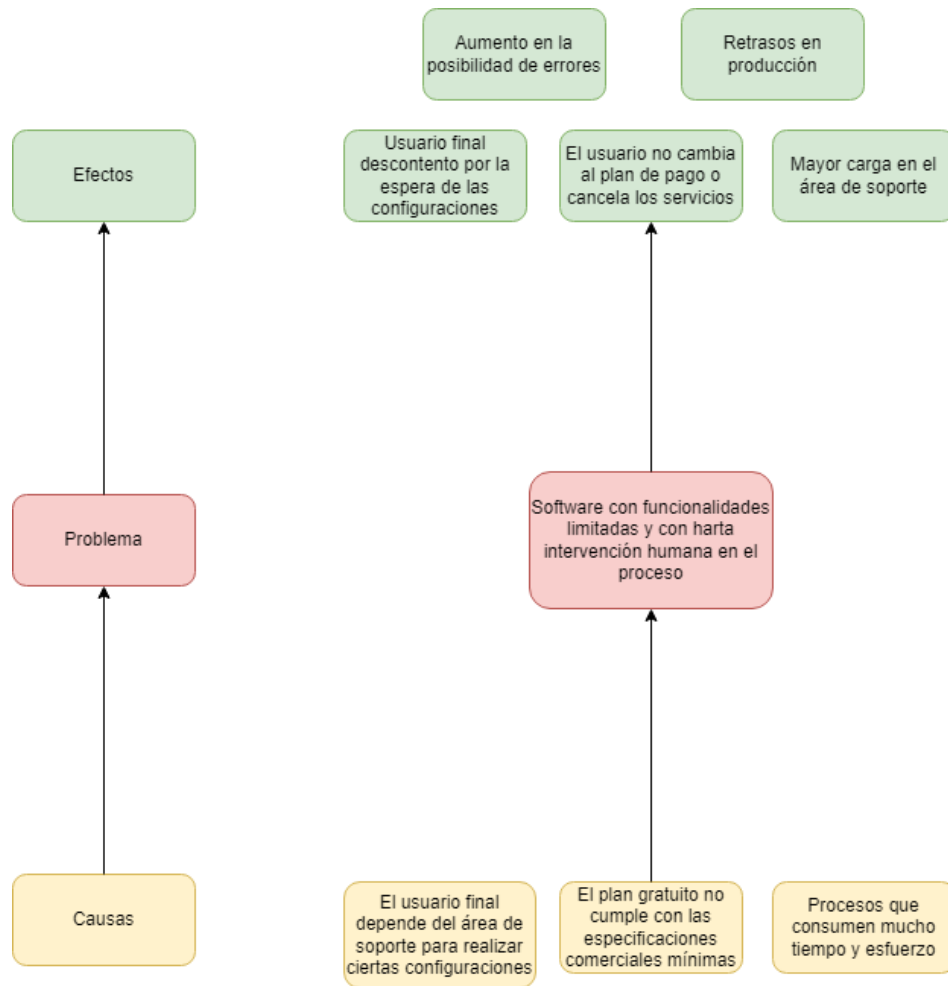


Figura 2: Árbol del problema
Fuente: Elaboración propia.

1.4.3. ALCANCE DE LA SOLUCIÓN

Al finalizar esta memoria se espera reducir los tiempos y esfuerzos en las áreas de soporte y producción de la empresa, ya sea, mejorando herramientas internas o transfiriendo tareas/procesos al usuario cliente. También, ayudar a reducir la brecha de cobros en servicios SaaS con alternativas gratuitas de ciertas funcionalidades, lo que permitiría que el plan gratuito de Wiseconn cumpla con las especificaciones comerciales.

CAPÍTULO 2

MARCO CONCEPTUAL

Se debe describir la base conceptual o fundamentos en los que se basa tu memoria, es decir, todos los conceptos técnicos, metodologías, herramientas, etc. que están involucradas en la solución propuesta. En el fondo esta parte permite precisar y delimitar el problema, estableciendo definiciones para unificar conceptos y lenguaje y fijar relaciones con otros trabajos o soluciones encontradas por otros al mismo problema evitando así plagios o repetir errores ya conocidos o abordados por otros.

