



**HERRAMIENTA DE SOFTWARE PARA GRAFICAR VARIABLES OBTENIDAS EN
UN SISTEMA DE BIOFLOC EN PISCICULTURA**

**MARTIN STEVEN BEDOYA RODRÍGUEZ
160002706**

**MARIELENA AYALA SEGURA
160002703**

**UNIVERSIDAD DE LOS LLANOS
FACULTAD DE CIENCIAS BÁSICAS E INGENIERÍAS
ESCUELA DE INGENIERÍAS
PROGRAMA DE INGENIERÍA DE SISTEMAS
VILLAVICENCIO, COLOMBIA
2017**

**HERRAMIENTA DE SOFTWARE PARA GRAFICAR VARIABLES OBTENIDAS EN
UN SISTEMA DE BIOFLOC EN PISCICULTURA**

**MARTIN STEVEN BEDOYA RODRÍGUEZ
160002706**

**MARIELENA AYALA SEGURA
160002703**

**TRABAJO DE GRADO PRESENTADO COMO REQUISITO PARCIAL PARA OPTAR
AL TÍTULO DE
INGENIERO DE SISTEMAS**

**DIRECTOR:
ING. ANA BETY VACCA CASANOVA, M.SC.**

**UNIVERSIDAD DE LOS LLANOS
FACULTAD DE CIENCIAS BÁSICAS E INGENIERÍAS
INGENIERÍA DE SISTEMAS
VILLAVICENCIO, COLOMBIA
2017**

AGRADECIMIENTOS

El éxito de una persona, se debe en su mayor parte a quienes estuvieron apoyándola incondicionalmente a lo largo del camino.

A mis padres por su amor, perseverancia, y humildad.

A Martín Steven Bedoya por su lucha ardua, orientación y confianza hacia mí.

A Olga Lucero Vega Márquez por su conocimiento, experiencia, disponibilidad y carisma en todo el proceso.

A ellos, muchas gracias.

Marihelena Ayala Segura.

AGRADECIMIENTOS

A todos los que hicieron parte de este proceso, gracias.

Martin Steven bedoya rodríguez.

CONTENIDO

LISTADO DE FIGURAS.....	7
LISTADO DE TABLAS.....	8
RESUMEN.....	9
1. INTRODUCCIÓN.....	9
1.1 INTRODUCCIÓN	9
1.2 DESCRIPCIÓN DEL PROBLEMA.....	10
1.3 OBJETIVOS	11
1.3.1 Objetivo general.....	11
1.3.2 Objetivos específicos.....	12
1.4 JUSTIFICACIÓN	12
2. MARCO REFERENCIAL	13
2.1 Marco contextual.....	13
2.1.1 Instituto de Acuicultura de los Llanos (IALL).....	13
2.1.2 Tecnología biofloc.....	15
2.1.3 Estado del arte.....	17
2.2 Marco teórico	18
2.2.1 Metodologías	18
2.2.2 Tecnologías y herramientas	19
3. DESARROLLO DEL ARTEFACTO.....	21
3.1 Análisis del entorno.....	21
3.2 Construcción base del conocimiento.....	22
3.2.1 Gestión de la información	22
Solución de software.....	23
Solución de hardware	24
3.3 Construcción y validación del modelo	25
3.3.1 Fase de iniciación	25
3.3.2 Fase de elaboración	32
3.3.3 Fase de construcción.....	43
3.3.4 Fase de transición	49

4. RESULTADOS	55
5. CONCLUSIONES	56
6. TRABAJOS FUTUROS.....	57
7. REFERENCIAS	58
8. ANEXOS.....	60

LISTADO DE FIGURAS

Figura 1. Instalaciones del Instituto de Acuicultura de los Llanos	14
Figura 2. Sistema de cultivo acuícola con biofloc.....	16
Figura 3. Fases Design Science Research	18
Figura 4. Flujo de la arquitectura Modelo-Vista-Controlador.....	20
Figura 5. BeagleBoard	21
Figura 6. Formato de los valores, diligenciado manualmente.	23
Figura 7. Top 10 OWASP 2013.....	24
Figura 8. Especificaciones de la BeagleBone Wireless Green.....	25
Figura 9. Mapa conceptual del modelo de negocio	26
Figura 10. Diagrama de casos de Uso V1.....	30
Figura 11. Diagrama de casos de uso general.....	33
Figura 12. Caso de uso "Generar reporte".....	34
Figura 13. Caso de uso "Gestionar estanques".....	34
Figura 14. Caso de uso "Gestionar variables".....	35
Figura 15. Caso de uso "Gestionar lecturas".....	36
Figura 16. Diagrama de clases.....	37
Figura 17. Modelo entidad-relación.....	38
Figura 18. Diagrama de proceso "AS-IS".....	38
Figura 19. Diagrama de procesos "TO-BE".....	39
Figura 20 Diagrama de colaboración "Generar reporte".....	40
Figura 21. Diagrama de comunicación del CU "Generar reporte".....	40
Figura 22. Diagrama de secuencia del CU "Generar reporte".....	41
Figura 23. Mockup del CU "Mostrar variables".....	42
Figura 24. Mockup con varios CU integrados.....	42
Figura 25. Interfaz de "Gestionar estanque".....	43
Figura 26. Fragmento de código de la clase entidad Estanque.....	44
Figura 27. Vista de variables	45
Figura 28. Fragmento de código de controlador de lecturas.....	46
Figura 29. Interfaz de reportes.....	47
Figura 30. Anidación triple.....	48
Figura 31. Objeto JSON con los datos del reporte.....	49
Figura 32. Diagrama de despliegue	50
Figura 33. Modelo real de interacción entre Hardware y Software.....	52
Figura 34. Strong params de RoR.....	53
Figura 35. Configuración de autorización básica.....	53

LISTADO DE TABLAS

Tabla 1. Declaración del problema.....	30
Tabla 2. Definición de la solución propuesta.....	31
Tabla 3. Roles y responsabilidades.....	32

RESUMEN

El apogeo tecnológico genera la necesidad de reemplazar todas las actividades que se realizan de una forma manual a una tecnológica, donde la simpleza y facilidad para llevarlas a cabo son una de las características destacables. Por ello, el desarrollo de software brilla por los excelentes resultados que ha logrado a través de los años para la solución de cualquier exigencia. El presente trabajo muestra el desarrollo de una herramienta de software para facilitar el monitoreo de ciertas variables que permiten mantener estable un sistema de biofloc en piscicultura. El objetivo primordial es graficar el comportamiento de dichas variables para generar diversos análisis que mejoren el estado y prevengan cualquier futura irregularidad en un estanque biofloc. Se ha usado como metodología de investigación la Design Science Research Methodology (DSR) y el Proceso Unificado Ágil (AUP) para el respectivo desarrollo de software. Entre los resultados obtenidos se realizó un artefacto de software que permite obtener gráficas, un reporte de promedio y la cantidad de anomalías detectadas. Asimismo se desarrolló la exportación a PDF de un reporte general y finalmente se desplegó la aplicación en una placa computadora de hardware libre. Se concluye que la herramienta favorece en tiempo la administración de lecturas, es completamente portable a cualquier lugar del mundo, reduce manejos manuales y facilita la flexibilidad del trabajo en piscicultura.

1. INTRODUCCIÓN

1.1 INTRODUCCIÓN

El instituto de acuicultura de los llanos (IALL) es una unidad académica fundada desde el año 1996. Ésta ha tenido como objetivo fundamental el estudio de organismos acuáticos, específicamente las especies hidrobiológicas del Orinoco. Para cada grupo de especies desde el punto de vista biológico, se crean macro proyectos propuestos por investigadores especializados y estudiantes (EcuRed, s.f.).

En el año 2016 se desarrolló un proyecto que utilizaba la tecnología Biofloc en diferentes estanques. Estos sistemas permiten aprovechar los residuos de la materia orgánica e inorgánica, convirtiéndose en una forma rentable y amigable con el medio ambiente y con los productores que lo emplean (Sistemas Acuicolas, 2017). Para prever por la vida de las especies alojadas en los estanques, se deben analizar y monitorear

rigurosamente el valor de las variables que se obtienen en él. Entre ella están la temperatura, PH, amonio, alcalinidad y oxígeno disuelto (Collazos, y otros, 2015).

En consecuencia, se debe administrar las variables y graficar estudiar su comportamiento y prevenir cualquier futura anomalía en el estanque Biofloc. Para cumplir este propósito los investigadores a cargo lo efectúan de manera manual en lápiz y papel desde su inicio, lo que causa enormes pérdidas de tiempo y complejidad a la hora de realizar cada gráfico que desaparecerían con una solución de TI.

Así pues, con el fin de cubrir dicha necesidad, se desarrolló un sistema de información web denominado FlocOS, que promete facilitar la administración de lecturas en diversos estanques en donde los parámetros de la calidad del agua como sean las variables esenciales para preservar la vida de una determinada especie acuática. Para el logro del cumplimiento efectivo, se utilizó como metodología de desarrollo de software AUP y el framework Ruby on Rails que promete una alta productividad y buenos resultados. Adicionalmente, FlocOS, fue desplegado en una tarjeta de hardware libre que cuenta con WiFi integrado y Bluetooth en un tamaño súper reducido.

En general, se muestran el logro de los objetivos a lo largo del trabajo, permitiendo mostrar paso a paso la metodología desarrollada, los resultados obtenidos y las recomendaciones propuestas para el éxito total de la solución del problema.

1.2 DESCRIPCIÓN DEL PROBLEMA

Debido al impacto ambiental negativo generado por las grandes cantidades de agua necesarias para mantener en funcionamiento los sistemas piscícolas surgió la tecnología Biofloc, la cual es una alternativa que optimiza la producción de peces, disminuyendo el costo de alimentación y reduciendo la contaminación. Esta tecnología se desarrolló en la década de los 70, basada en comunidades microbianas que ayudan a minimizar los recambios de agua, puesto que controlan los productos nitrogenados que se generan en el sistema, los cuales finalmente son proteína que puede ser utilizada como alimento para los peces (AQUACRIA, 2016).

Esta tecnología tiene un grado de tecnicidad más alto con respecto a la piscicultura común. Actualmente, el sistema Biofloc sigue los mismos principios: mantenimiento de la calidad del agua y nutrición de las especies acuáticas. Para ello se cultiva una bacteria que requiere que los sólidos estén suspendidos en la superficie del agua, para

evitar la pérdida de oxígeno disuelto; del mismo modo, requiere niveles específicos de pH, amonio, nitritos, y otras variables que mantienen estable el sistema. Este proceso exige que un especialista, utilizando ciertos dispositivos electrónicos de costo elevado, mida los valores periódicamente, en un plazo no mayor a una semana, y los ajustes utilizando diferentes compuestos como melaza, cloruro de amonio, bicarbonato de sodio, entre otros.

El año anterior, se empezó a implementar el sistema Biofloc en las instalaciones del Instituto de Acuicultura de los Llanos (IALL). En un sistema de Biofloc, la preservación de los cultivos microbianos y los peces, depende de la estabilidad de ciertas variables. Por este motivo, los investigadores del IALL utilizan dispositivos electrónicos, para monitorear los estanques. Todos los datos entregados por los dispositivos se registran en formatos en papel y luego se trazan gráficas de los mismos.

Sin embargo, no existe un artefacto que permita generar información a partir de las variables obtenidas, que además de mantener estable el sistema piscícola, permita visualizar gráficamente estos datos en cualquier instante de tiempo, con el fin de reducir el gasto de medición por el especialista.

Adicionalmente, en ciertas zonas de cultivo no se tiene acceso a infraestructura tecnológica ni redes de comunicación, lo cual hace aún más complejo el proceso de implementación de una solución de TI que fortalezca la tecnología Biofloc o cualquier tipo de cultivo piscícola.

Así pues, FlocOs apoya la necesidad de administrar la información del estanque permitiendo a los investigadores del IALL y a todo público en general, registrar y conservar datos importantes para el sostenimiento del cultivo.

1.3 OBJETIVOS

1.3.1 Objetivo general

Desarrollar un artefacto de software para administrar las variables fundamentales en un sistema de Biofloc.

1.3.2 Objetivos específicos

- Implementar un artefacto de software que permita visualizar a través de gráficos, las variables del estanque, en diferentes periodos de tiempo.
- Implementar un módulo que permita crear estanques para administrar sus propios valores de forma independiente.
- Implementar una solución tecnológica portable que permita desplegarse en cualquier sitio.

1.4 JUSTIFICACIÓN

La creación del proyecto de biofloc que creo el IALL, permitió usar una tecnología que no había sido implementada en la Universidad de los Llanos anteriormente. Ésta funciono durante varios meses sin posibilidad de ligarse a algún sistema de información que pudiese generar más beneficios. La modalidad de trabajo para realizar los análisis y graficas ha sido totalmente manual, lo que genera pérdidas enormes de tiempo y gran complejidad a la hora de ejecutarlas.

No obstante, el inicio del proyecto de biofloc no es el primero en cuanto a la utilización de estanques, lo que de una manera u otra refleja que son años en los que el IALL ha trabajado sin ninguna solución tecnológica. Como es de saber, el objetivo del instituto ha sido trabajar con especies acuáticas, lo que conlleva a que siempre se van a medir valores de variables como temperatura, pH, oxígeno disuelto, amonio, alcalinidad, entre otras. Ya que estas son las variables fundamentales que se tienen en cuenta para medir las condiciones del agua en relación con los requisitos de una o más especies bióticas según sea su propósito ("Meanings of environmental terms", 1997).

La ingeniería de sistemas, como pilar fundamental de toda solución moderna, pretende satisfacer esa necesidad de optimizar las actividades que se realizan en el instituto en cuanto a la administración de los valores obtenidos, graficación y personalización de los valores límite, proporcionar que el análisis posterior sea lo más cercano a la realidad. Adicionalmente, se desea que los investigadores, estudiantes y/o profesionales encargados de los proyectos, administren de forma ordenadas las lecturas, generando la posibilidad de crear múltiples cuentas de estanque privadas, que ayuden al máximo a la independencia y orden de los datos.

Sin embargo, ésta problemática no solo afecta al IALL si no a muchos cultivadores de la

región y del país, los cuales, en algunos casos, ni siquiera tienen acceso a tecnología en sus instalaciones físicas.

Por este motivo, la importancia en desarrollar la herramienta de software FlocOS radica en evidenciar que las soluciones TI, son indiscutiblemente las mejores a la hora de disminuir optimizar recursos y tiempo, aumentar la productividad, eficiencia y sinergias de procesos que, hoy en día no son temas ajenos a cualquier actividad laboral y/o académica.

