DISEÑO DE UN PROTOTIPO DE SISTEMA WEB PARA EL MONITOREO Y CONTROL DE LA CALIDAD DEL AGUA EN LOS ESTANQUES DE CULTIVO DE TILAPIAS EN LA FINCA ALLEGRO EN SAN FRANCISCO CUNDINAMARCA.

Director: Nancy Yaneth Gelvez García

Revisor: Oswaldo Alberto Romero Villalobos

Nicolás Andrés Arias Sabogal & Iván Camilo Mendoza Gutiérrez.

Noviembre 2018.

Universidad Distrital Francisco José de Caldas.

Facultad de Ingeniería.

Especialización en Ingeniería de Software

TABLA DE CONTENIDOS

INTRODUCCION 1

PARTE I. CONTEXTUALIZACIÓN DE LA INVESTIGACIÓN 3

DESCRIPCIÓN DE LA INVESTIGACIÓN 3

1.1 PLANTEAMIENTO DEL PROBLEMA 3

1.2 OBJETIVOS 3

1.2.1 Objetivo General 3

1.2.2 Objetivos específicos 4

1.3 JUSTIFICACIÓN DE LA INVESTIGACIÓN 4

1.4 HIPÓTESIS 5

1.5 MARCO REFERENCIAL 5

1.5.1 Marco Teórico 5

1.5.2 Marco Conceptual 17

1.5.3 Marco Espacial 18

1.6 ASPECTOS METODOLÓGICOS 19

1.6.1 Tipo de estudio 19

1.6.2 Método de Investigación 19

1.6.3 Fuentes y técnicas para la recolección de la información 19

1.6.4 Tratamiento de la información 20

1.7 ALCANCES, LIMITACIONES Y RESULTADOS ESPERADOS 20

1.7.1 Alcances 20

1.7.2 Limitaciones 21

1.7.3 Resultados esperados 21
PARTE II. DESARROLLO DE LA INVESTIGACIÓN 22
ARQUITECTURA EMPRESARIAL 22
2.1 FUNDACIÓN FINCA ALLEGRO 22
2.1.1 Misión 22
2.1.2 Visión 22
2.1.3 Objetivos Organizacionales 22
2.1.4 Organigrama 23
2.1.5 Procesos del negocio 23
2.1.6 Productos y Servicios 24
2.2 Archimate 24
2.3 CAPA DE NEGOCIO 25
2.3.1 Punto de vista de organización 26
2.3.2 Punto de Vista de Cooperación de Actor 26
2.3.3 Punto de Vista de Función Negocio 27
2.3.4 Punto de Vista de Proceso 28
2.3.5 Punto de Vista de Cooperación de Proceso 29
2.3.6 Punto de Vista de Producto 29
2.4 CAPA DE APLICACIÓN 30
2.4.1 Punto de Vista de Comportamiento 30
2.4.2 Punto de Vista de Cooperación de Aplicación 32
2.4.3 Punto de Vista de Estructura de Aplicación 33
2.4.4 Punto de Vista de Uso de Aplicación 33

2.5 CAPA DE TECNOLOGÍA..... 34
2.5.1 Punto de Vista de Infraestructura 34
2.5.2 Punto de Vista de Uso de Infraestructura 35
2.5.3 Punto de Vista de Implementación 36
2.5.4 Punto de Vista de Estructura de la Información..... 37
2.5.5 Punto de Vista de Realización del Servicio 38
2.5.6 Punto de Vista de Capas 39
2.6 CAPA DE MOTIVACIÓN 41
2.6.1 Punto de Vista de Stakeholder 41
2.6.2 Punto de Vista de Realización del servicio 42
2.6.3 Punto de Vista de Contribución de Objetivos 43
2.6.4 Punto de Vista de Principios..... 44
2.6.5 Punto de Vista de Realización de Requerimientos..... 45
2.6.6 Punto de Vista de Motivación 46
2.7 CAPA DE MIGRACIÓN Y DESPLIEGUE 47
2.7.1 Punto de Vista de Proyecto 48
2.7.2 Punto de Vista de Migración..... 49
2.7.3 Punto de Vista de migración e implementación..... 49
METODOLOGÍA..... 51
3.1 ESPECIFICACIÓN 53
3.1.1 Análisis de requerimientos..... 53
3.2 DISEÑO Y DESARROLLO 60
3.2.1 TilapiappWeb..... 60

3.2.2 TilapiappPi 72
3.3 DESPLIEGUE 78
3.3.1 Tilapiapp Web 78
3.3.2 Tilapiapp PI 84
3.4 PRUEBAS 95
3.4.1 Información base – estado previo a la implementación. 95
3.4.2 Pruebas de funcionalidad 97
PARTE III. CIERRE DE LA INVESTIGACIÓN 107
CONCLUSIONES, TRABAJOS, FUTUROS APORTES, Y CONTRASTACIÓN. 107
4.1 CONCLUSIONES 107
4.2 TRABAJOS FUTUROS 108
4.2.1 Ejecutar acciones remotamente 108
4.2.2 Alertas Visuales – Tilapiapp VA (Visual Alert) 108
4.2.3 Alertar por medio de SMS 109
4.3 APORTES 109
4.4 CONTRASTACIÓN 110
ANEXOS 111
A. ENTREVISTAS 111
B. CASOS DE USO 114
C. CARTA AUTORIZACIÓN USO SENSONRES 139
REFERENCIAS 140

LISTA DE TABLAS

Tabla 1 Variables de medición en el estanque 7

Tabla 2 Dominios de aplicación de Software 9

Tabla 3 Listado Requerimientos TilapiappWeb - Fuente: Los atuores 56

Tabla 4 Listado Requerimientos TilapiappPi - Fuente: Los autores 57

Tabla 5 Listado de requerimientos Tilapiapp Pi - Fuente: Los autores 58

Tabla 6 Registros de Oxigeno promediados por hora del tanque norte el día 27 de octubre 2018 - Fuente: Los Autores 103

LISTA DE FIGURAS

Imagen 1 Producción de Tilapias. Fuente: [1] 6

Imagen 2 Tanque de crianza. Fuente Internet 8

Imagen 3 Ciclo de vida del Software. Fuente: Los autoress 11

Imagen 4 Modelo en Cascada. Fuente [4] 12

Imagen 5 Modelo en V Fuente [4] 12

Imagen 6 Modelo Incremental Fuente [4] 13

Imagen 7 Modelo por prototipos Fuente [4] 13

Imagen 8 Modelo en Espiral Fuente [4] 13

Imagen 9 Arquitectura de Software Fuente Internet 15

Imagen 10 Diagrama Arquitectura Microservicios. Fuente Microsoft 15

Imagen 11 Organigrama Fundación Finca Allegro – Fuente: Fundación Finca Allegro 23