En esta parte es importante relacionar estos conceptos con la memoria y es fundamental utilizar referencias bibliográficas (o de la web) recientes, por ejemplo [Gettelfinger y Cussler, 2004].

2.1. CONCEPTOS TÉCNICOS

2.1.1. SOFTWARE AS A SERVICE (*SaaS*)

El software como servicio o SaaS (Software as a Service, en inglés) es un tipo de servicio de cloud computing que se enfoca en la entrega de software basado en la nube, para esto se necesita un proveedor de servicios de nube que desarrolla y mantiene el software de las aplicaciones, proporciona actualizaciones automáticas y lo pone a disposición de sus clientes a través de Internet con un sistema de pago por uso. De este modo, los clientes del SaaS reducen sus costos.¹

2.1.2. WEB FRAMEWORKS

En los inicios del desarrollo web todas las aplicaciones eran programadas “a mano”, lo que provocaba muchos errores. Para poder superar estas dificultades se introdujeron los Web Frameworks en los inicios de la década de los 2000. Los webs frameworks son herramientas que ayudan a construir un sitio web disminuyendo bugs, errores y tiempo. Los frameworks se dividen principalmente en dos categorías: Front-end y Back-end. [Curie *et al.*, 2019]

2.1.2.1. FRONT-END

¹Oracle - What is SaaS?

También conocido como *client-side*, es el componente de la aplicación o página web con el que el usuario interactúa. Incluye lo que el usuario ve como imágenes, botones, colores, gráficos, tablas, entre otros. HTML y CSS son usados para el diseño y estilo, mientras que JavaScript se utiliza para validaciones, animaciones, cambios de estado, etc. El rendimiento y la sensibilidad son los 2 grandes objetivos del desarrollo front-end. El desarrollador se asegura que la aplicación sea sensible, es decir, que se muestren las vistas correctamente en diferentes pantallas como las de monitor, celular, tablets. [Shetty, 2020]

2.1.2.2. **BACK-END**

En *back-end* las preocupaciones son otras, por ejemplo, protección de APIs de ataques externos, autenticar usuarios, interacción con bases de datos y manejar solicitudes de usuarios. Los back-end frameworks son evaluados según sus métodos de programación, lenguajes que soporta e interfaces. También, proveen herramientas y plantillas que ayudan a los desarrolladores en varias tareas del desarrollo. [Shetty, 2020]

2.1.3. **APPLICATION PROGRAMING INTERFACE (API)**

Application Programming Interface (API) es un conjunto de herramientas, definiciones y protocolos que se utiliza para que los productos y servicios se comuniquen con otros, sin tener que diseñar permanentemente una infraestructura de conectividad nueva. Las API pueden ser privadas (para uso interno), compartidas (con terceros para brindar flujos de ingresos adicionales) o públicas (entidades externas pueden desarrollar aplicaciones que interactúen con sus API para fomentar la innovación). El propósito de las API es la integración, es decir, se encargan de conectar los datos, las aplicaciones y los dispositivos para que todas las tecnologías puedan comunicarse y trabajar mejor en conjunto.

2.1.3.1. **API REST**

REST es una sigla que significa *REpresentational State Transfer*, es un término acuñado por Roy Fielding presentado en su tesis doctoral [Fielding, 2000], y lo define como un estilo de arquitectura para sistemas distribuidos de hipertexto, describiendo las restricciones de ingeniería de software que guían REST. Estas restricciones son las siguientes:

- **Arquitectura cliente-servidor:** La separación de responsabilidades es el principio detrás de esta restricción. Cuanto menos conoce el servidor del cliente, y viceversa, resulta más fácil el cambio de componentes.