2. MARCO REFERENCIAL

2.1 Marco contextual

2.1.1 Instituto de Acuicultura de los Llanos (IALL)

Es una unidad académica y de investigación en el campo de la acuicultura, tiene como objeto fundamental de su trabajo el estudio de organismos acuáticos de aguas continentales autóctonos de Colombia, con particular énfasis en las especies ícticas de la Orinoquía Colombiana. De esta forma, la línea matriz de investigación del Grupo de Investigación en Acuicultura del IALL la constituye el estudio de las especies hidrobiológicas de la cuenca del Orinoco, línea que es desarrollada en forma interdisciplinaria con base en las áreas de especialización de los miembros del grupo.

Para cada especie o grupo de especies estudiadas desde el punto de la biología (en ambientes naturales y/o en confinamiento), el grupo programa un abordaje interdisciplinario a través de macro-proyectos propuestos por los investigadores responsables de cada una de las áreas, los cuales son desarrollados con participación directa de profesores coinvestigadores y de estudiantes de pregrado y de postgrado en las modalidades de trabajos de grado, de tesis, monografías y pasantías.



Figura 1. Instalaciones del Instituto de Acuicultura de los Llanos

Misión

El Instituto de Acuicultura de Los Llanos busca contribuir con el desarrollo de la actividad acuícola a través de la investigación, la capacitación y la proyección social. Promueve la actividad dentro de los principios de la sostenibilidad social, cultural, económica, tecnológica y ambiental.

Visión

El Instituto de Acuicultura de Los Llanos se proyecta al futuro como el centro de referencia acuícola nacional en materia de especies nativas de la Orinoquia y de la Amazonía, siendo punto de apoyo vital para la cadena de producción y liderando los retos que la actividad trae con los cambios globales.

Objetivos

- Realizar, apoyar y promover programas y proyectos de investigación en los campos estratégicos del conocimiento básico y aplicado de la acuicultura.
- Establecer, apoyar y promocionar la creación de programas curriculares de pre y postgrado en su campo del saber.
- Participar y promover todas las formas de vinculación y servicios a las comunidades, estableciendo formas de participación comunitaria y preservación de los recursos hidrobiológicos en las actividades acuícolas regionales y nacionales.

- Ofrecer y realizar apoyos efectivos al desarrollo acuícola mediante la socialización de la información, la educación continuada, la asistencia técnica, la asesoría y la consultoría.
- Promover y ejecutar programas y proyectos técnico-productivos que le permitan la consecución de recursos de cofinanciación que le aseguren el cumplimiento de sus objetivos (EcuRed, s.f.).

2.1.2 Tecnología biofloc

La acuicultura es el cultivo de organismos acuáticos en agua dulce y salada que implica intervenciones en el proceso de cría para aumentar la producción. Es probablemente el sector de producción de alimentos de más rápido crecimiento y representa ahora casi el 50 por ciento del pescado destinado a la alimentación a nivel mundial (Desarrollo de un software basado en modelos para redes inalámbricas de sensores., 2016).

La acuicultura tiende a intensificarse para optimizar el espacio disponible de producción y para hacer más eficiente la producción de biomasa ya sea de peces o crustáceos. Una característica intrínseca de estos sistemas es el incremento de acumulación de residuos, materia orgánica y compuestos inorgánicos tóxicos (Lujan). Se han diseñado una serie de sistemas de producción para el cultivo de diversos organismos acuáticos, orientados a disminuir la utilización del agua y del espacio, aumentando considerablemente la densidad de cultivo. Uno de estos sistemas es el denominado Biofloc, que se presenta como una alternativa para mitigar los posibles impactos ambientales negativos generados por las descargas de agua de la acuicultura (FAO, 2017).

La tecnología de Biofloc (ver figura 2) consiste en el desarrollo de flóculos microbianos formados a partir de una alta relación entre el carbono y el nitrógeno en el agua, con poco o nulo recambio y alta oxigenación, en los cuales se utilizan dietas con bajo contenido de proteína cruda o fuentes de carbono externo tales como melaza (caña de azúcar) salvado de arroz, salvado de trigo, entre otros, lo que permite el crecimiento de una comunidad microbiana, sobretodo de bacterias heterótrofas, que metabolizan los carbohidratos y toman nitrógeno inorgánico (principalmente NH_4), reduciendo sus niveles y mejorando la calidad del agua (FAO, 2017).



Figura 2. Sistema de cultivo acuícola con biofloc

El uso de Biofloc en acuicultura proporciona principalmente dos beneficios en la producción de peces o crustáceos: en primer lugar ayuda al mantenimiento de la calidad del agua por medio de un “secuestro” de los compuestos nitrogenados, generando “in situ” proteína microbiana y en segundo lugar contribuye a la nutrición, disminuyendo la conversión alimenticia, los costos con los alimentos y consecuentemente aumentando la rentabilidad de los cultivos (Composición y abundancia de comunidades microbianas asociadas al biofloc en un cultivo de tilapia., 2016).

Un aspecto fundamental en el desarrollo y sostenimiento de un Biofloc, es que los sólidos deben estar suspendidos todo el tiempo ya que un asentamiento de estos traerá un rápido consumo de oxígeno disuelto, perjudicando a la especie cultivada, por lo tanto se debe mezclar y airear el agua constantemente. Además se debe tener en cuentas algunas condiciones básicas para el manejo de esta tecnología como lo es, la medición y control de la temperatura, el PH, el oxígeno disuelto y transparencia del agua a diario, también se debe tener en cuenta el amonio, el nitrato, el nitrito, la alcalinidad y el fosforo en periodos no mayores a una semana (Composición y abundancia de comunidades microbianas asociadas al biofloc en un cultivo de tilapia., 2016).

En consecuencia, la posibilidad de monitorear y controlar la calidad del agua, la adecuada aireación de los estanques, una temperatura apropiada y demás

factores nombrados anteriormente, serán de gran beneficio para la optimización y avance en la aplicación de la tecnología Biofloc en acuicultura.

2.1.3 Estado del arte

El desarrollo de software ha sido imparable en los últimos años, no existe límite de área para su aplicación. En consecuencia, la acuicultura tampoco es una excepción para ello. La creación de herramientas de software para el monitoreo, producción, análisis y creación informes para especies acuáticas ha sido una de las más importantes para la última década (AQUA, 2016).

A nivel mundial, AKVA group es uno de los principales partners en tecnología y servicios para la industria acuícola. Está presente en todos los mercados con oficinas en Noruega, Chile, Dinamarca, Escocia, Islandia, entre otros. En cuanto a software, ofrece herramientas para el control de los procesos de producción, presentación de informes y planificación inteligente (AKVA group, 2015).

En Sudamérica, existe el software AQUASOFT que agiliza los cálculos de gestión de la producción, según el número de campañas al año, determina la cantidad de alimento a suministrar, mortalidad, cosechas, rentabilidad, optimizar recursos, entre otros (Acuicultura Peru, 2014).

En Colombia algunas empresas piscícolas han desarrollado su propio software para gestionar todo el proceso de cultivo, desde inventarios de alimentos hasta registro de alevinos (Piscicola el Mantial, 2016).

Sin embargo, ni en el país ni particularmente en el departamento del Meta existe una herramienta desarrollada similar a FlocOS. Ésta particularmente se centra en solucionar un problema específico respecto a la administración de variables fundamentales de un sistema acuícola.

El instituto de acuicultura de los Llanos es una fuente de información importante para este tipo de desarrollos tecnológicos que permita optimizar y mejorar los procesos realizados durante el cultivo acuícola. Por consiguiente, la facultad de ciencias básicas e ingeniería de la Universidad de los Llanos podría posicionarse como pionera en éste tipo de desarrollos.

2.2 Marco teórico

2.2.1 Metodologías

2.2.1.1 Metodología DSR

El principio fundamental de la DSR es que el conocimiento, comprensión de un problema y su solución, se adquieren en la aplicación y construcción de un artefacto (Hevner, y otros, 2010).

Para ello DSR propone ciertas etapas que se agruparon en tres fases: análisis del entorno, que es donde se visualiza y comprende el problema. Construcción base del conocimiento, en donde se buscan las herramientas, procedimientos, metodologías, entre otros, que solucionarían el problema. Finalmente, la construcción y validación del modelo, donde se utiliza el conocimiento generado para solucionar el problema y ponerlo a prueba (Wanumen, 2012).

La figura que se muestra a continuación describe las fases fundamentales del DRS:

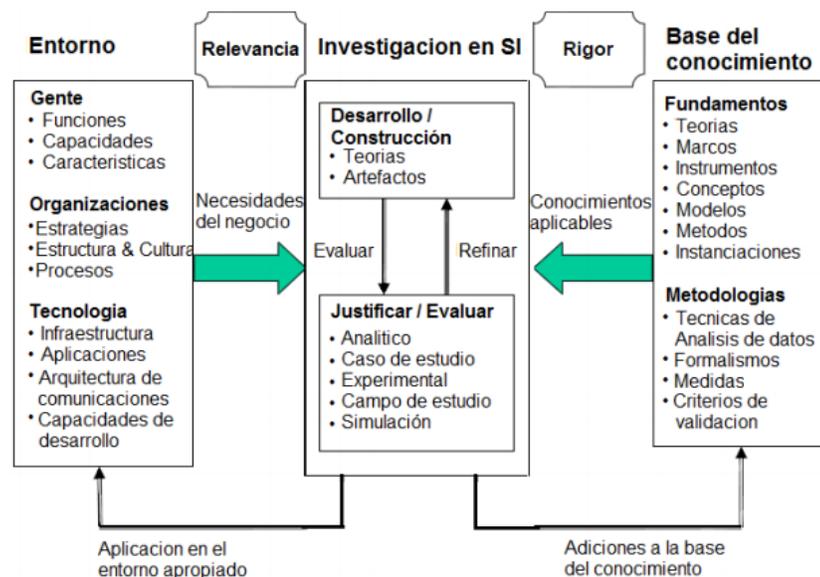


Figura 3. Fases Design Science Research

2.2.1.2 Metodología AUP

El Proceso Unificado Ágil (AUP) de Scott Ambler es una versión simplificada del Proceso Unificado de Rational (RUP) de IBM. Describe un enfoque simple, fácil de entender, del desarrollo de software de aplicación de negocios usando técnicas y conceptos ágiles (Stephens, 2015). AUP contempla 4 fases que completan el ciclo de vida del proyecto: Iniciación, elaboración, construcción, transición.

FlocOS, dentro de su alcance, contempló una pequeña cantidad de procesos puntuales, por consiguiente AUP se ajustó perfectamente en el proceso de software.

2.2.2 Tecnologías y herramientas

A continuación se describen las tecnologías utilizadas para el desarrollo de la herramienta de software FlocOS:

StarUML

Para el lenguaje unificado de modelado se utilizó la herramienta StarUML. Este software se caracteriza por ser de código abierto, flexible, extensible y no está vinculado a un lenguaje de programación específico (StarUML, s.f.). Adicionalmente se adapta perfectamente a la metodología AUP.

MVC

El Modelo–Vista–Controlador (MVC) es un patrón de arquitectura que separa los datos de una aplicación, la interfaz de usuario y la lógica del negocio en tres capas distintas (Yadin, 2016). Se eligió ésta arquitectura por su diseño granular adaptable a múltiples frameworks de desarrollo y a la metodología AUP.

La siguiente figura muestra el flujo de la arquitectura MVC

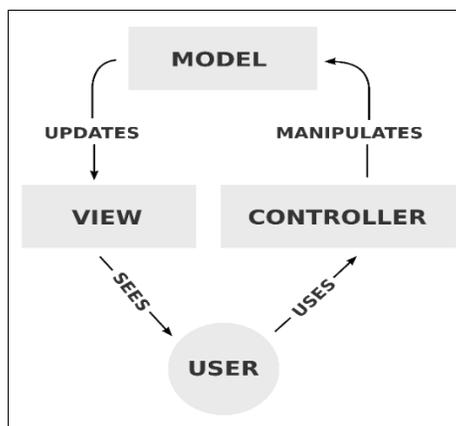


Figura 4. Flujo de la arquitectura Modelo-Vista-Controlador.

Backend

Para el desarrollo backend (CRUD) se utilizó el framework Ruby on Rails (RoR), caracterizado por permitir un desarrollo ágil, con alta productividad y buenos resultados. Esto se debe a que trabaja bajo el fundamento de Don't repeat yourself, DRY (no te repitas) que, en consecuencia, hace que el programador utilice menos líneas de código (Kehoe, 2016).

Frontend

Para el diseño de interfaces se utilizó el motor de plantillas que ofrece Ruby on Rails (RoR) en combinación con el framework Twitter Bootstrap, el cual proporciona una visualización óptima para una experiencia de navegación fácil (Arias, 2014). Bootstrap funciona mediante el llamado de clases que configuradas predeterminadamente en cada uno de los archivos, así pues, la mayor parte del trabajo se basa en la configuración de rejillas, Ids, y modificación de otros archivos de estilos. Por este motivo, FlocOS está disponible para utilizar en cualquier dispositivo, ya sea portátil, smartphone o tablet con cualquier orientación.

OWASP TOP10 2013

El proyecto abierto de seguridad en aplicaciones Web (OWASP por sus siglas en inglés) es una comunidad abierta dedicada a facultar a las organizaciones a desarrollar, adquirir y mantener aplicaciones que pueden ser confiables.

Periódicamente esta fundación genera lineamientos mínimos de seguridad que

toda aplicación web debería cumplir para considerarse confiable, por ende, se tuvo en cuenta el TOP 10 liberado en 2013 para el proceso de desarrollo de software (The OWASP Foundation, 2013).

BeagleBoard

Finalmente, para resolver el aspecto de portabilidad de la solución de TI, se utilizó una plataforma de cómputo de Hardware Libre denominada BeagleBoard. Dentro de las nuevas características se presenta cierta mejora en las especificaciones de la placa y un módulo inalámbrico que permite WIFI Direct. Ésta plataforma permitió la instalación de un sistema operativo tipo Unix sobre el cual opera FlocOS. La siguiente figura muestra la BeagleBoard:



Figura 5. BeagleBoard

3. DESARROLLO DEL ARTEFACTO

Según la metodología elegida (DSR) se tienen las siguientes 3 fases:

3.1 Análisis del entorno

En primera instancia se aborda el objetivo general “Desarrollar un artefacto de software para administrar las variables fundamentales en un sistema de Biofloc”.

Para conocer el problema fue necesario conocer el sistema biofloc y el proceso piscícola en general, el cual consta de 3 etapas: iniciación del cultivo, mantenimiento del cultivo y procesamiento del cultivo.