Imagen 12 Correlación entre lenguaje ArchiMate y TOGAF ADM [12] 25

Imagen 13 Punto de vista de organización - Fuente: Los autores 26

Imagen 14 Punto de vista de cooperación de actor - Fuente: Los autores 27

Imagen 15 Punto de Vista de Función Negocio – Fuente: Los Autores 28

Imagen 16 Punto de Vista de Proceso – Fuente: Los Autores 28

Imagen 17 Punto de Vista de Cooperación de Proceso – Fuente: Los Autores 29

Imagen 18 Punto de Vista de Producto – Fuente: Los Autores 30

Imagen 19 Punto de Vista de Comportamiento – Fuente: Los Autores 31

Imagen 20 Punto de Vista de Cooperación de Aplicación 32

Imagen 21 Punto de Vista de Estructura de Aplicación 33

Imagen 22 Punto de Vista de Uso de Aplicación – Fuente: Los Autores 34

Imagen 23 Punto de Vista de Infraestructura – Fuente: Los Autores.....35
Imagen 24 Punto de Vista de Uso de Infraestructura – Fuente: Los Autores..... 36
Imagen 25 Punto de Vista de Implementación – Fuente: Los Autores..... 37
Imagen 26 Punto de Vista de Estructura de la Información – Fuente: Los Autores 38
Imagen 27 Punto de Vista de Realización del Servicio – Fuente: Los Autores..... 39
Imagen 28 Punto de Vista de Capas – Fuente: Los Autores..... 40
Imagen 29 Punto de Vista de Stakeholder – Fuente: Los Autores..... 42
Imagen 30 Punto de Vista de Realización del servicio – Fuente: Los autores..... 43
Imagen 31 Punto de Vista de Contribución de Objetivos 44
Imagen 32 Punto de Vista de Principios – Fuente: Los Autores 45
Imagen 33 Punto de Vista de Realización de Requerimientos – Fuente: Los Autores . 46
Imagen 34 Punto de Vista de Motivación – Fuente: Los Autores..... 47
Imagen 35 Punto de Vista de Proyecto – Fuente: Los Autores..... 48
Imagen 36 Punto de Vista de Migración – Fuente: Los Autores 49
Imagen 37 Punto de Vista de migración e implementación – Fuente: Los Autores 50
Imagen 38 Diagrama de casos de uso – Fuente: Los Autores..... 55
Imagen 39 Modelo relacional Tilapiapp Web – Fuente: Los autores..... 60
Imagen 40 Listado de tablas directamente en la base de datos - Fuente: Los Autores 62
Imagen 41 Listado de clases de Visual Studio correspondientes a la capa de datos – Fuente: Los Autores..... 63
Imagen 42 Diagrama de clases correspondientes a las interfaces en el patrón de fabricación – Fuente: Los Autores..... 64

Imagen 43 Listado de clases e interfaces en la capa de negocio de Tilapiapp Web - Fuente : Los Autores..... 65
Imagen 44 Scketch login de usuario – Fuente: Los Autores 67
Imagen 45 Scketch Listado de usuarios del sistema – Fuente: Los Autores 67
Imagen 46 Formulario de creación / edición de usuarios - Fuente: Los Autores..... 68
Imagen 47 Scketch Listado de tanques - Fuente: Los Autores 68
Imagen 48 Scketch Formulario de creación / edición de tanques – Fuente: Los Autores 68
Imagen 49 Scketch Listado de parámetros de tanques - Fuente: Los Autores 69
Imagen 50 Scketch Formulario de creación / edición de parámetros de tanques 69
Imagen 51 Scketch Dashboard de tanques - Fuente: Los Autores 70
Imagen 52 Scketch representación gráfica - Fuente: Los Autores 70
Imagen 53 Scketch Tabulación de resultados - Fuente: Los Autores 70
Imagen 54 Scketch Listado de vistas en el proyecto MVC..... 71
Imagen 55 Scketch Listado de vistas para el controlador personas – Fuente: Los Autores 72
Imagen 56 Árbol de tablas SensoresDB - Fuente: Los Autores 74
Imagen 57 Modelo de paquetes Tilapiapp PI - Fuente: Los autores 74
Imagen 58 Arbol explorador de solución Tilapiapp Pi - Fuente: Los Autores 76
Imagen 59 Diagrama de conexión sensor y tarjeta - Fuente: Los Autores..... 77
Imagen 60 Listado de capas del proyecto TilapiappWeb - Fuente: Los Autores..... 78
Imagen 61 Menú de opciones de la capa de presentación de Tilapiapp web – Fuente: Los autores 79

Imagen 62 Archivo de publicación en Visual Studio 2017- Fuente: Los autores. 80
Imagen 63 Proceso de publicación en ejecución- Fuente: Los autores 80
Imagen 64 Listado de archivos generados por el proceso de publicación - Fuente: Los autores. 81
Imagen 65 Internet Information Services - Fuente: Los Autores 81
Imagen 66 Formulario de creación de nuevo sitio web en IIS - Fuente: Los autores 82
Imagen 67 Pantalla e Login de Tilapiapp Web – Fuente: Los Autores. 82
Imagen 68 Cadena de conexión de TilapiappWeb a la base de datos SQL Server - Fuente: Los Autores 83
Imagen 69 Listado de tablas y objetos creados en la base de datos de Tilapiapp a través de Entity Framework Core - Fuente: Los Autores 84
Imagen 70 Configuración Raspberry Pi 01 - Fuente: Los Autores 85
Imagen 71 Configuración Raspberry Pi 02 - Fuente: Los Autores 85
Imagen 72 Prueba conexión SSH Raspberry Pi 01 - Fuente: Los Autores 86
Imagen 73 Prueba conexión SSH Raspberry Pi 02 - Fuente: Los Autores 86
Imagen 74 Configuración MySQL Raspberry Pi 01 - Fuente: Los Autores 87
Imagen 75 Configuración MySQL Raspberry Pi 02 - Fuente: Los Autores 88
Imagen 76 Configuración MySQL Raspberry Pi 03 - Fuente: Los Autores 88
Imagen 77 Configuración MySQL Raspberry Pi 04 - Fuente: Los Autores 89
Imagen 78 Configuración DB SensoresDB 01 - Fuente: Los Autores 90
Imagen 79 Configuración DB SensoresDB 02 - Fuente: Los Autores 90
Imagen 80 Configuración DB SensoresDB 03 - Fuente: Los Autores 90
Imagen 81 Instalación Node.js 01 - Fuente: Los Autores 91