- **Ausencia de estado:** La comunicación debe ser sin estado por naturaleza, es decir, cada petición del cliente al servidor debe contener solo la información necesaria para entender la petición y no puede tomar ventaja de ningún contexto almacenado en el servidor. Por esto, las sesiones se deben mantener en el cliente.
- **Caché:** La data de una response a una petición debe estar implícita o explícitamente etiquetada como cacheable o no cacheable. Si las respuestas es cacheable, se pueden enviar respuestas cacheadas desde cualquier punto de la red sin necesidad que la petición llegue al servidor.
- **Sistema por capas:** Tiene relación con la separación de responsabilidades y establece que un cliente debe conocer únicamente la capa a la que le está hablando. Es decir, no debe saber qué bases de datos se está utilizando, cachés, proxies, etc.
- **Interfaz uniforme:** La principal característica que distingue la arquitectura REST de otras es el énfasis de una interfaz uniforme entre componentes, para que la información se transfiera de forma estandarizada.
- **Código bajo demanda:** REST permite extender la funcionabilidad del cliente mediante la descarga y ejecución de código en forma de scripts.

REST es un estilo de arquitectura no un estándar, es por esto, que las API REST suelen llamarse API RESTful, es decir, una API que sigue la arquitectura REST. Puede resultar complejo implementar todos los principios de REST en una API, por ello, el modelo de madurez de Richardson (Fowler, 2010) que describe cuanto se apega un servicio a las características de REST. Posee cuatro niveles:

- **Nivel 0:** El servicio cuenta con una sola URI que acepta todo el rango de operaciones, con unos recursos poco definidos.
- **Nivel 1:** El servicio cuenta con varias URIs para distintos recursos. Necesita saber qué tipo de operación realizar a través de la URI o el payload.
- **Nivel 2:** El servicio hace uso de las URIs para recursos identificando la operación a través de los métodos HTTP. Además, hace un correcto uso de los códigos de estado.
- **Nivel 3:** El servicio implementa Hypermedia as the engine of application state (HATEOAS).

Los servicios que alcancen el nivel 3 podrán ser considerados como API RESTful.

2.2. HERRAMIENTAS TECNOLÓGICAS

AMAZON WEB SERVICES (AWS)

Amazon Web Services², o AWS, es la plataforma en la nube más utilizada en el mundo, que cuenta con una cantidad de servicios y características ofreciendo desde tecnologías de infraestructura como cómputo, almacenamiento y bases de datos hasta tecnologías emergentes como aprendizaje automático e inteligencia artificial, data lakes e Internet of Things. Dentro de los servicios más importantes están:

- **EC2:** *Amazon Elastic Compute Cloud* ofrece capacidad de computación con más de 500 instancias y la posibilidad de elegir procesador, almacenamiento, redes, sistema operativo y modelo de compra para ajustar a las necesidades del usuario.
- **S3:** *Amazon Simple Storage Service* es un servicio de almacenamiento de objetos que ofrece escalabilidad, disponibilidad de datos y un gran rendimiento. Por las clases de almacenamiento y características de administración fáciles de usar, es posible optimizar costos, organizar datos y configurar controles de acceso.
- **RDS:** *Amazon Relational Database Service* es una colección de servicios administrados que facilita las tareas de configuración, operación y escalado de una base de datos en la nube. Se puede elegir entre motores como Amazon Aurora, MySQL, PostgreSQL, MariaDB, SQL Server, entre otros.
- **Lambda:** Servicio informático sin servidor (serverless) y basado en eventos que permite ejecutar código para cualquier tipo de aplicación o servicio back-end sin la necesidad de servidores.

2.2.1. JAVASCRIPT

JavaScript, o JS, es un lenguaje de programación orientado a objetos, utilizado mayormente para el desarrollo web en ambos *front-end* y *back-end*. JS tiene una gran comunidad, contiene un conjunto de bibliotecas de código abierto que facilitan tareas para el desarrollador como la visualización de datos y comunicación con el usuario.