La fase de mantenimiento del cultivo es la más crítica, dado que se debe mantener estable el sistema para permitir el crecimiento de los flóculos bacterianos de los cuales se alimentarán los alevinos, por consiguiente el estanque debe monitorearse todo el día durante todo el tiempo que se mantenga el cultivo.

El proceso de monitoreo genera la acción de registrar y almacenar datos específicos del estanque, variables fundamentales, como la temperatura, la alcalinidad y el pH, no son propias del sistema biofloc, sino de cualquier cultivo piscícola.

En ese sentido, lo primero que se tuvo en cuenta es que se necesitaba un sistema de información que soportara el proceso de monitoreo, y dado que el desarrollo del registro de información no resulta complejo, fue pertinente el uso de una metodología ágil.

Por otro lado, el desarrollo debería ser portable, capaz de desplegarse en cualquier sitio y a bajo costo, ya que, ésta solución de TI podría beneficiar no solo al IALL, sino a múltiples productores acuícolas de toda la región, que no tengan acceso a infraestructura tecnológica.

3.2 Construcción base del conocimiento

Una vez analizado el entorno se procedió a indagar procesos que solucionasen el problema de la manera más eficiente, para esto fue necesario conocer el proceso de registro de información en el IALL.

3.2.1 Gestión de la información

Luego de varias entrevistas con los funcionarios del IALL se encontró que todo el registro de datos de las variables obtenidas en el sistema biofloc se realizaba manualmente, los datos se escribían en un formato de tabla previamente seleccionado y luego se trazaban gráficas en un tablero de los mismos contra el tiempo. La figura 4 muestra uno de los múltiples formatos que utilizaba el IALL para el registro de los datos.

EVALUACIÓN DE ALTAS DENSIDADES DE CULTIVO SOBRE EL COMPORTAMIENTO PRODUCTIVO DE CACHAMA BLANCA
(*Piaractus brachyomus*) EN UN SISTEMA INTEGRADO DE PRODUCCIÓN BIOFLOC

Variable: calidad de agua

Fecha / hora	Id tanque	Temp (°C)	pH	OD (ppm)	VARIABLES							observaciones		
					Nitrat os (ppm)	nitrito s	Amonio (mg L ⁻¹)	Dureza (mg L ⁻¹)	Alcalin. (mg L ⁻¹)	TDS (ppm)	Salini. (ppm)		Cond. (µS/cm)	
	1	25,3	8,26	8,25 79,2			<<				80	94	194,1	
	2	25,8	8,19	8,33 100,5			<<				67,5	78,1	163,5	
	3	24,8	8,19	8,24 79,4							85	100	271	
	4	24,8	8,10	8,75 77,2							85	100	271	
	5	24,9	8,08	8,12 78,0							66,5	88	170,8	
	6	25,3	8,14	8,03 78,4				30			62	83,6	179,5	
	7	25,4	8,12	8,03 79,9							77,8	94,2	190,6	
	8	25,4	8,09	8,23 79,3							69,8	81,6	171,5	
	9	25	8,06	8,34 100,5							73,8	88,6	180,3	
	10	24,9	8,03	8,28 79,9							66,8	79,5	161,8	
	11	24,7	8,04	8,37 100,4				30			66,3	77	161,9	
	12	24,7	7,97	8,25 79,6							73,9	80,3	185,2	

Otras observaciones:

Figura 6. Formato de los valores, diligenciado manualmente.

Por supuesto, éste proceso tenía ciertas falencias como la pérdida de trazabilidad de los datos y en algunos casos, la mezcla de datos entre uno o más estanques, lo que concluía en información errónea.

Solución de software

Un sistema de información que fuese capaz de gestionar los datos de cada estanque individualmente, sin pérdida de trazabilidad y con una presentación adecuada era suficiente para resolver el problema.

Se eligió una metodología ágil para el desarrollo, particularmente AUP, la versión simplificada de RUP. El proceso unificado ágil, obvia algunos elementos robustos de RUP que retrasan la documentación del mismo.

Dentro de la etapa de construcción del software de la metodología AUP, se eligió Ruby on Rails como framework de desarrollo por su corta curva de aprendizaje y por su principio DRY (Don't Repeat Yourself). Asimismo RoR proporciona múltiples herramientas que hacen el desarrollo más ágil.

La realización de un sistema de información web implicaba ciertas prácticas seguras, que proporcionarían un grado de confiabilidad en el sistema, RoR proporcionó lineamientos que mejoraban la seguridad de la aplicación, para

finalmente evaluarse de acuerdo al criterio de la fundación OWASP y su TOP10 (ver figura 7).

OWASP TOP 10 – 2013
A1 – Injection
A2 – Broken Authentication and Session Management
A3 – Cross-Site Scripting (XSS)
A4 – Insecure Direct Object References
A5 – Security Misconfiguration
A6 – Sensitive Data Exposure
A7 – Missing Function Level Access Control
A8 – Cross-Site Request Forgery (CSRF)
A9 – Using Known Vulnerable Components
A10 – Unvalidated Redirects and Forwards

Figura 7. Top 10 OWASP 2013.

Solución de hardware

Dentro de los objetivos secundarios de éste proyecto se estableció una solución de TI portable que pudiese desplegarse en cualquier sitio sin infraestructura de redes o conexiones a internet. Por tal razón, se pensó en una plataforma de hardware libre de bajo costo donde pudiese ejecutarse el software a través de un navegador (BeagleBoard.org Foundation).

BeagleBoard, es un proyecto de hardware libre de bajo consumo y buen rendimiento producido por la empresa estadounidense Texas Instruments. Luego de conocer las diferentes plataformas ofrecidas por BeagleBoard y buscar una relación costo-beneficio que se ajustase a la economía de la región se optó por el uso de la BeagleBone Green Wireless.

La BeagleBone Wireless (BBGW) (Ver figura 8) permite instalar una distribución de un sistema operativo Unix basado en debían, descargable desde el propio

sitio web de BeagleBoard adicionalmente permite controlar diferentes actuadores gracias a su salidas análogas y digitales.

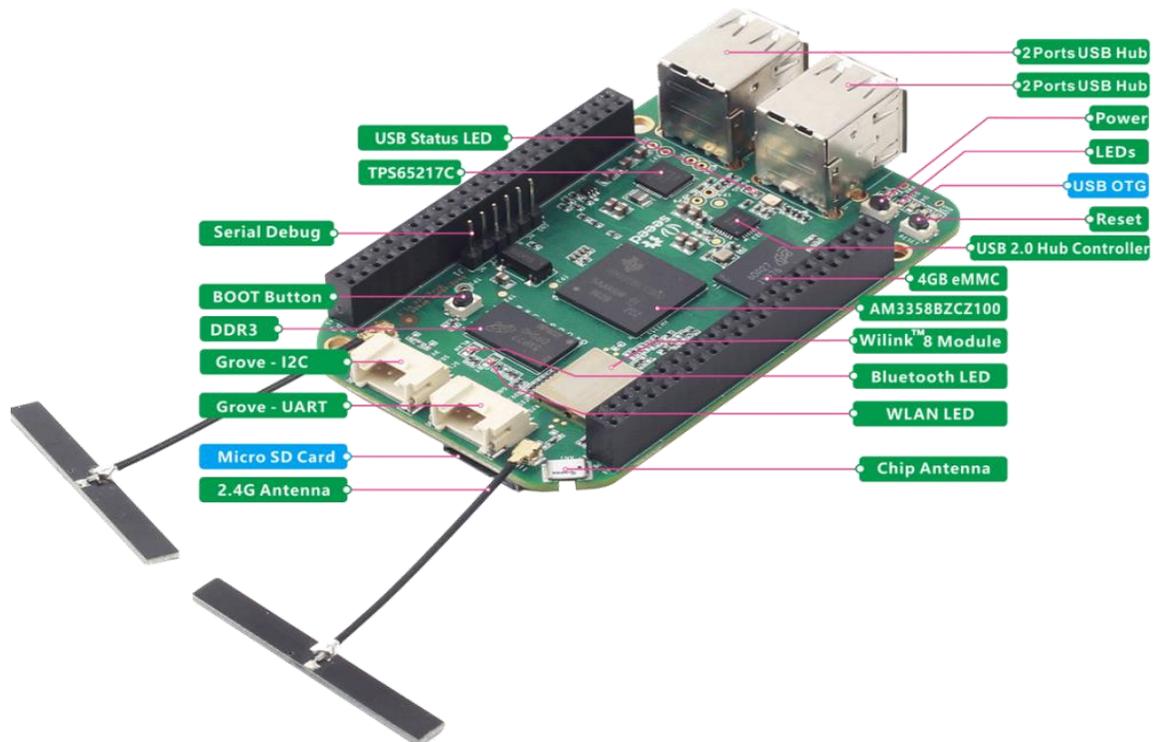


Figura 8. Especificaciones de la BeagleBone Wireless Green

3.3 Construcción y validación del modelo

En esta fase se aborda el cumplimiento de los 3 objetivos específicos definidos. Para la construcción del software de la solución de TI, se utilizó la metodología AUP, a continuación se detalla el proceso de software.

3.2.1 Fase de iniciación

Para iniciar el proceso de software se realizó un análisis del modelo de negocio que permitió conocer los procesos requeridos por el IALL afianzando los conocimientos obtenidos en las dos primeras fases de la metodología DSR, ésta actividad se registró en el siguiente mapa conceptual:

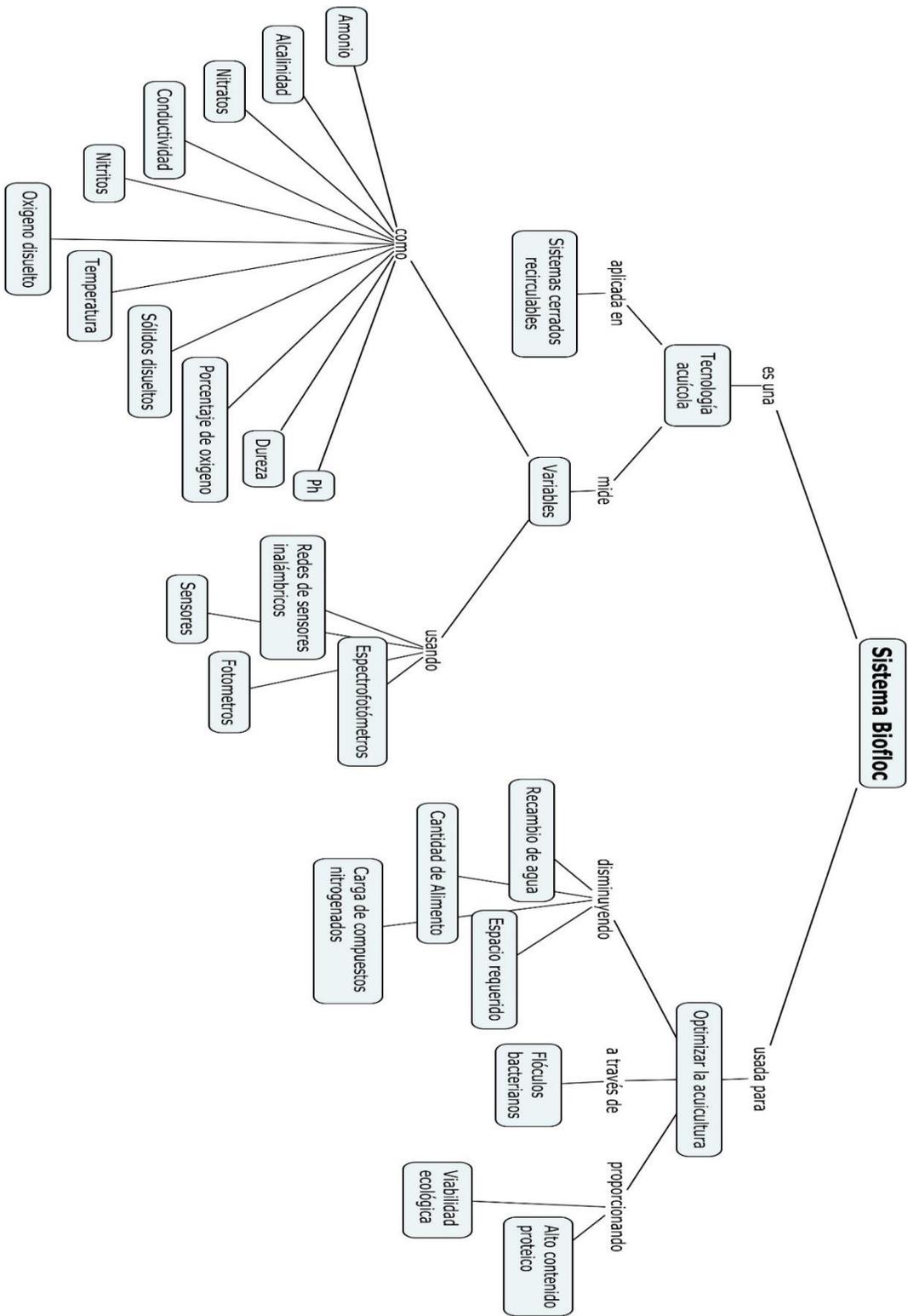


Figura 9. Mapa conceptual del modelo de negocio

En primera instancia se levantaron una serie de requerimientos funcionales y no funcionales base para la construcción del software.

2.2.2.1 Levantamiento de requerimientos V1

Este proyecto suponía la continuación del proyecto iniciado por dos estudiantes de ingeniería electrónica que desarrollaron una red de sensores inalámbricos que permitía obtener las variables fundamentales de un sistema biofloc y enviarlas a un servidor de almacenamiento. No obstante, por diferentes motivos ajenos al desarrollo de éste proyecto, la construcción de la interfaz de comunicación entre la red de sensores y el servidor de almacenamiento nunca se culminó. Por ende la obtención automática de los datos para alimentar un sistema de información no fue posible. A continuación se describen los requerimientos encontrados antes del inconveniente descrito:

Requerimientos funcionales

Número de requisito	RF01
Nombre de requisito	Crear estanque
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	RF02
Nombre de requisito	Iniciar Sesión
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	RF03
Nombre de requisito	Cargar Archivo
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	RF04
Nombre de requisito	Registrar límites
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	

Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional
-------------------------	--

Número de requisito	RF05
Nombre de requisito	Registrar lecturas
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	RF06
Nombre de requisito	Graficar lecturas
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	RF07
Nombre de requisito	Reportar anomalías
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Requerimientos NO funcionales

Número de requisito	RNF01
Nombre de requisito	El sistema FlocOS debe ser un sistema de información WEB.
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	RNF02
Nombre de requisito	La interfaz de usuario debe ser adaptable a múltiples resoluciones.
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	RNF03
---------------------	-------

Nombre de requisito	FlocOS debe cumplir con el 70% o más del TOP10 del OWASP 2013.
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	RNF04
Nombre de requisito	FlocOS debe funcionar en cualquier navegador basado en Chromium.
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

Número de requisito	RNF05
Nombre de requisito	FlocOS debe registrar automáticamente las anomalías en las lecturas, dependiendo de los valores límites definidos previamente por el piscicultor.
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

En principio, la obtención de los datos sería directamente desde la red de sensores inalámbricos, luego se pensó en el uso de un servidor de almacenamiento. Por tal motivo, el RF03 y RF05, estaban pensados manualmente, es decir, en algunos casos el piscicultor interactuaría con la inserción.