Imagen 82 Instalación Node.js 02 - Fuente: Los Autores 91
Imagen 83 Instalación Node.js 03 - Fuente: Los Autores 92
Imagen 84 Visualización de los sensores de medición en un estanque en la finca Allegro – Fuente: Finca Allegro 96
Imagen 85 Evidencia de muerte de peces debido a la falta de oxígeno – Fuente: Finca Allegro 97
Imagen 86 Conexión tarjeta de sensores con RaspberryPi – Fuente: Los Autores 98
Imagen 87 Cardumen de tilapias en el Tanque norte - Fuente: Finca Allegro 99
Imagen 88 Listado de parámetros asociados al tanque norte en Tilapiapp - Fuente: Los Autores 99
Imagen 89 Listado de usuarios creados en Tilapiapp 100
Imagen 90 Dashboard Tilapiapp presentación de datos del sensor - Fuente: Los Autores 100
Imagen 91 Informe de eventos en Tilapiapp – Fuente: Los Autores 101
Imagen 92 Email de notificación de Tilapiapp por nivel bajo de oxigeno - Fuente: Los Autores 101
Imagen 93 Email de notificación de Tilapiapp por nivel bajo de oxigeno (2) - Fuente: Los Autores 102
Imagen 94 Tanque Norte con los sistemas de aireación activados - Fuente: Los autores 102
Imagen 95 Histórico de medicines de oxígeno en el tanque norte - 27 de octubre Fuente: Los Autores 105

Imagen 96 Visualización de nivel inferior al umbral parametrizado en para el tanque norte. Fuente: Los Autores 105

INTRODUCCION

Esta investigación tiene como fin desarrollar una herramienta haciendo uso de tecnologías web y del concepto de internet de las cosas para automatizar el proceso de medición de la calidad del agua en los tanques de cultivos de tilapias de la finca *Allegro*.

El concepto de internet de las cosas ha permitido en la actualidad llevar a cabo muchos tipos de automatización a diferentes procesos ya que permite combinar uno de los recursos más grandes que tenemos actualmente como el internet con dispositivos electrónicos de bajo consumo, los cuales pueden ser aplicados en diferentes áreas del conocimiento como es el caso del cultivo de tilapias.

En el proceso de cultivo de tilapias intervienen varios factores para lograr un proceso exitoso, como la alimentación de los peces y la calidad del agua de los tanques donde se encuentran los animales. Esta calidad del agua debe mantenerse bajo los rangos nominales y estables según su edad y tamaño para evitar que algún tipo de estrés o enfermedad afecte a los animales y se produzca su muerte.

La investigación de esta problemática se realizó con el fin entender el proceso de producción de tilapias y como el entorno y la calidad del agua afectan directamente el crecimiento y calidad del animal, esto permitió entender la necesidad de una solución tecnológica para automatizar un proceso riguroso expuesto a fallos.

En el contenido de este trabajo se presenta el diseño y desarrollo de un sistema multifuncional que permite precisamente materializar una automatización directa del proceso de monitoreo de la calidad del agua, que a su vez permitirá profundizar no solo en el análisis de los históricos de datos del proyecto sino en el proceso global de cultivo de tilapia en Colombia.

PARTE I. CONTEXTUALIZACIÓN DE LA INVESTIGACIÓN

DESCRIPCIÓN DE LA INVESTIGACIÓN

1.1 PLANTEAMIENTO DEL PROBLEMA

En los últimos 10 años la exportación de tilapia roja en Colombia ha crecido 8 veces, lo que se ha empezado a ver como un negocio rentable en el país, sin embargo, actualmente los proyectos de cultivos de tilapias presentan diferentes inconvenientes tanto operativos como de producción, uno de los principales problemas es la muerte de los propios animales por no ejecutar una rápida acción correctiva frente a los cambios abruptos del medio, ya sea la falta de oxígeno disuelto en el agua, cambios en temperatura o niveles de pH fuera de los umbrales. Estas acciones correctivas no se pueden realizar si no existe el personal dedicado 24/7 a la medición y análisis de la calidad del agua, por lo que la dependencia de personal humano en el proceso de producción de tilapias es indispensable.

1.2 OBJETIVOS

1.2.1 Objetivo General

Diseñar un prototipo de sistema web que permita la integración, monitoreo y control de los sensores encargados de la lectura de la calidad del agua en el cultivo de tilapias de la finca Allegro, haciendo uso del concepto de internet de las cosas.

1.2.2 Objetivos específicos

- Determinar la dinámica, las variables y las reglas de negocio que intervienen en el proceso de monitoreo de la calidad del agua haciendo uso de conceptos y procesos de ingeniería de software buscando su aplicación en el diseño y desarrollo del sistema web.
- Diseñar los patrones funcionales y vistas del sistema, haciendo uso de arquitectura web basada en microservicios, con el propósito de tener las bases para la integración con los sensores de medición.
- Desarrollar un prototipo de sistema web basado en la arquitectura y requerimientos definidos, haciendo uso de tecnologías web .Net Core que permitan registrar los datos de los sensores integrados al sistema, con el fin de monitorear la calidad del agua.

1.3 JUSTIFICACIÓN DE LA INVESTIGACIÓN

Las enfermedades y el estrés provocado por las variaciones de la calidad del agua a las que están expuestas las tilapias han generado la necesidad de analizar el comportamiento de este fenómeno en el proyecto de cultivo de tilapias de San Francisco Cundinamarca.

Para realizar un análisis adecuado es necesario disponer de información confiable y efectiva, por lo cual realizar un sistema web que permita la adaptación y el monitoreo de variables de calidad de agua se hace evidente, dado que la medición y registro de la fluctuación de la temperatura, el oxígeno disuelto, la

saturación de oxígeno y el pH, son indispensables en un análisis que permita de manera conjunta, automatizar los procesos de monitorio y generar planes de contingencia en el cultivo de tilapias, todo a través de un mismo sistema.

Otro punto de vista en torno al desarrollo del sistema es la transferencia de tecnología que juega un papel importante; Países como Colombia dependen de tecnología extranjera haciendo necesario importar nuevas tecnologías, esto genera que Colombia sea un país con bajo desarrollo industrial, el desarrollo de un sistema de integración y monitorio innovaría la industria colombiana dando pie para que a futuro se puedan desarrollar nuevos sistema y nuevos procesos.

1.4 HIPÓTESIS

“A través de la implementación de un sistema web de integración y monitoreo de la calidad del agua, el usuario podrá tener acceso remoto a la información de su cultivo y tendrá la posibilidad de crear planes de acción frente a las alertas generadas por el sistema web relacionadas con contingencias que atenten contra el proceso de producción de tilapias”.

1.5 MARCO REFERENCIAL

1.5.1 Marco Teórico

El desarrollo del proyecto se realizará teniendo en cuenta la siguiente documentación relacionada tanto como el cultivo de tilapias como con desarrollo y arquitectura de software.

1.5.1.1 Cultivo de Tilapias

De acuerdo de un informe del Departamento Administrativo Nacional de Estadística - DANE [1] en el 2014, la cultura de cultivo de peces nace en 1938 con la trucha arcoíris, en donde los campesinos se dieron cuenta del potencial de piscicultura comercial, lo que impulso la llegada de nuevas especies dentro de ella la tilapia roja o conocida como mojarra roja.

La mojarra roja es un hibrido resultante del cruce de varias especies originarias de África e Israel, con un amplio margen de ganancia frente a otras especies. [1]

En la siguiente gráfica se observa la relación de producción de tilapias tanto rojas como plateadas por lo métodos de estanques y jaulas.