2.2.1.1. NODE.JS

Node.js es un entorno en tiempo de ejecución asíncrono basado en eventos diseñada para desarrollar, mayormente, en server-side. Soporta el manejo de muchas conexiones al mis-

²¿Qué es AWS?

mo tiempo. Node.js tiene propiedades que lo hacen escalable. Usa el motor V8 de Google para ejecutar código de JavaScript transformándolo en código máquina y optimiza mediante métodos complicados. [Shetty, 2020]

2.2.1.2. REACTJS

ReactJS es una biblioteca de JavaScript para el desarrollo de interfaces de usuarios dinámicas. Creado y mantenido por Facebook. Interfaces complejas pueden ser compuestas de piezas pequeñas, aisladas y reusables de código llamadas "componentes". ReactJS usa JSX, preprocesador que agrega sintaxis XML a JavaScript, para la escritura simple de HTML. [Shetty, 2020]

2.2.2. JAVA

Java³ es un lenguaje de programación orientado a objetos lanzado en el año 1995 por Sun Microsystems. Una de las mayores ventajas del desarrollo con Java es su portabilidad, independiente en que dispositivo se programe Java se podrá correr en otros dispositivos. Java es una tecnología que consiste en el lenguaje de programación y una plataforma de software. Para crear aplicaciones con Java se necesita descargar el Java Development kit (JDK). Se escribe el código, luego un compilador lo transforma a Java bytecode que corre en la Java Virtual Machine (JVM). Cualquier sistema que soporte JVMs puede correr aplicaciones Java.

2.2.2.1. JAVASERVER FACES

JavaServer Faces⁴ (JSF) es una infraestructura de interfaz de usuario o de APIs que facilita el desarrollo de aplicaciones web de Java. JSF usa principalmente JavaServer Pages (JSP) para el despliegue de las páginas. Los principales componentes son:

- Una API para representar componentes de interfaz de usuario y gestionar su estado.
- Dos bibliotecas de etiquetas JSP para expresar componentes en una página JSP y enlazar los componentes a objetos del servidor.

Una de las grandes ventajas de JSF es la clara separación entre el comportamiento y la presentación, esto permite que cada desarrollador de un equipo se enfoque en su parte del proceso de desarrollo, y proporciona un modelo de programación sencillo para enlazar las piezas. No obstante, en comparación con otros entornos o arquitecturas, no es muy rápida.

³What is Java? | IBM

⁴JavaServer Faces (JSF)

2.3. COMPETENCIAS EN EL MERCADO

2.3.1. *IRROMETER*

IRROMETER, fundada en 1951, manufactura herramientas para la medición, control y muestreo de agua de suelo. Utilizando tensiómetros y sensores

CAPÍTULO 3

PROPUESTA DE SOLUCIÓN

3.1. DISEÑO DE LA SOLUCIÓN

Entendiendo los problemas de los productos de software que ofrece la empresa y las consecuencias que esto conlleva, se plantean nuevas herramientas de software y/o mejoras a herramientas existentes que ayuden a liberar carga y esfuerzo de ciertos procesos a las áreas de soporte y producción, además, darle al usuario cliente nuevas herramientas para que tenga un mejor flujo de trabajo.

3.1.1. CONFIGURADOR DE MAPA

En la aplicación *DropControl*, el usuario tiene a su disposición el mapa de su campo (Agregar imagen). Para mostrar el mapa se utiliza *Google Maps* y se utilizan los polígonos para marcar los sectores del campo y los puntos para los nodos. Para poder configurar este mapa se puede hacer de dos maneras:

1. En la aplicación *Admin de DropControl*, ingresar a cada sector/nodo y agregar su respectivo polígono/punto de manera individual.
2. Enviar un ticket al área de soporte con un archivo con formato .kmz o .kml que contiene los polígonos y/o puntos para sus respectivos sectores y/o nodos.