El RF07 permitiría enviar un correo electrónico en caso de reportar anomalías en las lecturas, sin embargo, luego de las entrevistas realizadas en el IALL, parecía que la salida de una lectura del rango definido era algo común, por consiguiente, enviar un correo por cada anomalía no sería óptimo, adicionalmente, éste proceso requeriría una conexión a internet.

Éste levantamiento de requerimientos generó un bosquejo preliminar del diagrama de casos de uso (Figura 10).

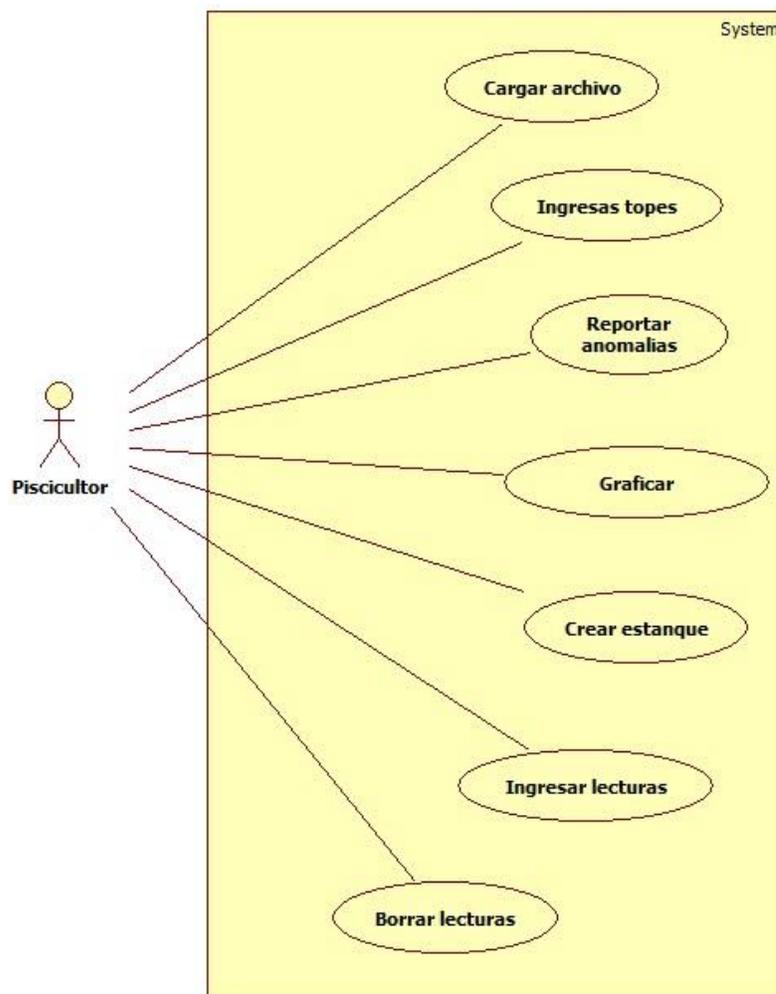


Figura 10. Diagrama de casos de Uso V1.

Adicionalmente, dentro de los anexos se agregan los siguientes documentos pertenecientes a la fase de iniciación de AUP: Glosario, declaración del problema, definición de la solución propuesta, resumen del sistema y roles y responsabilidades.

Tabla 1. Declaración del problema

El problema	Radica en la falta de sistematización de los procesos que se realizan en los estanques. Tales como guardar lecturas, graficar los valores de las mismas y en general la ausencia de generación de reportes.
--------------------	---

Afecta	<ul style="list-style-type: none"> ▪ Piscicultores y encargados de los estanques. ▪ Personal encargado de graficar y administrar los valores de las lecturas.
El impacto del problema es	<ul style="list-style-type: none"> ▪ Acumulación de documentos físicos y riesgo de perder los valores de las lecturas por no poseer un sistema de almacenamiento de información. ▪ Enormes pérdidas de tiempo y gran complejidad a la hora de realizar labores cotidianas. .
Una solución con éxito debería ser	Crear un sistema de información web que administre adecuadamente estanques y lecturas. Adicionalmente éste deberá generar gráficas y reportes de una manera automática.

Tabla 2. Definición de la solución propuesta

Para	El instituto de acuicultura de los llanos (IALL).
Quien	<ul style="list-style-type: none"> ▪ Lideran macro-proyectos como los estanques con sistemas biofloc. ▪ guardan y grafican los valores de las variables fundamentales en un estanque. ▪ Trabajan para preservar y crear promueven la preservación de los recursos hidrobiológicos.
La	Herramienta de software FlocOS.
Que	Crear un sistema de información web que administre adecuadamente estanques y lecturas. Adicionalmente éste deberá generar gráficas y reportes de una manera automática.
A diferencia del	Sistema actual
Nuestro artefacto	Administrara variables fundamentales de un sistema de biofloc, permitiendo visualizar a través de graficar los valores de las mismas. Adicionalmente, nuestra solución será totalmente portable.

Descripción de los roles y responsabilidades

Luego la declaración del problema y la definición de la solución propuesta, se obtiene un listado, se tiene un listado de las personas involucradas en el desarrollo y ejecución de este proyecto, el cual se resume así:

Tabla 3. Roles y responsabilidades

3. Roles y responsabilidades

Nombre	Rol	Responsabilidades
Martín Steven Bedoya Rodríguez	Documentador, desarrollador backend	Documentar DSR, documentar fase de construcción de AUP, desarrollar backend, implementar solución de hardware.
Marihelena Ayala Segura	Documentador, desarrollador frontend	Documentar DSR, documentar AUP, desarrollar frontend, implementar OWASP
Ana Betty Vacca Casanova	Directora de proyecto	Dirigir el proyecto, solicitar reuniones, revisar documentación.
Juan Merlano	Director proyecto Biofloc	Proporcionar información acerca de los estanques y su sistema biofloc. Asimismo, califica los avances del proyecto gracias a su conocimiento en piscicultura.

3.2.2 Fase de elaboración

En ésta fase se iteran los requerimientos encontrados previamente y se ajustan de acuerdo a posibles cambios durante el desarrollo de los mismos.

1.3.2.1 Revisión de los requerimientos

Con el inconveniente presentado con la red de sensores inalámbricos, estos requerimientos cambiaron en cierto porcentaje, algunos se unificaron y otros se eliminaron. Adicionalmente, se ajustó la comunicación y obtención de los datos, basados en la información obtenida en la construcción base del conocimiento de la metodología DSR, suministrada por el IALL. A continuación se describen los cambios generados:

- Se ajustaron los requerimientos RF01, “Crear Estanque” y RF03, “Iniciar Sesión”, y se unificaron en el caso de uso, “Gestionar estanque”.
- Se ajustó el requerimiento RF03, “Registrar Límites” y se incluyó la visualización de las variables, en un solo caso de uso general nombrado “Gestionar Variables”.
- Se ajustó el requerimiento RF05, “Registrar Lecturas” y el caso de uso “Borrar Lecturas”, en uno solo caso de uso general “Gestionar Lecturas”, que también incluyó al RFN05, el cual permite registrar anomalías.
- Se ajustó el requerimiento RF06, “Graficar Lecturas” por “Generar Reporte”, el cual permite además de graficar lecturas, obtener el promedio de las mismas, este cambio sugerido directamente por los investigadores del IALL.

A continuación se presenta el nuevo diagrama de casos de uso general:

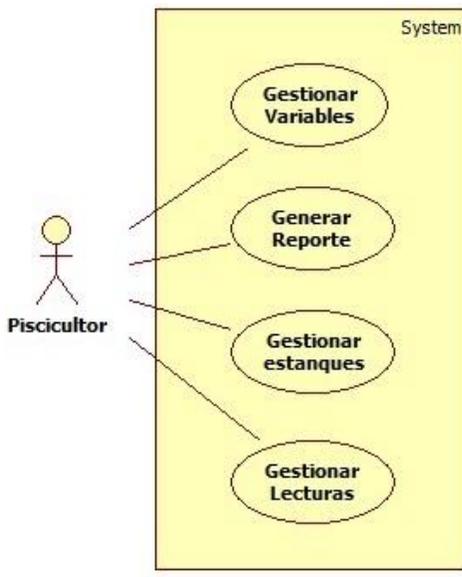


Figura 11. Diagrama de casos de uso general.

Y cada caso de uso específicos, descritos a continuación:

El caso de uso “Generar reporte” permite graficar todas las variables, obtener el promedio y la cantidad de anomalías registradas en un rango

de fechas previamente definido por el piscicultor. Este caso de uso es el más importante dado que permite convertir los datos ingresados, en información que pueda utilizar el piscicultor para realizar ajustes en sus estanques.

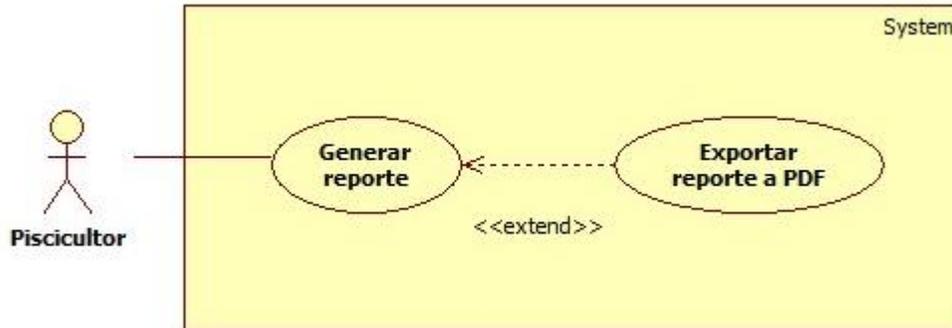


Figura 12. Caso de uso "Generar reporte".

El caso de uso "Gestionar estanques", incluye dos funciones que son: "Iniciar estanque" (anteriormente llamado "Iniciar sesión") y "Crear estanque", que hace alusión a registrar una nueva cuenta de estanque.

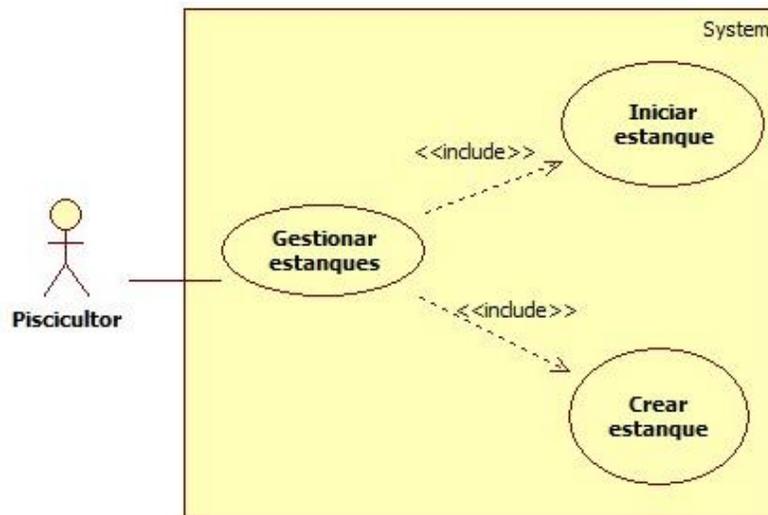


Figura 13. Caso de uso "Gestionar estanques".

"Gestionar variables" entonces, incluye dos casos de uso que se denominan: "Mostrar variables" y "Editar límites". El primero lista las variables fundamentales del estanque (temperatura, pH, oxígeno disuelto,

alcalinidad y amonio) y el segundo, configura los parámetros máximos y mínimos que una variable puede tener para que el sistema funcione adecuadamente.

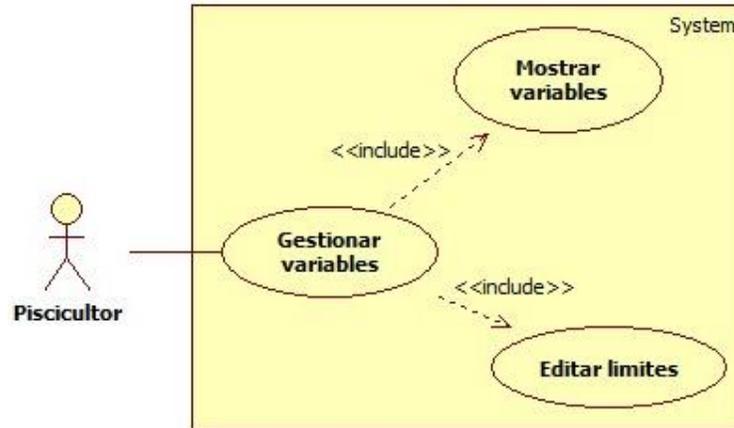


Figura 14. Caso de uso "Gestionar variables".

Finalmente, "Gestionar lecturas", permite visualizar las lecturas que se han ingresado al sistema y eliminar lecturas, permite eliminar una por una, cada lectura, esto con el fin de corregir algún tipo de error en el ingreso de las mismas.

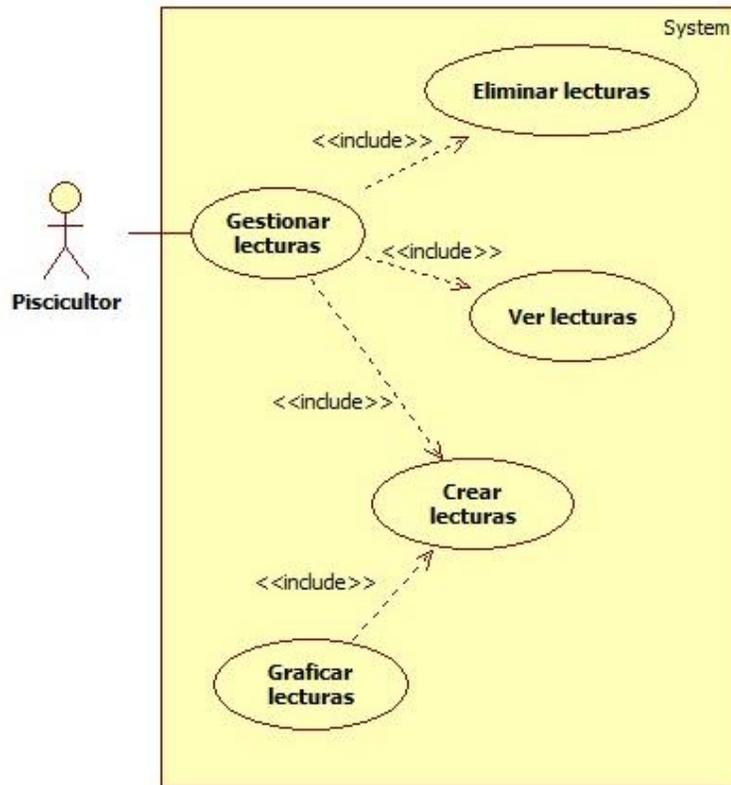


Figura 15. Caso de uso "Gestionar lecturas".