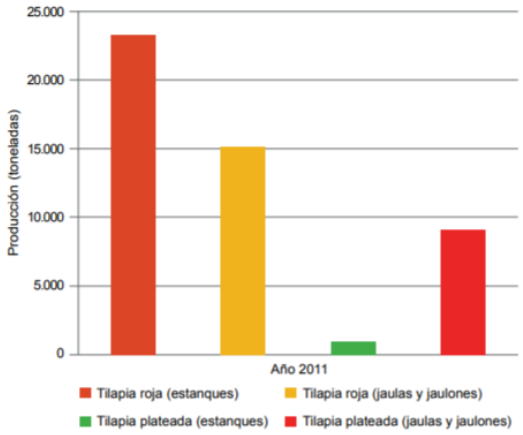


Imagen 1 Producción de Tilapias. Fuente: [1]

Sin embargo, y al tratarse de seres vivos, se necesita tener controles que garanticen esencialmente la vida de los peces durante todo el proceso de crianza. Por lo que se hace necesario tener presente los siguientes factores [2] que serán

de análisis para conocer si las condiciones en el ambiente, en este caso el estanque, se encuentran optimas o si por el contrario hay presencia de riesgo que puede desencadenar más adelante efectos que afecten la producción de cada estanque:

Tabla 1 Variables de medición en el estanque

Variable	***Descripción***
Temperatura	Los rangos óptimos de temperatura oscilan entre 20°C y 30°C. A temperaturas menores de 15°C no crecen.
Oxígeno Disuelto	Se recomiendan valores de 2 o 3 mg/l, incluso pueden soportar niveles de 1 mg/l, pero a menor nivel de oxígeno, menor nivel de alimento y por ende menor crecimiento.
pH – Acidez	Los valores óptimos se encuentran entre 7 y 8. No resisten valores ácidos menores a 5, pero pueden resistir valores alcalinos de hasta 11.
Turbidez	Se deben mantener 30 cm de visibilidad.
Altitud	Entre 850 y 2.000 metros sobre el nivel del mar.
Saturación de Oxígeno	La saturación se debe encontrar en 5 ppm.

Fuente: Los autores



Imagen 2 Tanque de crianza. Fuente Internet

1.5.1.2 Software

La definición dada por la RAE para la palabra *software* es la siguiente:

"Conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora." [3]

No obstante, Roger Pressman afirma que pueden darse definiciones más complejas para este término, pero aun así dichas definiciones no pueden mejorar la manera de comprenderlo. Es por eso por lo que él indica que el software posee 3 características que son [4]:

- El software se desarrolla o modifica con intelecto
- El software no se "desgasta"
- Aunque la industria se mueve hacia la construcción basada en componentes, la mayor parte del software se construye para un uso individualizado.

Asimismo, Pressman [4] categoriza el software en siete grandes grupos relacionados en la siguiente tabla:

Tabla 2 Dominios de aplicación de Software

Dominio	***Descripción***
Software de Sistemas	Es aquel software que da soporte a otro software como lo son por ejemplo los controladores, compiladores, editores o software para comunicaciones. Se caracteriza por gran interacción con el hardware
Software de Aplicación	Son aquellos programas que resuelven una necesidad puntual de un negocio, como un sistema de nómina. Facilitan la toma de decisiones.
Software de Ingeniería y Ciencias	Aplicaciones destinadas al campo de la investigación. Se les conoce como algoritmos “devoradores de números”. Como por ejemplo simulaciones de fluidos, de aerodinámica, control de un transbordador.
Software Incrustado	Es el software que se encuentra dentro de un producto final como por ejemplo una lavadora, y permite la interacción del usuario con el producto.

Software de línea de productos	Son aplicaciones creadas para dar solución a un problema puntual de un grupo común de usuarios como por ejemplo los sistemas SAP, de control de inventarios, suite ofimática.
Aplicaciones Web	También conocidos como *Webapps*, son todos aquellos programas que trabajan bajo el protocolo de red, caracterizadas por el uso de la hipermedia. Cada vez son más sofisticadas.
Software de Inteligencia Artificial	Aplicaciones que no hacen uso de algoritmos numéricos y que tratan complejidades muy altas como por ejemplo en la robótica, sistemas expertos, redes neurales artificiales o juegos.

Fuente: Los autores

1.5.1.3 Metodologías de Desarrollo de Software

Las metodologías de desarrollo de software tienen como principio base el ciclo de vida de software (Ver imagen 3), aun así, y debido a la dinámica a la que se ha enfrentado la Ingeniería de Software se han ido desarrollando diferentes métodos que permiten resolver un problema en concreto o ser aplicados de acuerdo a ciertos criterios que la práctica ha encontrado y que pueden ser de ayuda al momento del desarrollo de software.

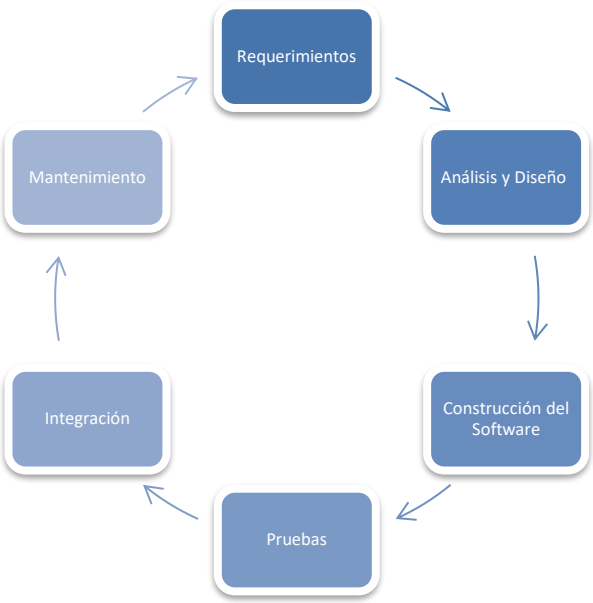


Imagen 3 Ciclo de vida del Software. Fuente: Los autoress

De ahí que, se hayan desarrollado las siguientes metodologías:

- Cascada
- Modelo en V
- Iterativos
- Prototipos
- Evolutivos
- Espiral
- Concurrentes
- Componentes
- Agiles

Para cada una de las anteriores metodologías, se realizan diferentes procesos que va desde el levantamiento de requerimientos hasta la puesta en producción y

mantenimiento, por lo que cada una puede ser más extensa que la otra o sencillamente, no ser la correcta para un pequeño o gran proyecto. De ahí nace la importancia que tener un contexto y una buena abstracción del problema a resolver.

Hoy día el uso de metodologías ágiles se encuentra en auge y esto es debido a su eficiencia al momento de realizar entregas periódicas al cliente y así poder realizar correcciones a tiempo para tener productos listos en un menor tiempo.