Para la primera opción, si el campo tiene demasiados sectores/nodos conllevará mucho tiempo en asignar los polígonos/puntos respectivos. En la segunda opción existen problemas como el tiempo que demora el área tome el ticket y realice la configuración y/o que las instrucciones no están bien indicadas llevando a que esta configuración requiera varios tickets para que se haga como el usuario cliente quiera.

Para esto se plantea una herramienta en la aplicación de *Admin de DropControl* para que el usuario cliente sea el que configure el mapa, subiendo su archivo .kmz o .kml y asignando los polígonos/puntos a sus respectivos sectores/nodos.

3.1.2. ALINEADOR DE IMÁGENES

Una de las herramientas que ofrece *DropControl* es el análisis de imágenes utilizando el índice de vegetación de diferencia normalizada o NDVI⁵ por sus siglas en inglés. Para obtener

⁵NDVI: Índice De Vegetación De Diferencia Normalizada

estas imágenes satelitales se utilizan otros proveedores y se muestran en *DropControl* con *Google Maps*, pero no todas las imágenes tienen las mismas coordenadas en *Google Maps* por lo que se verán desalineadas. Esto se hace mandando un ticket al área de soporte y son ellos lo que se encargan de alinear la imagen.

La solución para este problema se plantea una nueva herramienta para la aplicación de *Admin de DropControl* en la que se puedan alinear las imágenes satelitales a *Google Maps*, para esto se coloca la imagen sobre el mapa y es el usuario cliente el que debe alinear la imagen.

3.2. EJEMPLO DE COMO CITAR FIGURAS E ILUSTRACIONES

Se colocó una imagen que se puede referenciar también desde el texto (Ver figura 3).