2.2.2.2 Diagrama de clases

La figura 14 representa el diagrama de clases que indica una estructura general de la organización de la estructura del proyecto, dividiendo las clases en 3 tipos, entidad, control y frontera, haciendo similitud al modelo vista controlador.

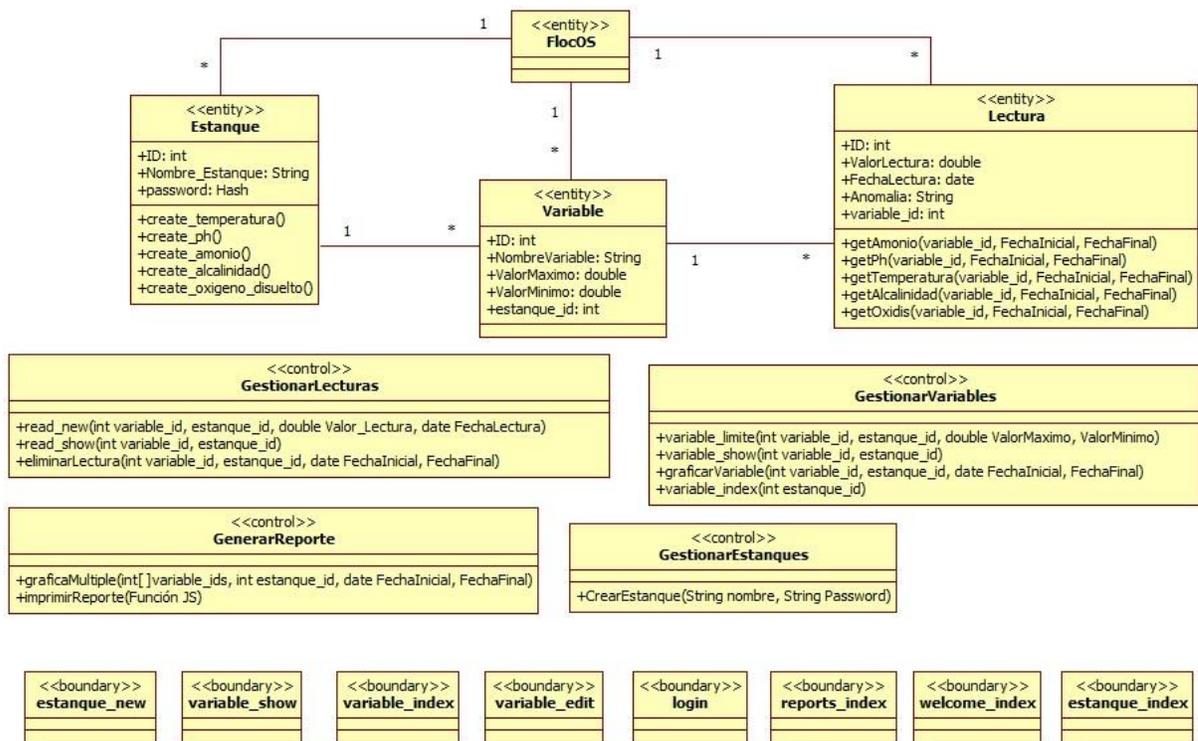


Figura 16. Diagrama de clases.

Las vistas del diagrama de clases, fueron modificadas para que concordaran con el andamiaje generado por Ruby on Rails, aunque StarUML no convierte diagramas en código para un lenguaje como Ruby, no existe inconveniente en utilizarle para visualizar las clases implementadas.

1.3.2.2 Modelo entidad-relación

A continuación se presenta el modelo entidad-relación, que permite visualizar cómo se mapean las clases entidad del diagrama de clases en una base de datos.

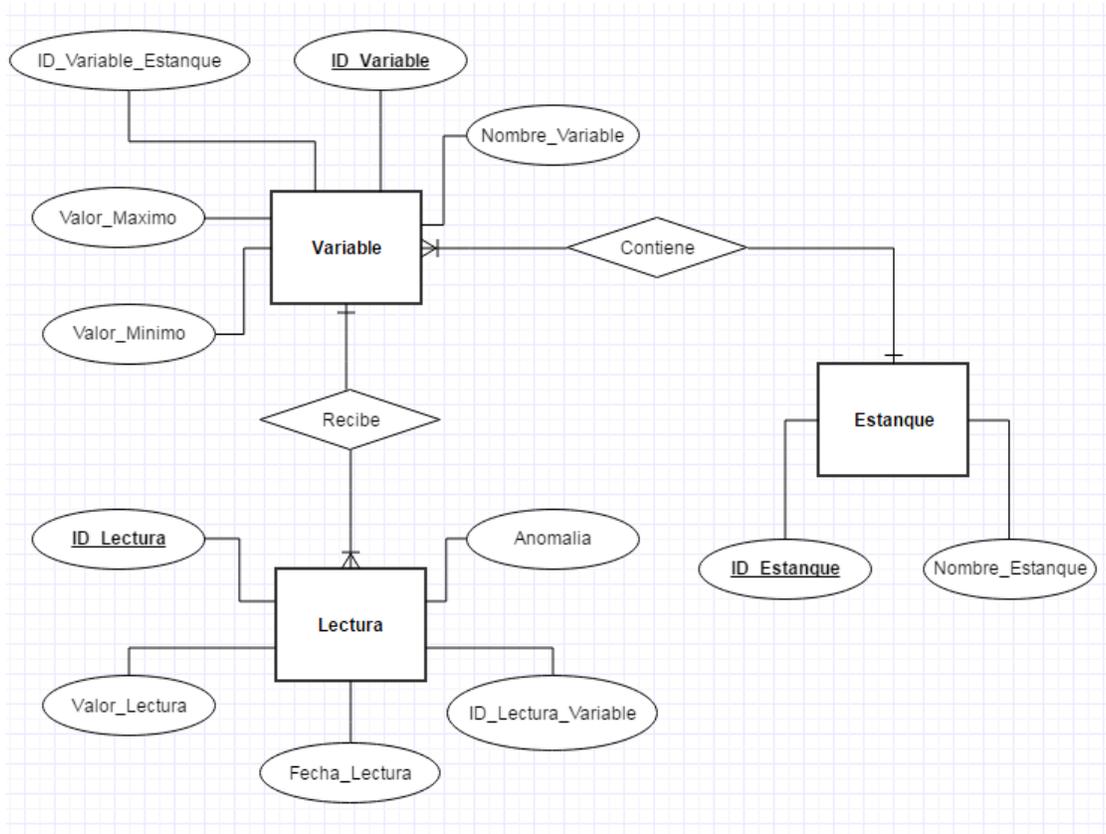


Figura 17. Modelo entidad-relación.

1.3.2.3 Diagrama de actividad

Antes de la construcción del diagrama de actividad, se elaboró un diagrama, tal como está, "AS-IS" (ver figura 18), del proceso de gestión de la información manual llevado a cabo por el IALL.

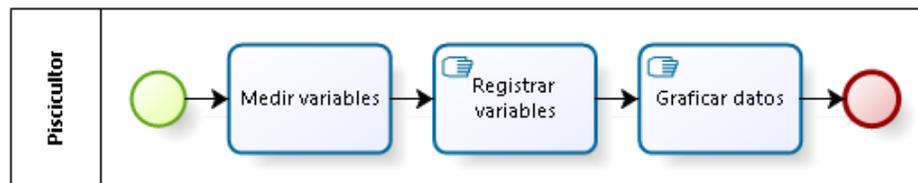


Figura 18. Diagrama de proceso "AS-IS".

A continuación se presenta el diagrama de actividad "TO-BE" en el cual se representa la interacción del piscicultor con el software, el flujo de los datos y la lógica del sistema.

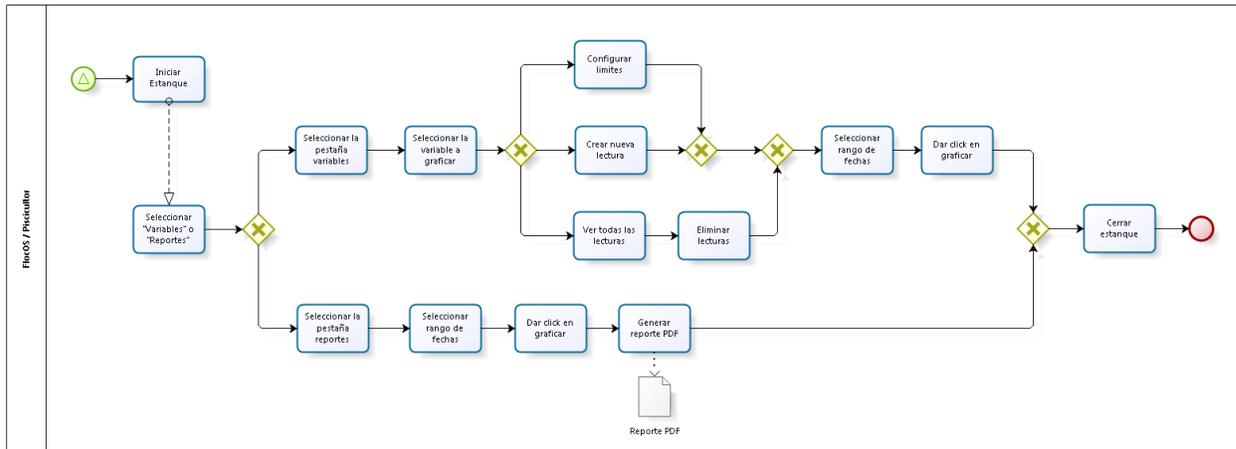


Figura 19. Diagrama de procesos "TO-BE".

2.2.2.3 Diagrama de interacción

Para la interacción y el flujo de los datos, se presenta el caso de uso "Generar reporte". Para ver el detalle de todos los diagramas de interacción diríjase a los anexos "Diagramas de interacción".

A continuación se describen: el diagrama de colaboración como base de la interacción, el diagrama de comunicación para la visualización de los métodos que intervienen en el flujo y finalmente el diagrama de secuencia para la visualización de los objetos.

1.3.2.3.1 Diagrama de colaboración

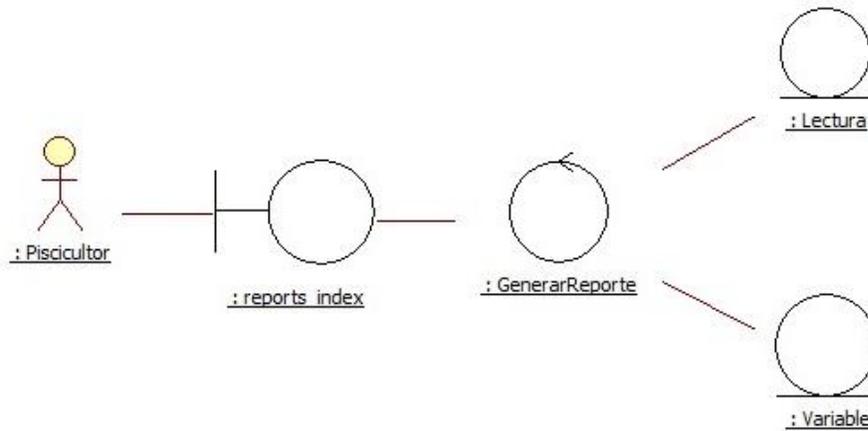


Figura 20 Diagrama de colaboración "Generar reporte".

1.3.2.3.2 Diagrama de comunicación

A continuación se muestra el diagrama de comunicación del caso de uso "Generar reporte"

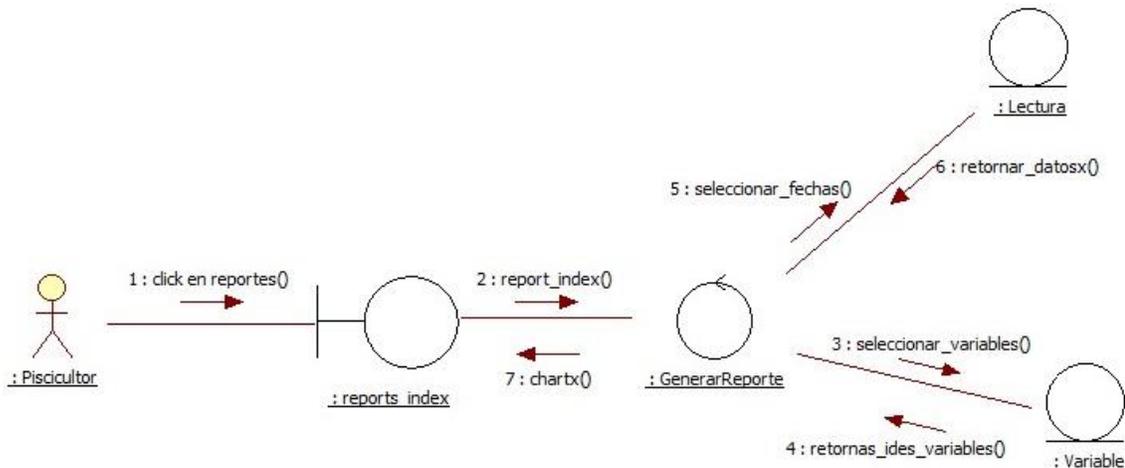


Figura 21. Diagrama de comunicación del CU "Generar reporte".

1.3.2.3.3 Diagrama de secuencia

A continuación se muestra el diagrama de secuencia del caso de uso "Generar reporte".