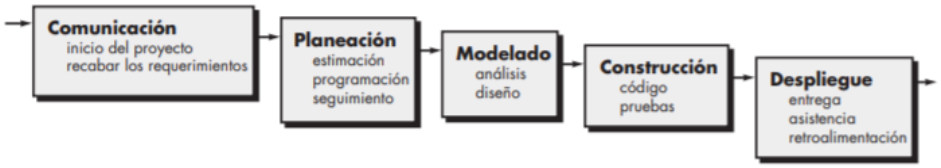


Imagen 4 Modelo en Cascada. Fuente [4]

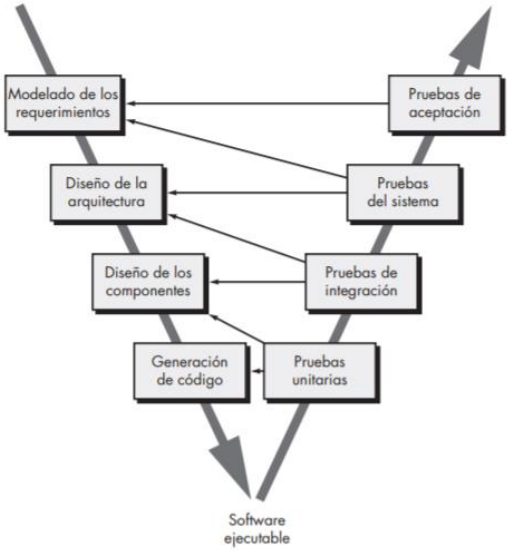


Imagen 5 Modelo en V Fuente [4]

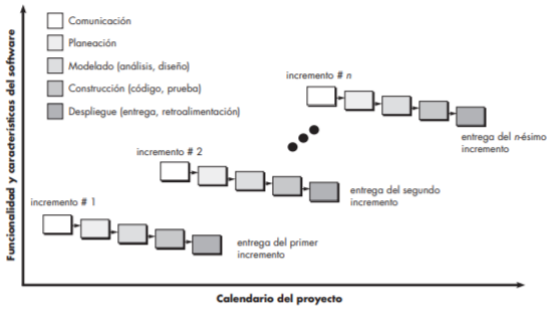


Imagen 6 Modelo Incremental Fuente [4]

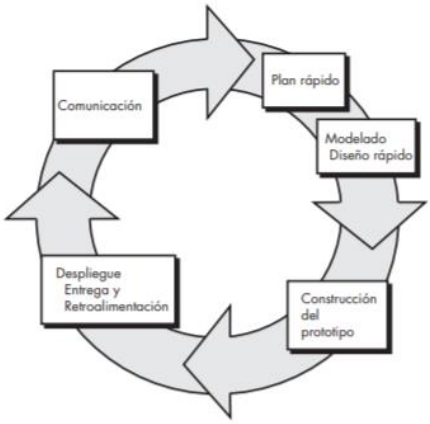


Imagen 7 Modelo por prototipos Fuente [4]

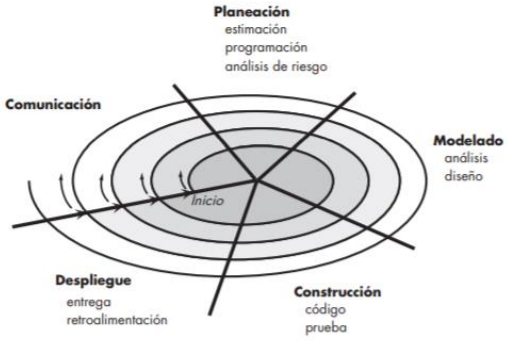


Imagen 8 Modelo en Espiral Fuente [4]

1.5.1.4 Arquitectura de Software

La IEEE Std 1471-2000 da la siguiente definición al concepto de Arquitectura de Software:

"La Arquitectura de Software es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución."

Es considerada el nivel más alto del desarrollo de una solución, pues establece la estructura, funcionamiento e interacción entre las partes del software.

La arquitectura de Software se compone por:

- Clientes y servidores
- Bases de datos
- Filtros
- Niveles de jerarquía del sistema

En vista de que la arquitectura de software es la columna vertebral de los desarrollos de sistemas de información y en general de cualquier clase de software, se puede clasificar en varios tipos, entre los que tenemos los siguientes:

- Cliente – Servidor
- Blackboard
- Modelo en N capas
- Intérprete
- Orientado a servicios

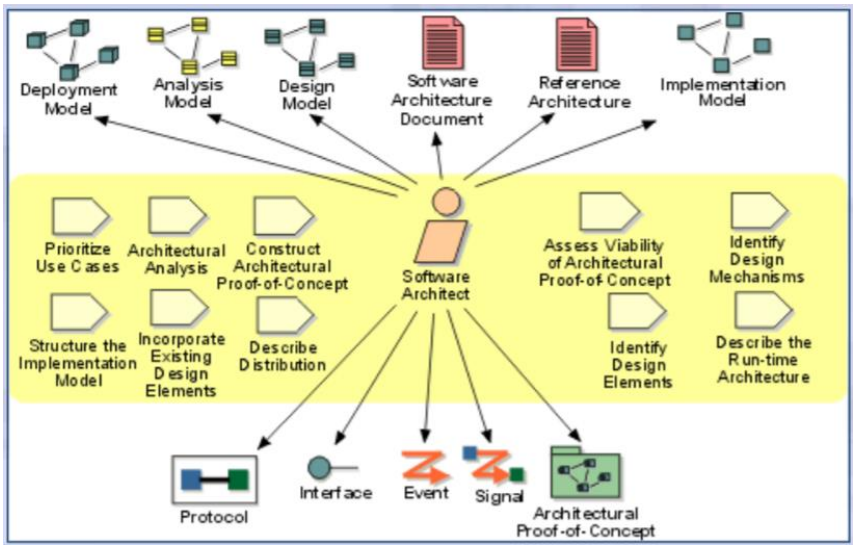


Imagen 9 Arquitectura de Software Fuente Internet

Para el desarrollo de cada vista se pueden utilizar herramientas como lo son Enterprise Architect o ArchiMate.