| Malla Curricular Ingeniería Civil Informática | | | | | | | | | | | | Plan 73 13 | | | | | | | | | | | | | |
|--|--|-----------------------|--|--------------------------------|--|--------------------------------|--|---|--|-----------------------------------|--|---------------------------------------|--|--------------------------------|--|-------------------------------------|--|--|--|---------------------|--|--|--|-------|--|
| AÑO 1 | | | | AÑO 2 | | | | AÑO 3 | | | | AÑO 4 | | | | AÑO 5 | | AÑO 5 1/2 | | | | | | | |
| SEMESTRE I | | SEMESTRE II | | SEMESTRE III | | SEMESTRE IV | | SEMESTRE V | | SEMESTRE VI | | SEMESTRE VII | | SEMESTRE VIII | | SEMESTRE IX | | SEMESTRE X | | SEMESTRE XI | | | | | |
| INF-131 | | QUI-010 | | INF-134 | | INF-253 | | INF-239 | | INF-236 | | INF-225 | | INF-322 | | INF-302 | | | | | | | | | |
| Programación | | Química y Sociedad | | Estructuras de Datos | | Lenguajes de Programación | | Bases de Datos | | Análisis y Diseño de Software | | Ingeniería de Software | | Diseño Interfaces Usuarios | | Electivo Informática II | | | | | | | | | |
| 3 5 | | 3 5 | | 3 5 | | 3 5 | | 3 5 | | 3 5 | | 3 5 | | 3 5 | | 3 5 | | | | | | | | | |
| MAT-021 | | MAT-022 | | MAT-023 | | MAT-024 | | INF-245 | | INF-246 | | INF-256 | | INF-343 | | INF-303 | | INF-304 | | | | | | | |
| Matemáticas I | | Matemáticas II | | Matemáticas III | | Matemáticas IV | | Arquitectura y Organización de Computadores | | Sistemas Operativos | | Redes de Computadores | | Sistemas Distribuidos | | Electivo Informática III | | Electivo Informática IV | | | | | | | |
| 5 8 | | 2 5 | | 5 7 | | 4 7 | | 3 5 | | 3 5 | | 3 5 | | 3 5 | | 3 5 | | 3 5 | | | | | | | |
| FIS-100 | | FIS-110 | | FIS-130 | | FIS-120 | | FIS-140 | | INF-276 | | ICN-270 | | INF-301 | | INF-311 | | INF-313 | | | | | | | |
| Introducción a la Física | | Física General I | | Física General II | | Física General II | | Física General IV | | Ingeniería Informática y Sociedad | | Información y Matemáticas Financieras | | Electivo Informática I | | Electivo I | | Electivo III | | | | | | | |
| 3 6 | | 3 5 | | 4 8 | | 4 8 | | 4 8 | | 3 5 | | 3 5 | | 3 5 | | 3 5 | | 3 5 | | | | | | | |
| IWG-101 | | INF-152 | | INF-155 | | | | INF-280 | | INF-221 | | INF-285 | | INF-295 | | INF-312 | | INF-314 | | | | | | | |
| Introducción a la Ingeniería | | Estructuras Discretas | | Informática Teórica | | | | Estadística Computacional | | Algoritmos y Complejidad | | Computación Científica | | Inteligencia Artificial | | Electivo II | | Electivo IV | | | | | | | |
| 2 3 | | 3 5 | | 3 5 | | | | 3 5 | | 3 5 | | 3 5 | | 3 5 | | 3 5 | | 3 5 | | | | | | | |
| HRW-132 | | HRW-133 | | INF-260 | | IWN-170 | | INF-270 | | INF-292 | | INF-293 | | INF-266 | | INF-360 | | INF-228 | | | | | | | |
| Humanístico I | | Humanístico II | | Teoría de Sistemas | | Economía IA | | Organizaciones y Sistemas de Información | | Optimización | | Investigación de Operaciones | | Sistemas de Gestión | | Gestión de Proyectos de Informática | | Taller Desarrollo de Proyecto de Informática | | | | | | | |
| 2 3 | | 2 3 | | 3 5 | | 3 5 | | 3 5 | | 3 5 | | 3 6 | | 3 5 | | 3 5 | | 6 10 | | | | | | | |
| DEW-100 | | DEW-101 | | INF-1 | | INF-2 | | INF-3 | | INF-4 | | INF-5 | | INF-6 | | INF-7 | | INF-309 | | INF-310 | | | | | |
| Educación Física I | | Educación Física II | | Libre/ Actividad co-curricular | | Libre/ Actividad co-curricular | | Libre/ Actividad co-curricular | | Libre/ Actividad co-curricular | | Libre/ Actividad co-curricular | | Libre/ Actividad co-curricular | | Libre/ Actividad co-curricular | | Trabajo de Título 1 | | Trabajo de Título 2 | | | | | |
| 1 2 | | 1 2 | | 1 2 | | 1 2 | | 1 2 | | 1 2 | | 1 2 | | 1 2 | | 1 2 | | 1 2 | | 12 20 | | | | | |
| 14 24 | | | | 18 28 | | | | 18 31 | | | | 17 30 | | | | 16 27 | | | | 16 27 | | | | 12 20 | |
| BACHILLER EN CIENCIAS DE LA INGENIERÍA | | | | | | | | LICENCIADO EN CIENCIAS DE LA INGENIERÍA | | | | | | | | | | | | | | | | | |
| <div><div><div><div>Código asignatura</div><div>FIS-110</div><div>Número asignatura</div><div>Física General I</div><div>Nombre asignatura</div></div><div><div>Pre Requisito</div><div>023, 111</div></div><div><div>Créditos</div><div>USM SCT</div></div></div><div><div>Matemáticas, Física y Química</div><div>Transversal y de Integración</div><div>Humanistas, Educación Física y Libres</div><div>Industrial y Comercial</div></div><div><div>Fundamentos de Informática</div><div>Sistemas de Información y de Decisión</div><div>Ingeniería de Software y Datos</div><div>Infraestructura TIC</div></div><div><div>Computación Aplicada en Ciencia e Ingeniería</div><div>Electivos Informática y Electivos</div></div></div> | | | | | | | | | | | | | | | | | | | | | | | | | |
| <div><div><div><div></div><div>Al reverso perfil de egreso, inglés, prácticas, titulación, otros</div></div></div></div> | | | | | | | | | | | | | | | | | | | | | | | | | |
| <div><div><div><div></div><div>Departamento de Informática</div><div>Universidad Técnica Federico Santa María</div></div></div></div> | | | | | | | | | | | | | | | | | | | | | | | | | |

Figura 3: Malla Curricular Ingeniería Civil Informática.
Fuente: Departamento de Informática.