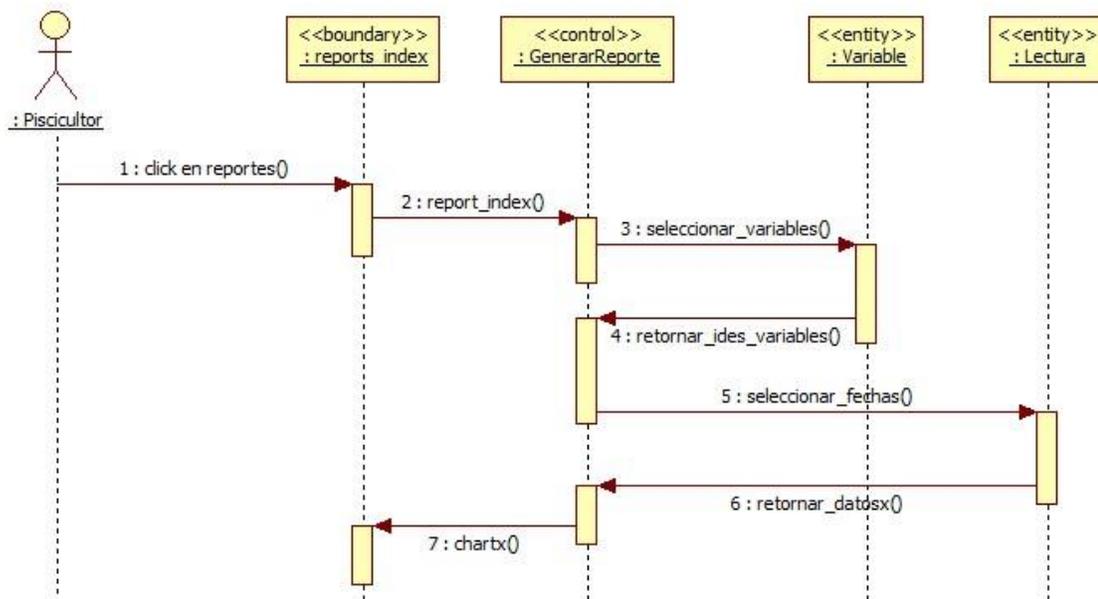


Figura 22. Diagrama de secuencia del CU "Generar reporte".

No obstante, la versión 2.5 de UML suprime el diagrama de colaboración y lo convierte en diagrama de comunicación.

1.3.2.3.4 Prototipos no funcionales (Lo-Fi design)

Para realizar la interfaz gráfica de FlocOS se elaboraron interfaces iniciales que permitieron concretar los diseños de la aplicación orientados al usuario final facilitando la navegación y por supuesto la adaptación de la pantalla.

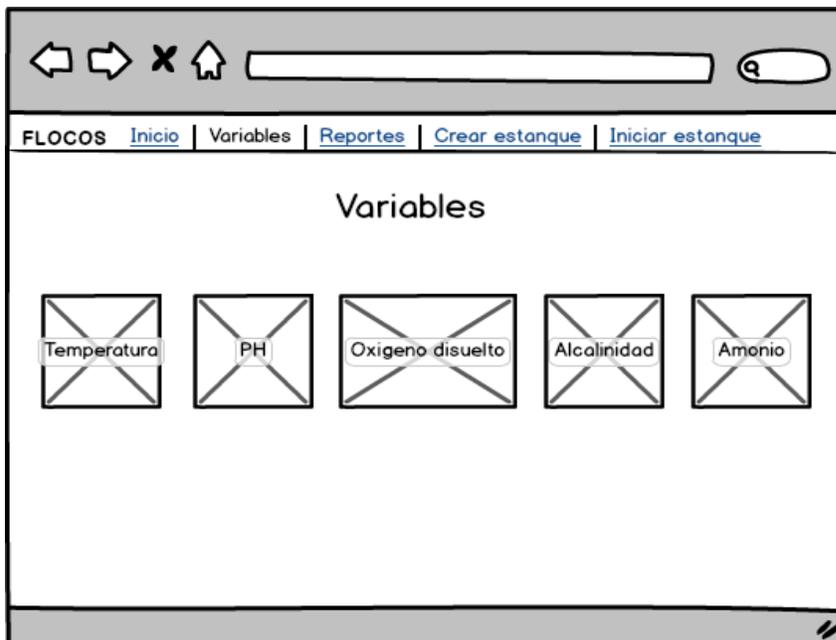


Figura 23. Mockup del CU "Mostrar variables".

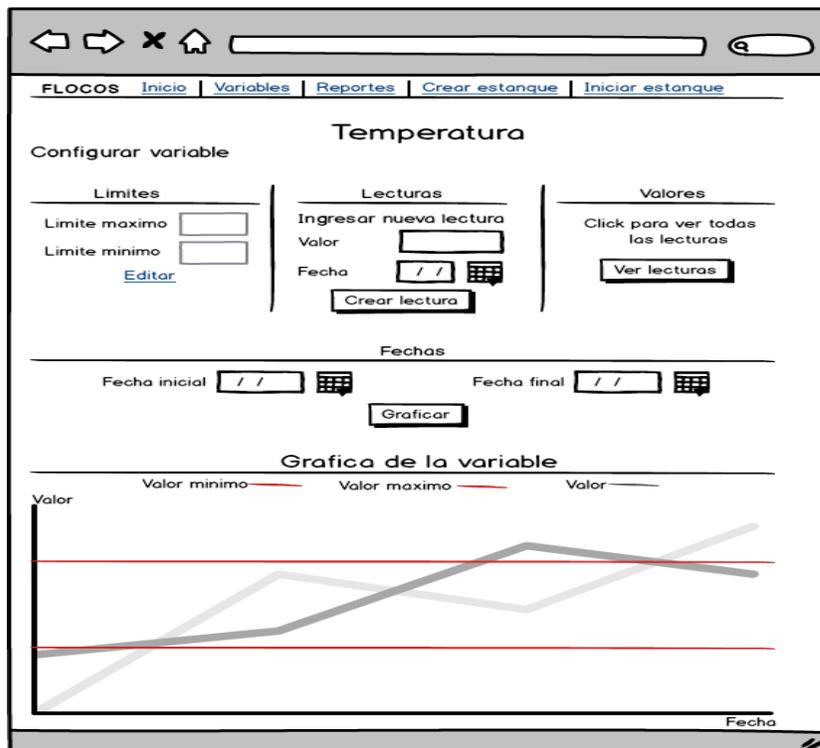


Figura 24. Mockup con varios CU integrados.

Para ver los mockups (Lo-Fi design) completos diríjase al anexo “Mockups”

3.2.3 Fase de construcción

Esta fase se desarrolló mediante múltiples iteraciones, se utilizó el andamiaje de Ruby on Rails para generar las operaciones CRUD básicas y finalmente se adaptaron y se desarrollaron las funcionalidades puntuales para dar cumplimiento a todos los casos de uso, adicionalmente el framework proporcionó un módulo de pruebas unitarias, que verificaron el funcionamiento correcto de los métodos implementados.

1.3.3.1 Gestionar estanque

El primer caso de uso que se desarrolló fue la gestión del estanque, el cual cuenta con dos funcionalidades principales, la creación e inicio del estanque, posteriormente se agregó la funcionalidad de ver los datos del estanque.

Para este caso de uso en particular se utilizó la gema “Devise” desarrollado en Ruby y compatible con Ruby on Rails para la gestión de autenticación y autorización. Devise proporciona diversos “Helpers” que permiten controlar algunos puntos del OWASP TOP10.

Una vez desarrollado el backend se procedió a implementar el frontend con la ayuda de la gema “Twitter-Bootstrap” la cual contiene todos los archivos del framework Bootstrap y los asocia al proyecto.

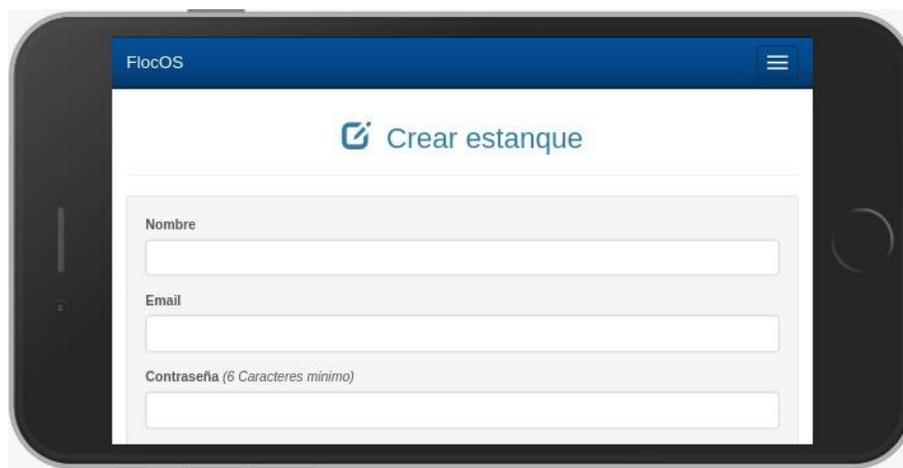
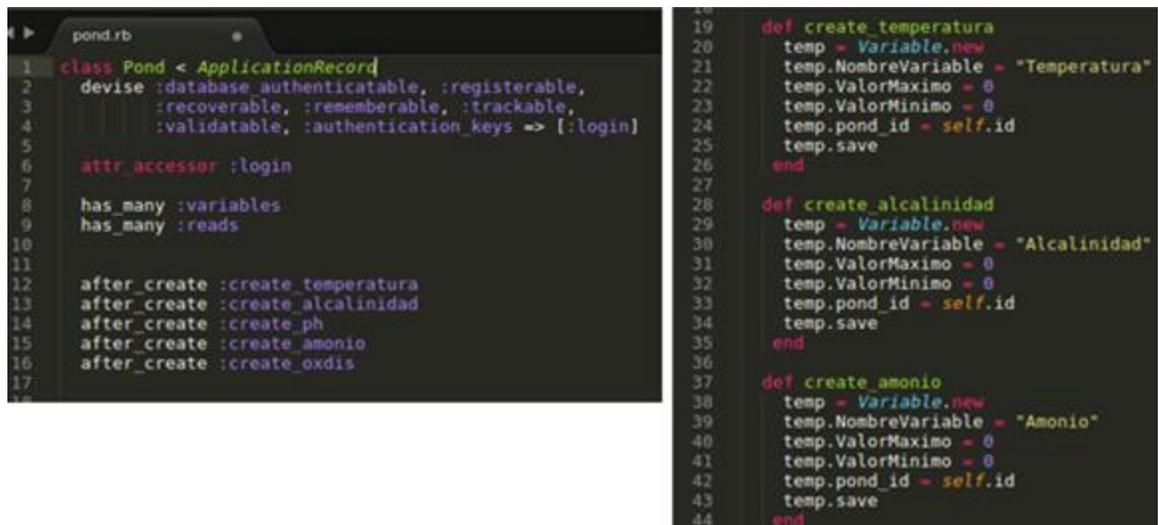


Figura 25. Interfaz de "Gestionar estanque"

1.3.3.2 Gestionar Variable

Inmediatamente culminado el caso de uso anterior, se procede a escribir el código de creación de variables, el cual se refactorizó varias veces, debido a que un estanque siempre contiene las mismas variables fundamentales, en ese sentido, se utilizó la propiedad de creación de métodos de clase de Ruby on Rails para la creación de las variables, en otras palabras, cuando se crea un estanque nuevo, las 5 variables fundamentales se crean automáticamente.

Ruby on Rails permite acceder a entidades relacionadas a través de sus “helpers” desde diferentes clases entidad, esto permite la ejecución de código de un modelo en otro, ahorrando múltiples líneas de código y facilitando el mantenimiento del mismo, pues podrían crearse N variables más.



```
pond.rb
1 class Pond < ApplicationRecord
2   devise :database_authenticatable, :registerable,
3         :recoverable, :rememberable, :trackable,
4         :validatable, :authentication_keys => [:login]
5
6   attr_accessor :login
7
8   has_many :variables
9   has_many :reads
10
11
12   after_create :create_temperatura
13   after_create :create_alcalinidad
14   after_create :create_ph
15   after_create :create_amonio
16   after_create :create_oxdis
17
18
19   def create_temperatura
20     temp = Variable.new
21     temp.NombreVariable = "Temperatura"
22     temp.ValorMaximo = 0
23     temp.ValorMinimo = 0
24     temp.pond_id = self.id
25     temp.save
26   end
27
28   def create_alcalinidad
29     temp = Variable.new
30     temp.NombreVariable = "Alcalinidad"
31     temp.ValorMaximo = 0
32     temp.ValorMinimo = 0
33     temp.pond_id = self.id
34     temp.save
35   end
36
37   def create_amonio
38     temp = Variable.new
39     temp.NombreVariable = "Amonio"
40     temp.ValorMaximo = 0
41     temp.ValorMinimo = 0
42     temp.pond_id = self.id
43     temp.save
44   end
45
```

Figura 26. Fragmento de código de la clase entidad Estanque.

El andamiaje de Ruby on Rails, crea los controladores y las vistas, y las relaciona a través de “helpers” y de su dispatcher, esto permite ahorrar tiempo de desarrollo. En la vista variables_index se posicionaron las 5 variables y se realizó una redirección a variables_show cuando se selecciona alguna.

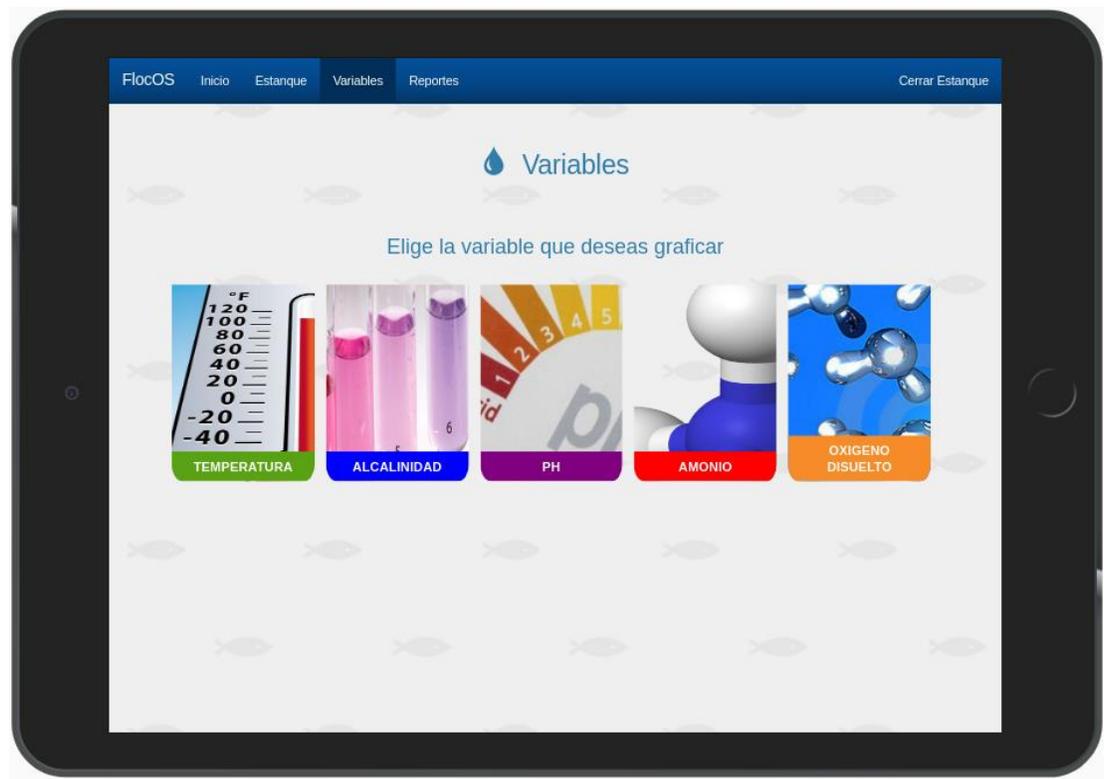


Figura 27. Vista de variables

Variables_show contiene la vista CORE del modelo de negocio, dado que aquí, se registran las lecturas, se visualizan, y se leen los valores máximo y mínimo que puede tener la variable. Gracias a las propiedades de RoR, es posible anidar múltiples vistas de diferentes clases. En el siguiente caso de uso se explica ésta propiedad.