1.5.1.5 Arquitectura de Microservicios

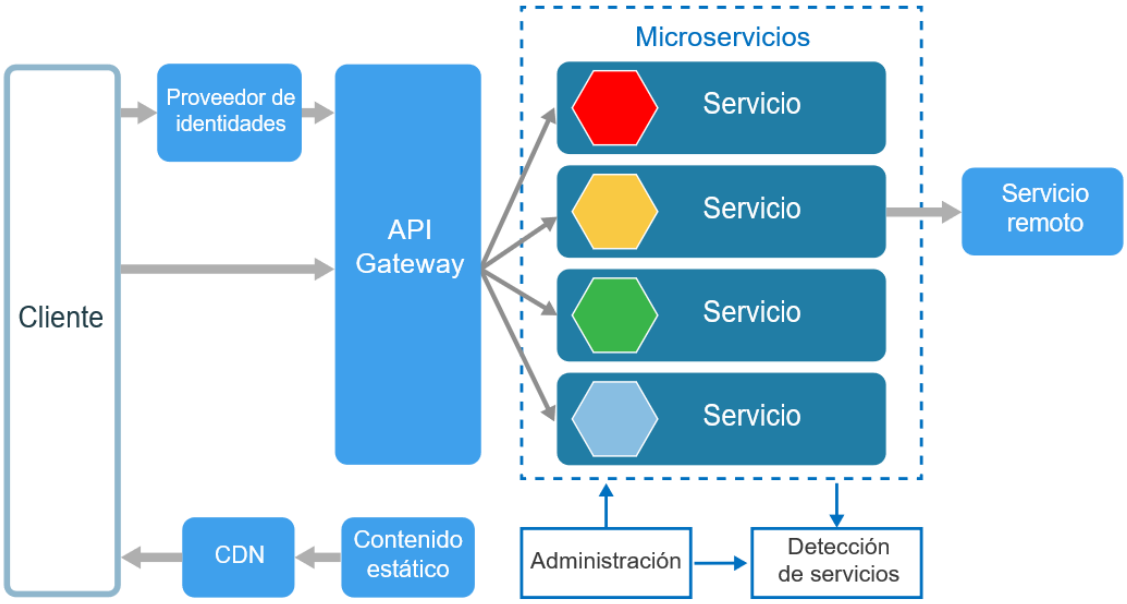


Imagen 10 Diagrama Arquitectura Microservicios. Fuente Microsoft

La arquitectura de Microservicios es una evolución de la arquitectura por servicios como lo es SOA, de hecho, es bastante similar a este. La idea de esta arquitectura es tratar de “descomponer aplicaciones en servicios más pequeños para escalamiento y manejo de ciclos de vida más eficientes”, como asegura la empresa AseSoftware en su blog [5].

Microsoft por otro lado define las siguientes características [6]:

- En una arquitectura de microservicios, los servicios son pequeños e independientes y están acoplados de forma flexible.
- Cada servicio es un código base independiente, que puede administrarse por un equipo de desarrollo pequeño.
- Los servicios pueden implementarse de manera independiente. Un equipo puede actualizar un servicio existente sin tener que volver a generar e implementar toda la aplicación.
- Los servicios son los responsables de conservar sus propios datos o estado externo. Esto difiere del modelo tradicional, donde una capa de datos independiente controla la persistencia de los datos.
- Los servicios se comunican entre sí mediante API bien definidas. Los detalles de la implementación interna de cada servicio se ocultan frente a otros servicios.
- No es necesario que los servicios compartan la misma pila de tecnología, las bibliotecas o los marcos de trabajo.

1.5.2 Marco Conceptual

1.5.2.1 Raspberry PI

Es un ordenador de bajo costo y de código abierto tanto a nivel de hardware (a excepción de su SoC principal) como a nivel de software del tamaño de una tarjeta de crédito que puede ser integrado con sensores para medición. Este dispositivo garantiza un bajo consumo energético, lo que garantiza su uso con recursos limitados. (Raspberry Pi foundation, 2015)

1.5.2.2 Comunicación Serial

La comunicación serial es un protocolo para la transmisión entre dos dispositivos que se incluye de manera estándar en todos los dispositivos computacionales. Además, permite adquirir datos de sensores remotos. [7]

1.5.2.3 Internet of Things (IoT)

Es un sistema de dispositivos para el hogar, estudio, personal o industria, interconectados a una red, que no necesariamente tienen que ser a internet y que tienen la particularidad de funcionar sin la interacción de las personas. Actualmente existen diversos proyectos a nivel de agricultura, monitoreo ambiental, producción de alimentos. [8]

1.5.2.4 Servicios Web

Es un conjunto de métodos o funciones expuestas a través de la red o internet y que pueden ser consumidas o ejecutadas por otras aplicaciones.

1.5.2.5 Oxígeno Disuelto

Es la medad de gas oxigeno que se encuentra disuelto en una solución acuosa, por lo que se hace importante la medición instantánea, sobre todo en los ambientes donde se alberga vida. Para el caso de los peces a una concentración más baja, se produce una mayor tensión en el ecosistema, lo que se traduce en la muerte de ellos. Se puede medir en parte por millón -ppm o en miligramos por litro -mg/L. [9]

1.5.2.6 pH

Es una unidad de medida de la alcalinidad o acidez en una solución, como lo puede ser el agua. Esta unidad realiza la medición de la cantidad de iones de hidrogeno que se encuentran en la solución [10]. La acidez en componentes como el agua es clave para la supervivencia de algunos organismos.

1.5.2.7 Tilapia

Pez de origen africano, de gran valor comercial en algunos países y rápido crecimiento. Su cultivo se realiza sobre zonas tropicales. Las escamas de este pez tienen un alto contenido en colágeno, lo que lo hace atractivo para el comercio cosmético. [11]

1.5.3 Marco Espacial

El proyecto se realizará tomando como referencia la problemática que se presenta en el proceso de medición de la calidad del agua en los tanques de

cultivos de tilapias de la finca *Allegro* ubicada en el municipio de San Francisco del departamento de Cundinamarca – Colombia.

1.6 ASPECTOS METODOLÓGICOS

1.6.1 Tipo de estudio

El estudio de este fenómeno es de orden exploratorio, teniendo en cuenta que esta investigación está analizando y proponiendo una solución tecnológica innovadora a un problema directo en la producción de tilapias en la finca Allegro, que, aunque está identificado, no hay registros de propuestas tecnológicas concretas que permitan automatizar de alguna manera el monitoreo continuo de la calidad del agua en los estanques donde se produce la tilapia, así como la toma de acciones en tiempo real para dar solución a las alertas que se presenten.

1.6.2 Método de Investigación

A través del método de investigación inductivo y partiendo del problema que se presenta actualmente en la producción de tilapias, se pretende analizar el proceso de monitoreo de la calidad del agua en la finca Allegro, con el fin de obtener información relevante que sirva como insumo para el desarrollo de un prototipo de sistema web como solución a esta problemática y a su vez como antecedente en futuras investigaciones relacionadas con el tema.

1.6.3 Fuentes y técnicas para la recolección de la información

Los insumos informativos para la investigación se obtendrán mediante, fuentes primarias como observaciones en sitio, entrevistas directas a los

encargados del monitoreo de la calidad del agua en la finca Allegro y como fuentes secundarias, documentación técnica referente a desarrollo y arquitectura de software, comunicación entre sistemas y producción de cultivos de tilapias.

1.6.4 Tratamiento de la información

La información que se obtenga a parir de fuentes primarias y secundarias será utilizada para contextualizar el fenómeno y proponer el flujo teórico y práctico para el desarrollo del sistema web. Esta información se ubicará en el marco referencial y en los anexos de la investigación.

Para el caso de la información que se recolecte por medio de los sensores se almacenará en la base de datos propia del sistema web desarrollado, con el fin de realizar un análisis del comportamiento de la calidad del agua y presentar al usuario una interfaz de monitoreo continuo de las mediciones de calidad del agua.