CAPÍTULO 4

VALIDACIÓN DE LA SOLUCIÓN

Se debe validar la solución propuesta. Esto significa probar o demostrar que la solución propuesta es válida para el entorno donde fue planteada.

Tradicionalmente es una etapa crítica, pues debe comprobarse por algún medio que vuestra propuesta es básicamente válida. En el caso de un desarrollo de software es la construcción y sus pruebas; en el caso de propuestas de modelos, guías o metodologías podrían ser desde la aplicación a un caso real hasta encuestas o entrevistas con especialistas; en el caso de mejoras de procesos u optimizaciones, podría ser comparar la situación actual (previa a la memoria) con la situación final (cuando la memoria está ya implementada) en base a un conjunto cuantitativo de indicadores o criterios.

4.1. EJEMPLO DE COMO CITAR TABLAS

Se colocó una tabla que se puede referenciar también desde el texto (Ver tabla 1).

Tabla 1: Coloquios del Ciclo de Charlas Informática.
Fuente: Elaboración Propia.

| Título Coloquio | Presentador, País |
|---|------------------------------------|
| "Sensible, invisible, sometimes tolerant, heterogeneous, decentralized and interoperable... and we still need to assure its quality..." | Guilherme Horta Travassos, Brasil. |
| "Dispersed Multiphase Flow Modeling: From Environmental to Industrial Applications" | Orlando Ayala, EE.UU. |
| "Líneas de Producto Software Dinámicas para Sistemas atentos el Contexto" | Rafael Capilla, España. |
| ... | ... |

CAPÍTULO 5

CONCLUSIONES

Las Conclusiones son, según algunos especialistas, el aspecto principal de una memoria, ya que reflejan el aprendizaje final del autor del documento. En ellas se tiende a considerar los alcances y limitaciones de la propuesta de solución, establecer de forma simple y directa los resultados, discutir respecto a la validez de los objetivos formulados, identificar las principales contribuciones y aplicaciones del trabajo realizado, así como su impacto o aporte a la organización o a los actores involucrados. Otro aspecto que tiende a incluirse son recomendaciones para quienes se sientan motivados por el tema y deseen profundizarlo, o lineamientos de una futura ampliación del trabajo.

Todo esto debe sintetizarse en al menos 5 páginas.

ANEXOS

En los Anexos se incluye todo aquel material complementario que no es parte del contenido de los capítulos de la memoria, pero que permiten a un lector contar con un contenido adjunto relacionado con el tema.

REFERENCIAS BIBLIOGRÁFICAS

- [Banco Mundial, 2017] Banco Mundial (2017). El agua en la agricultura. <https://www.bancomundial.org/es/topic/water-in-agriculture>. Visitado el 30-11-2022.
- [Curie *et al.*, 2019] Curie, D., Jaison, J., Yadav, J., y Fiona, J. (2019). Analysis on web frameworks. *Journal of Physics: Conference Series*, 1362:012114.
- [Fielding, 2000] Fielding, R. (2000). *Architectural Styles and the Design of Network-based Software Architectures*. Tesis doctoral.
- [Gettelfinger y Cussler, 2004] Gettelfinger, B. y Cussler, E. (2004). Will humans swim faster or slower in syrup? *AIChE journal*, 50(11):2646–2647.
- [Shetty, 2020] Shetty, Dash, J. K. J. G. (2020). Review paper on web frameworks, databases and web stacks. *International Research Journal of Engineering and Technology (IRJET)*, 7:5735–5738.