Finalmente, se tienen las pruebas unitarias relacionadas a las vistas de las variables por estanque, y de la interacción dentro variables_show. Adicionalmente, se utiliza la librería ChartJS escrita en JavaScript para generar un gráfico de valores en cierto rango de fechas proporcionadas por la librería DatePicker de JQueryUI.

1.3.3.3 Gestionar Lecturas

Este caso de uso contiene tres acciones, las cuales se visualizan directamente sobre la variable, dado que no tendría sentido mostrar lecturas desasociadas a algún tipo de variable fundamental.

RoR permite anidar vistas de diferentes clases en una sola; en este caso se anidaron todas las vistas de lecturas dentro de su respectiva variable, de ésta manera, desde la vista de la variable seleccionada previamente, se pueden agregar, eliminar y ver nuevas lecturas; no obstante el procesamiento de las peticiones que se generan, se realiza en la clase controlador de lecturas; esto con el fin de conservar el MVC y el principio DRY.

Dentro del controlador de lecturas, se utilizó el Helper de Devise que permite conocer el estanco "logueado", de dicho estanco se obtiene la variable seleccionada y cuando se realiza la inserción del registro de lectura, se llama al controlador de Variable para validar si esa lectura se sale de los valores máximos y mínimos permitidos por dicha variable. En el ejemplo se muestra el esquema del código:

```
reads_controller.rb
1 class ReadsController < ApplicationController
2   before_action :set_read, only: [:show, :edit, :update]
3   before_action :set_variable, except: [:eliminarLectura]
4   before_action :authenticate_pond!
5
6   # GET /reads
7   # GET /reads.json
8   def index
9     @reads = Read.all
10  end
11
12  def new
13    @read = Read.new
14  end
15
16  def create
17    @read = current_pond.reads.new(read_params)
18    @read.variable = @variable
19
20    lmax = @read.variable.ValorMaximo
21    lmin = @read.variable.ValorMinimo
22
23    if @read.ValorLectura < lmin || @read.ValorLectura > lmax
24      @read.Anomalia = "SI"
25    else
26      @read.Anomalia = "NO"
27    end
28    respond_to do |format|
29      if @read.save
30        format.json { render :json => {val: "Registrado!"} }
31      else
32        format.html { render :new }
33        format.json { render json: @read.errors, status: :unprocessable_entity }
34      end
35    end
36  end
37
38  def eliminarLectura
39    read = params[:id]
40    Read.destroy(read)
41    respond_to do |format|
42      format.json { render :json => {val: "Eliminado"} }
43    end
44  end
45 end
```

Figura 28. Fragmento de código de controlador de lecturas.

Esto permite alterar el atributo “Anomalías” de la entidad lectura, para su posterior análisis. Se presentan las pruebas unitarias del controlador.

1.3.3.4 Generar Reporte

Finalmente, se abordó el caso de uso más importante, dado que convierte múltiples datos registrados y gestionados en los anteriores casos de uso en información. Este es el único caso de uso donde no se utilizó el andamiaje de RoR, dado que sus características requirieron cierto tipo de codificación diferente.

En principio, el controlador de reporte, utiliza la clase entidad lectura, donde se construyeron métodos para acceder a sus atributos, repitiendo el proceso utilizado en la creación del estanque, por esta razón, no existe una clase entidad reporte y no se hace persistencia de los mismos. El controlador tiene diferentes métodos que reciben las entradas de una vista propiamente diseñada para el reporte y finalmente gracias a JavaScript se presentan los datos en forma de gráfica y se permite exportar el resultado a un formato PDF.

Dentro de los métodos a los que accede el controlador de reportes, se encuentran métodos de la clase entidad lectura, como por ejemplo, obtener el promedio de los datos y la cantidad de anomalías en ese rango de fechas. De esta manera se cumple a cabalidad el caso de uso.

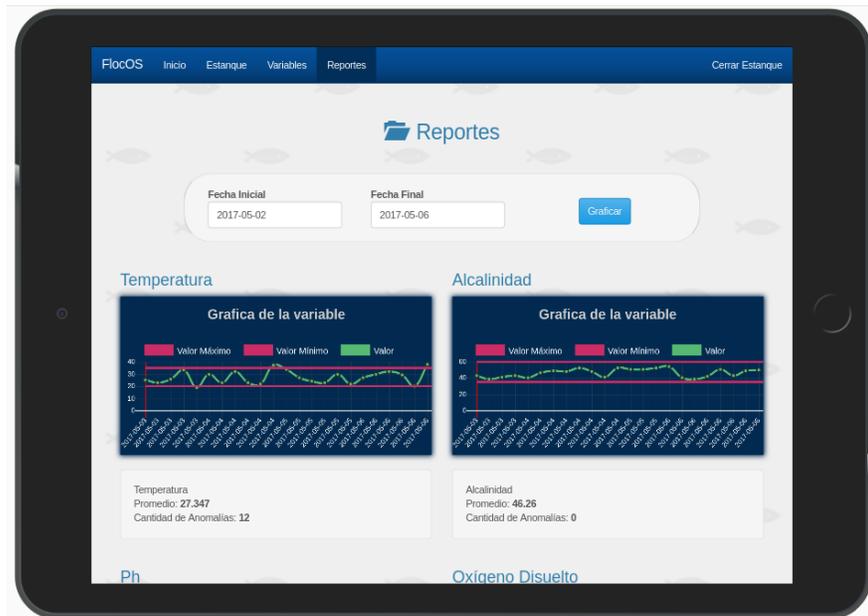


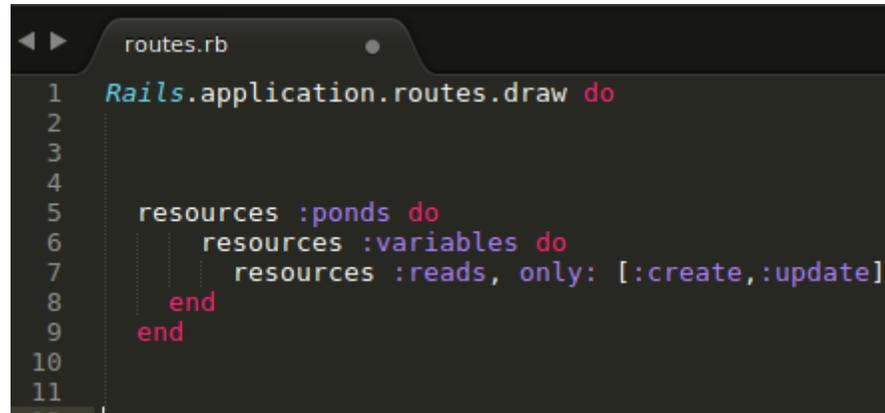
Figura 29. Interfaz de reportes.

1.3.3.5 Helpers y JavaScript

Como se mencionó anteriormente, se codificaron algunos “helpers” adicionales que permitieron pasar datos entre clases controladores y vistas, facilitando la interacción entre las diferentes clases entidad.

RoR, relaciona las entidades a nivel de aplicación gracias a su robusto ORM “ActiveRecord” esto quiere decir que las relaciones a nivel de bases de datos no existen, o no si no se configuran, en algunos casos esto puede representar ciertas desventajas, sin embargo, durante el desarrollo no se presentó algún tipo de problema.

El modelo entidad-relación presentado en la fase de elaboración de la metodología AUP sugiere que un estanque puede tener múltiples variables, en éste caso 5, y que cada variable posee múltiples lecturas, en principio se utilizó una anidación de recursos triple, sin embargo, esto representaba, según la documentación de RoR una mala práctica [referencia ver foro ROR], para solucionar éste problema se implementó una anidación de recursos entre variables y lecturas y para los estanques y variables el propio “helper” de DEVISE.



```
routes.rb
1  Rails.application.routes.draw do
2
3
4
5    resources :ponds do
6      resources :variables do
7        resources :reads, only: [:create, :update]
8      end
9    end
10
11
12
```

Figura 30. Anidación triple

Para las vistas, RoR ofrece el framework JQuery por defecto, con la ayuda del mismo, se mejoraron funcionalidades y se mejoraron procesos del lado del usuario haciendo más sencilla la interacción con el software.

Finalmente el controlador de reporte envía los datos a la vista en un objeto JSON el cual recibe la librería ChartJS para hacer la respectiva gráfica, ésta implementación de renderizado de la gráfica en el lado del cliente, se tuvo

en cuenta, por la cantidad de procesamiento necesario en el lado del servidor, lo cual representaba tiempos de espera más altos para el piscicultor.

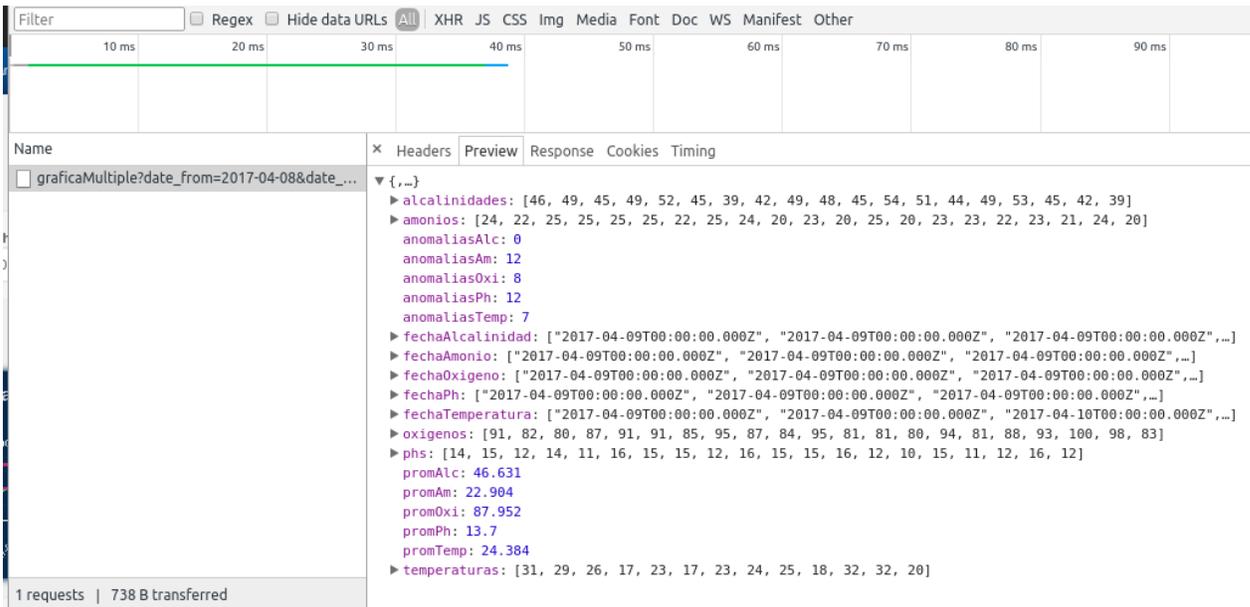


Figura 31. Objeto JSON con los datos del reporte.

3.2.4 Fase de transición

1.3.4.1 Diagrama de despliegue

Finalmente, se diseñó el diagrama que muestra la posición física de las clases y cómo Ruby on Rails interviene para facilitar el trabajo, adicionalmente se muestra el desarrollo por capas y los diferentes componentes esenciales para la consecución del software.

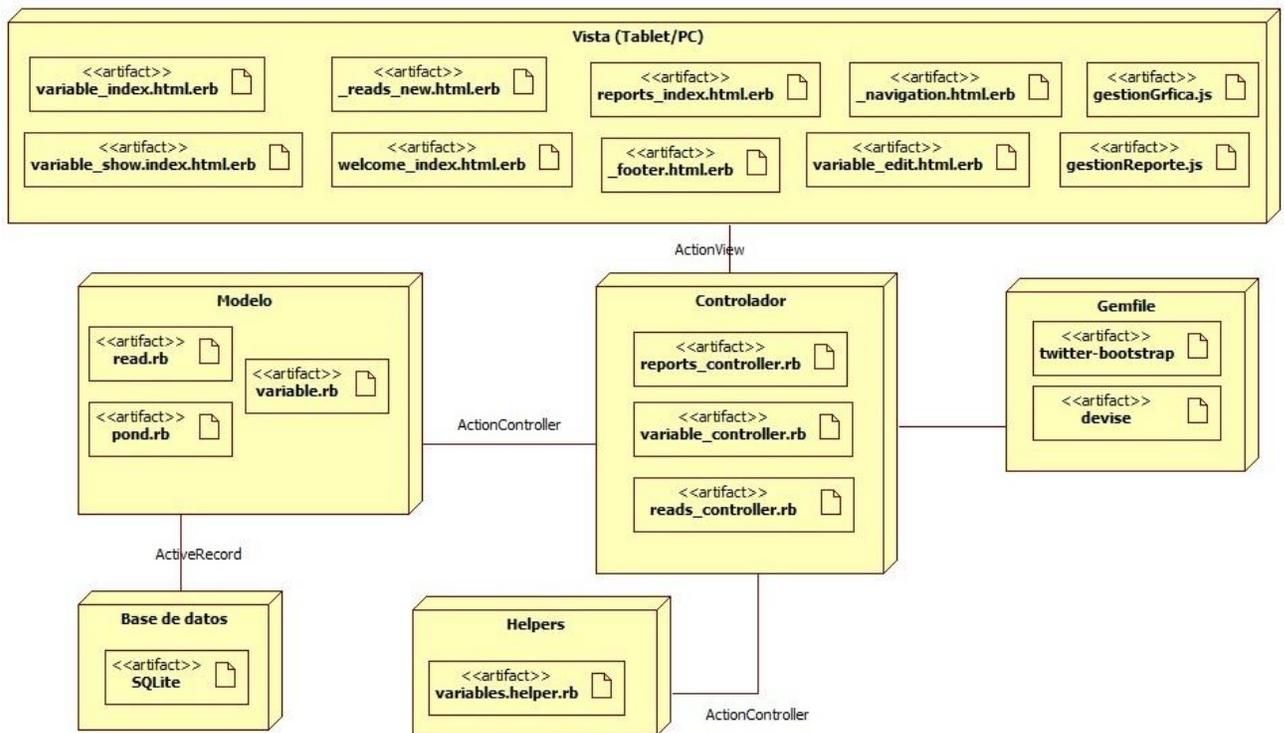


Figura 32. Diagrama de despliegue

1.3.4.2 Pruebas

Aunque Ruby on Rails se presta para Test-Driven Development (TDD). Lamentablemente no fue posible realizar pruebas de integración con el uso de las gemas existentes, dado que ésta implementación se sale de los paradigmas comunes de implementación y adicionalmente, no existe mucha documentación entre las librerías de pruebas de integración y la versión de RoR que se usó. En el anexo X. se encuentran las pruebas unitarias de las clases controladores.