1.7 ALCANCES, LIMITACIONES Y RESULTADOS ESPERADOS

1.7.1 Alcances

- Integración con los sensores de calidad del agua de los tanques.
- Implementación de un sistema web para el almacenamiento y visualización de las mediciones de calidad del agua.
- Notificación de alertas por lecturas que se encuentren fuera de los rangos deseados de calidad del agua.

1.7.2 Limitaciones

- La integración de los sensores se realizará a través de comunicación serial.
- Las variables de calidad del agua serán Temperatura, Oxigeno, Saturación y pH.
- El sistema web se diseñará y desarrollará teniendo en cuenta el contexto del proyecto de cultivo de tilapias en la finca *Allegro* en el municipio San Francisco del departamento de Cundinamarca Colombia.
- Las notificaciones se realizarán vía email.
- Se requiere acceso a internet desde la finca *Allegro*.

1.7.3 Resultados esperados

El resultado global que se espera obtener al implementar el sistema web es la automatización del proceso de monitoreo de la calidad del agua en los tanques de cultivo de tilapias incluyendo las alertas a encargados por eventualidades en el proceso de medición.

También se espera que el usuario pueda consultar los históricos de la calidad del agua registrados por el sistema con el fin de realizar un análisis que le permita estudiar el comportamiento y las variaciones de las variables en su proyecto de cultivo.

PARTE II. DESARROLLO DE LA INVESTIGACIÓN

ARQUITECTURA EMPRESARIAL

Con el fin de realizar un mejor entendimiento de la organización – Finca Allegro – se procede a realizar un proceso de arquitectura empresarial, esto, ya que se observa que es importante conocer cuál es la motivación de ellos, y por qué se hace necesario la implementación de este prototipo.

2.1 FUNDACIÓN FINCA ALLEGRO

2.1.1 Misión

Fomentar el desarrollo rural a través de producción orgánica de vegetales, crianza avícola y acuicultura, como parte de la educación que reciben los niños del campo.

2.1.2 Visión

Ser un ejemplo de crecimiento rural en Colombia fomentando la producción de productos orgánicos, siempre orientando la importancia de cuidado del campo en el desarrollo de toda la niñez.

2.1.3 Objetivos Organizacionales

- Controlar el proceso de levante, engorde y crianza de Tilapia roja.
- Mantener los estándares más altos de calidad en el proceso de cultivo.

2.1.4 Organigrama

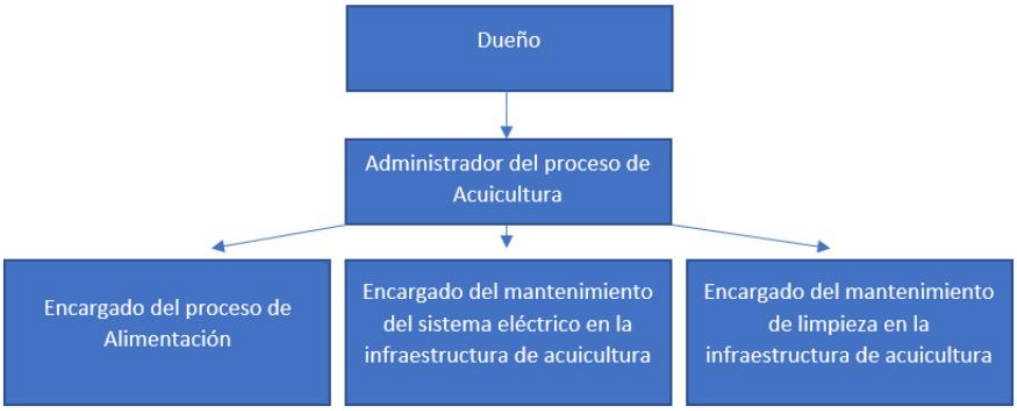


Imagen 11 Organigrama Fundación Finca Allegro – Fuente: Fundación Finca Allegro

2.1.5 Procesos del negocio

Los procesos relacionados con el cultivo de tilapias en la finca Allegro San Francisco, están catalogados como procesos operativos de gestión y producción.

Entre los procesos de gestión encontramos la administración de los recursos físicos y tecnológicos para mantener la operación. Este proceso se encarga de comprar, organizar y proveer las fuentes alimenticias, de personal y eléctricas, necesarias para producir el cultivo de tilapias.

Los procesos operativos del proyecto de cultivo de tilapia son alimentación y medición de la calidad del agua. En donde, el proceso de alimentación se realiza 3 veces por día y dependiendo de la edad del pez se le suministra la cantidad de alimento necesaria para mantener el crecimiento constante del animal. Y, por otro lado, el proceso de medición de la calidad del agua se realiza 4 veces por día, el

operario se encarga de obtener la información directamente de los sensores que poseen los tanques, dependiendo de los resultados el operario determina si es necesario estabilizar o adicionar ciertos elementos para mantener una calidad adecuada para el cultivo.

2.1.6 Productos y Servicios

Venta al por mayor de Tilapia Roja.

2.2 Archimate

El lenguaje ArchiMate, complementa a TOGAF brindando independencia de vendedor, y se centra en los conceptos, incluyendo representaciones graficas que ayudan a crear consistencia e integración a través de las diferentes figuras y vistas de TOGAF.

La estructura del núcleo del lenguaje ArchiMate corresponde estrechamente con las tres Arquitecturas de la TOGAF ADM.

Algunas vistas de TOGAF no concuerdan en el núcleo de ArchiMate, y esto se da debido a que TOGAF es más amplio y se ocupa tanto de estrategias de alto nivel como de aspectos del desarrollo de sistemas, mientras el núcleo de ArchiMate es limitado nivel de abstracción de la arquitectura empresarial. [12]