Una vez finalizado el desarrollo, se procedió a realizar pruebas de aceptación presenciales en las instalaciones del IALL, aunque el proyecto Biofloc se finalizó hace un tiempo por diferentes causas, se realizaron pruebas con otras implementaciones piscícolas, con lo cual, los investigadores del IALL quedaron complacidos con la herramienta. A continuación se anexa una de las pruebas realizadas:

ENCUESTA DE SATISFACCION
Programa ingenieria de sistemas
FlocOS

Por favor contesta las preguntas con base en la experiencia de utilizar **FlocOS** considerando 1 como el puntaje más bajo y 5 como el puntaje más alto. Encierre su respuesta con un círculo o márkela con una x.

1. ¿Considera que el proceso de administración de lecturas por medio de **FlocOS** es más eficiente que el actual?

1 2 3 (4) 5

2. Si tuviera la posibilidad de elegir entre utilizar **FlocOS** o el proceso manual para administrar las lecturas de los estanques, ¿elegiría **FlocOS**?

1 2 3 4 (5)

3. ¿Considera que el manejo de **FlocOS** es amigable, fácil de utilizar?

1 2 3 (4) 5

4. Comparando el tiempo que gasta administrando las lecturas con **FlocOS** con el tiempo que gasta actualmente realizando el proceso de forma manual, ¿considera que es más rápido el proceso por medio de **FlocOS**?

1 2 3 4 (5)

Duración del proceso realizado en **FlocOS** 1 minutos.

5. ¿Hay algún aspecto o sugerencia que le gustaría compartir?

SI x NO

¿Cuál?

Que el software se comunique
directamente con los sensores y
se monitoree los variaciones diarias.

Gracias por responder esta encuesta

mf

1.3.4.3 Implementación

Retomando los conocimientos obtenidos en la fase 2 de la metodología DSR se procedió a implementar el artefacto de software dentro de la plataforma de Hardware libre.

Inicialmente, se instaló un sistema operativo tipo Unix basado en Debían, esta distribución fue creada específicamente para ésta plataforma y se obtuvo del repositorio propio de BeagleBoard. Seguidamente, se instaló el framework Ruby on Rails sobre la plataforma, se escribieron algunos scripts en bash para la inicialización de la conexión Wifi-Direct y la puesta en marcha de Ruby on Rails como un demonio.

Se configuró la conexión de la Tablet a través del punto de acceso proporcionado por la plataforma y finalmente se procedió a visualizar la solución de software desde el navegador web de la Tablet. Para ver las pruebas de rendimiento de esta implementación diríjase al Anexo X. Pruebas de rendimiento BeagleBone.



Figura 33. Modelo real de interacción entre Hardware y Software.

1.3.4.4 OWASP TOP

Todo el desarrollo contempló el TOP10 del OWASP, el cual contiene las 10 vulnerabilidades más recurrentes en desarrollo de sistemas de información web. Ruby on Rails previene y soluciona algunas de las vulnerabilidades, para las demás fue necesario implementar ciertos procedimientos. A continuación se presentan las principales:

Inyección

Ruby on Rails previene la inyección de SQL a través de “Strong params”, esto permite validar la entrada de datos enviados por POST, GET, o PUT antes de realizar la operación en la base de datos.

```
def variable_params
  params.require(:variable).permit(:NombreVariable, :ValorMaximo, :ValorMinimo)
end

def create
  @variable = current_pond.variables.new(variable_params)
  if @variable.save
    redirect_to @variable
  else
    render :new
  end
end
```

Figura 34. Strong params de RoR.

Pérdida de autenticación y gestión de sesiones:

Devise previene casi por completo ésta vulnerabilidad, utilizando cookies en lugar de variables de sesión, adicionalmente restringe el acceso a las diferentes clases del sistema si no se está autenticado. El robo de cookie, puede venir por otro tipo de vulnerabilidades como por el ejemplo la vulnerabilidad XSS.

Configuración de seguridad incorrecta:

Un buen manejo de seguridad depende de múltiples factores, un adecuado manejo de sesiones, gestión de usuarios por defecto en servidores, autorización de acceso a objetos entre otros, son prácticas que mejoran considerablemente la seguridad en una aplicación web, no obstante los frameworks no llegan hasta este nivel, dado que depende más del conocimiento y la experiencia del equipo desarrollador, a continuación se presenta una validación de sesión actual implementada en un “Helper” de Rails.

```
def current_variable_pond(variable)
  current_pond.id == variable.pond_id
end
```

Figura 35. Configuración de autorización básica.

Este pequeño método permite que controlar el acceso a las variables que pertenecen a un estanque, de ésta manera, no se puede navegar por las variables de otro estanque intentando modificar el parámetro asociado a la ruta. Para ver todas las vulnerabilidades del TOP10 y la matriz de vulnerabilidades asociada a éste proyecto diríjase al [anexo 8.4](#) “Matriz de vulnerabilidades”.

4. RESULTADOS

- Se implementó un aplicativo web que permite administrar las variables fundamentales en un estanque piscícola y generar reportes.
- Se desarrolló una solución de TI completamente portable, utilizable en cualquier sitio y sin necesidad de infraestructura de red.
- Se cumplió el TOP10 del OWASP 2013 en un 75% lo cual proporciona un buen nivel de seguridad y confiabilidad del aplicativo web.
- Con el desarrollo del sistema de información web FlocOs, se mejoró el proceso manual de la administración de las variables fundamentales de un estanque en el IALL.
- Se escribió un artículo científico donde se resumen los aspectos importantes de este desarrollo

5. CONCLUSIONES

- La metodología Design Science Research fortalece los procesos de desarrollo de soluciones de TI dividiendo el proceso en tres etapas: entorno, conocimiento y desarrollo.
- Los investigadores del IALL sugirieron la comunicación de la red de sensores inalámbricos y éste aplicativo web para optimizar completamente el monitoreo de las variables fundamentales de cualquier estanque piscícola.
- Esta solución de TI aunque se basó completamente en el sistema biofloc es absolutamente implementable en cualquier otro tipo de cultivo piscícola.
- La plataforma de Hardware libre sacrifica un poco de rendimiento por portabilidad, sin embargo, si se ubica un enrutador en el sitio de implementación la velocidad de las peticiones en el software mejora, dado que la tarjeta ya no tiene que computar el enrutamiento.
- El OWASP solo contempla seguridad en aplicativos web, por consiguiente no se contempló la seguridad física de la implementación, la transmisión de datos entre la plataforma de Hardware y el dispositivo que se conecte no está cifrada.

6. TRABAJOS FUTUROS

- El sistema puede complementarse desarrollando un algoritmo para predecir el comportamiento de las variables en un estanque. Así pues, se logrará obtener un máximo control y estudiar los factores que posiblemente estén afectando el funcionamiento habitual del mismo.
- Se pueden implementar otras opciones al sistema de información, para trabajar con un conjunto de factores piscícolas que permitan un mejor análisis, y en consecuencia, un reporte más completo.
- La herramienta FlocOS puede convertirse en un software para labores cotidianas. en consecuencia, se puede desarrollar un conjunto de alertas para hacerle saber al piscícola en qué momento se está presentando la anomalía y reaccionar de inmediato, ya que de ésta depende que las especies alojadas en el estanque sigan con vida

7. REFERENCIAS

ACUICULTURA PERU. [En línea]. 19 de Marzo de 2014. [Citado el: 27 de Julio de 2017.] Disponible en: <http://acuiculturaperu.blogspot.com.co/2014/03/aquasoft-el-software-que-facilitara-la.html>.

AKVA group. [En línea]. 2015. [Citado el: 27 de Julio de 2017.] Disponible en: <http://www.akvagroup.com/p-gina-de-inicio>.

AQUA. 2016. AQUA. [En línea] 13 de Octubre de 2016. [Citado el: 27 de Julio de 2017.] <http://www.aqua.cl/informes-tecnicos/uso-software-acuicultura-produccion-control/>

AQUACRIA. [En línea]. 2016. [Citado el: 8 de Junio de 2017]. Recuperado de: <http://www.aquacria.com/articulos/9/biofloc-tecnologia-acuicola>.

ARIAS, Miguel. Webs Responsivas. Responsive Design con Bootstrap [En línea]. 2014. Recuperado de: <https://books.google.com.co/books?id=J8lJDQAAQBAJ>

BEAGLEBOARD.ORG FOUNDATION. [En línea] [Citado el: 30 de Julio de 2017]. Recuperado de: <https://beagleboard.org/>

COLLAZOS, Luis & ARIAS, José. Orinoquia Unillanos. [En línea]. 2015. [Citado el: 23 de julio de 2017]. Recuperado de: <http://orinoquia.unillanos.edu.co/index.php/orinoquia/article/view/341/934>.

MONROY, Maria. Composición y abundancia de comunidades microbianas asociadas al biofloc en un cultivo de tilapia. Junio de 2016, Revista de biología marina y oceanografía, pp. 511-520.

REYES, Ricardo & DIAZ, Manuel. Desarrollo de un software basado en modelos para redes inalámbricas de sensores [En línea]. Junio de 2016, Revista Académica de TLATEMOANI.

DIMES, Troy. 2015. JavaScript Una Guía de Aprendizaje para el Lenguaje de Programación JavaScript. Babelcube Incorporated, 2015.

EcuRed. EcuRed. [En línea] s.f. [Citado el: 23 de Julio de 2017]. Recuperado de: https://www.ecured.cu/Instituto_de_Acuicultura_de_Los_Llanos.

FAO. 2017. Organización de las Naciones Unidas para la Alimentación y la Agricultura [En línea]. 2017. [Citado el: 29 de Julio de 2017]. Recuperado de: <http://www.fao.org/aquaculture/es/>.

HEVNER, Alan y CHATTERJEE, Samir. Design research in information systems: theory and practice. Londres: Springer Science & Business Media, 2010. Vol. 22.

KEHOE, Daniel. Learn Ruby on Rails [En línea]. 2016. Recuperado de:
<https://books.google.com.co/books?id=tvazDQAAQBAJ>

JOHNSON, D.L. Meanings of environmental terms. 1997, Journal of Environmental Quality, pp. 581-589.

LUJAN, Milthon. Portal del Agro. [En línea]. Recuperado de:
http://www.portaldelagro.com/index.php?option=com_content.

PISCICOLA EL MANTIAL. Piscicola el Mantial. [En línea] 2016. [Citado el: 30 de Julio de 2017]. Recuperado de: <http://www.piscicolaelmanatial.com/content/quienes-somos>.

REDUSERS. Diseño Web con HTML y CS. s.l. : Creative Andina Corporation, 2012. pág. 192. Vol. 26.

SISTEMAS ACUICOLAS. [En línea]. 2017. [Citado el: 23 de Julio de 2017.]. Recuperado de: <https://sistemasacuicolas.com/biofloc-solucion-sustentable-sistemas-acuicolas>.

StarUML. s.f.. StarUML. [En línea] s.f. [Citado el: 28 de Julio de 2017]. Recuperado de: <http://staruml.sourceforge.net/v1/about.php>.

STEPHENS, Rod. Beginning Software Engineering [En línea]. 2015. Recuperado de: <https://books.google.com.co/books?id=uiHWBgAAQBAJ>

The OWASP Foundation. OWASP Foundation. [En línea] 2013. [Citado el: 30 de Julio de 2017]. Recuperado de: https://www.owasp.org/images/5/5f/OWASP_Top_10_-_2013_Final_-_Espa%C3%B1ol.pdf.

W3schools. [En línea] [Citado el: 29 de Julio de 2017]. Recuperado de: https://www.w3schools.com/css/css_intro.asp.

WANUMEN, Luis. Estrategia arquitectónica de interacción entre simuladores y sig para visualizar zonas vulnerables en crisis [En línea]. Pontificia Universidad Javeriana. Bogota, D.C. : s.n., 2012. [Tesis]. Recuperado de <https://repository.javeriana.edu.co/bitstream/handle/10554/15071/WanumenSilvaLuisFelipe2012.pdf?sequence=1&isAllowed=y>

YADIN, Aharon. Computer Systems Architecture [En línea]. Taylor & Francis group, 2016. pág. 467. Recuperado de: <https://books.google.com.co/books?id=yMDBDAAAQBAJ>

8. ANEXOS

8.1 Prototipos no funcionales

Ver (Informe/Anexos_8.1)

8.2 Diagramas UML

Ver (Informe/Anexos_8.2)

8.3 Encuesta de satisfacción del uso de FlocOS

Ver (Informe/Anexo_8.3)

8.4 Matriz de vulnerabilidades

Ver (Informe/Anexo_8.4)

8.5 Acta de reunión N°4 con el Instituto de Acuicultura de los Llanos

Ver (Informe/Anexo_8.5)

8.6 Artículo científico denominado “Herramienta de software para graficar variables obtenidas en un sistema de biofloc en piscicultura”.

Ver (Articulo/Anexo 8.6)

8.7 OWASP

Ver (Informe/Anexo_8.7)

8.8 Pruebas

Ver (Informe/Anexo_8.8)

8.9 Manual técnico

Ver (Manuales/Manual técnico)

8.10 Manual de usuario

Ver (Manuales/Manual de usuario)