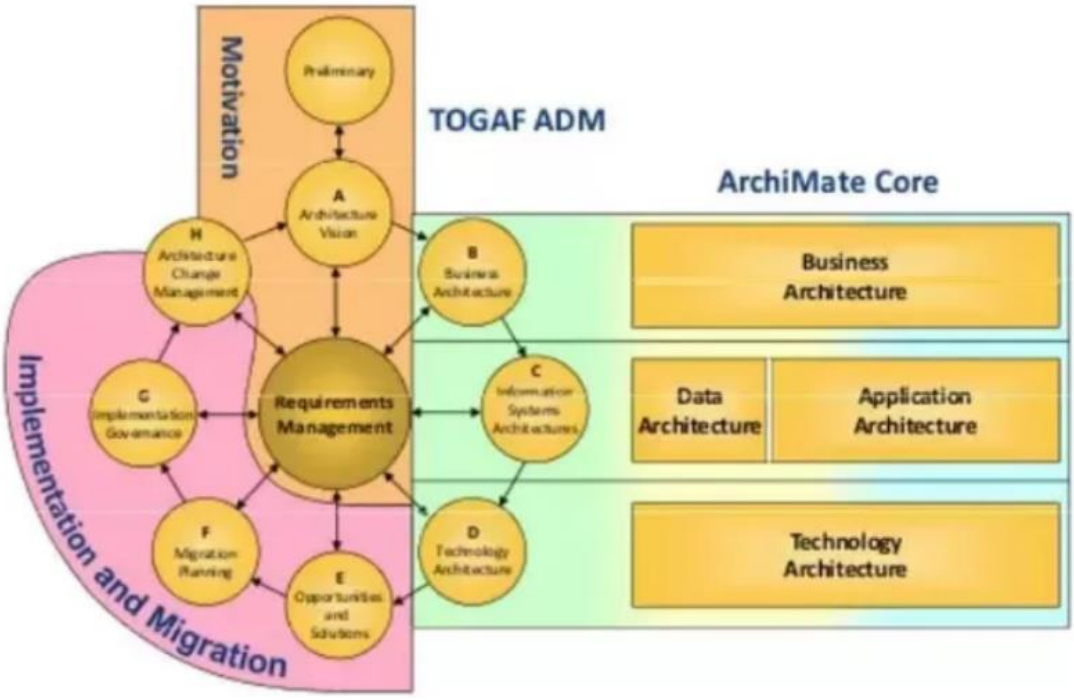


Imagen 12 Correlación entre lenguaje ArchiMate y TOGAF ADM [12]

2.3 CAPA DE NEGOCIO

Los puntos de vista de ArchiMate son diagramas de modelos compactos con un enfoque que cambia dependiendo del involucrado y del proceso, en el caso del negocio o la esencia de la entidad que tienen la necesidad.

Estos puntos de vista permiten mediante de diagramas, representar los procesos internos del negocio para aclarar el flujo y los componentes que juntos forman los pilares del negocio.

Los puntos de vista de a través de Archimate sobre la capa de negocio buscan representar y contemplar los factores internos y externos de la organización.

2.3.1 Punto de vista de organización

En el modelo de organización del caso de estudio se presenta las interacciones de los roles dentro de la finca Allegro en San Francisco Cundinamarca. Entre los roles que interactúan encontramos al administrador y los operarios.

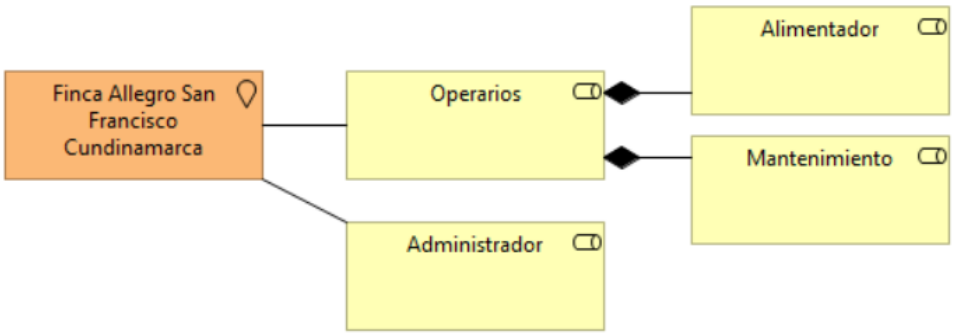


Imagen 13 Punto de vista de organización - Fuente: Los autores

2.3.2 Punto de Vista de Cooperación de Actor

En el caso de estudio relacionado a el cultivo de tilapias y el poder realizar una automatización al proceso de monitoreo y control de la calidad del agua, el punto de vista de cooperación de actor, enfocado a los operarios y administradores, presenta una relación de como los actores contarán con el servicio de monitoreo a través de una plataforma web donde podrán administrar usuarios, parámetros de tanques y visualizar reportes y monitoreo en tiempo real.

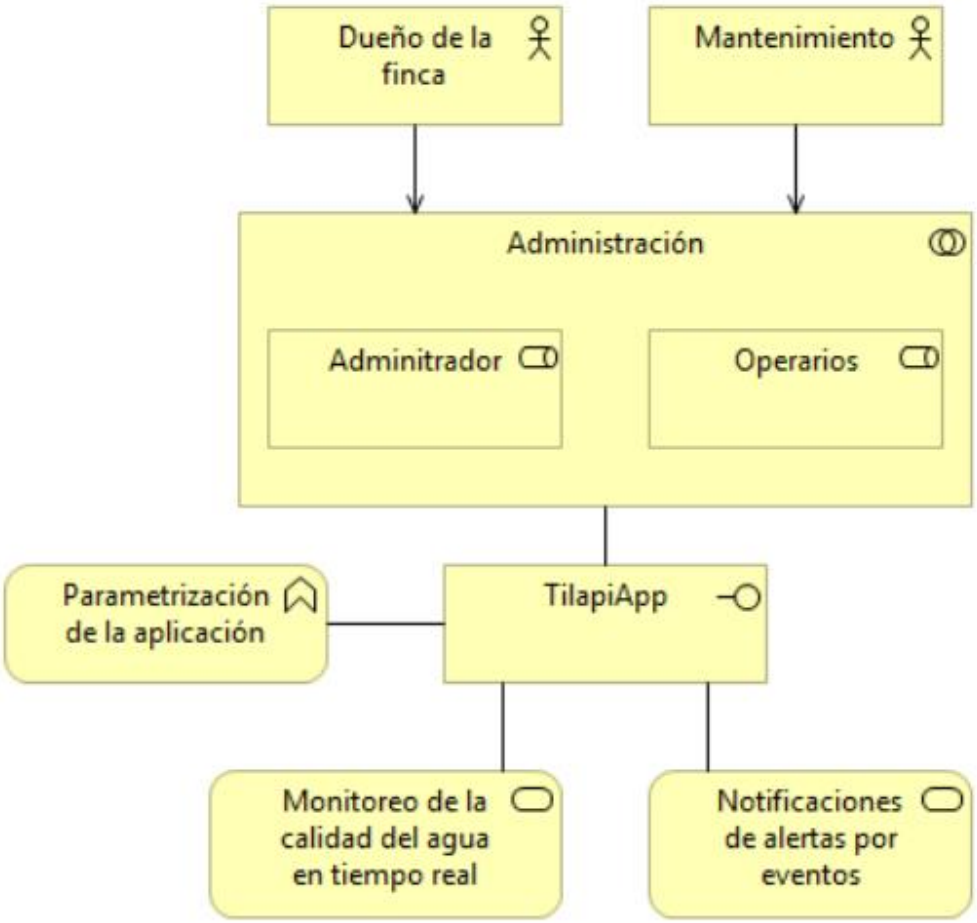


Imagen 14 Punto de vista de cooperación de actor - Fuente: Los autores

2.3.3 Punto de Vista de Función Negocio

Este modelo presenta cada función que está a cargo de cada Rol, dentro del sistema se encuentran dos tipos de perfil asociados a cada Rol (Administrador y Operarios), cada uno posee funciones diferentes y está a cargo de diferentes procesos.

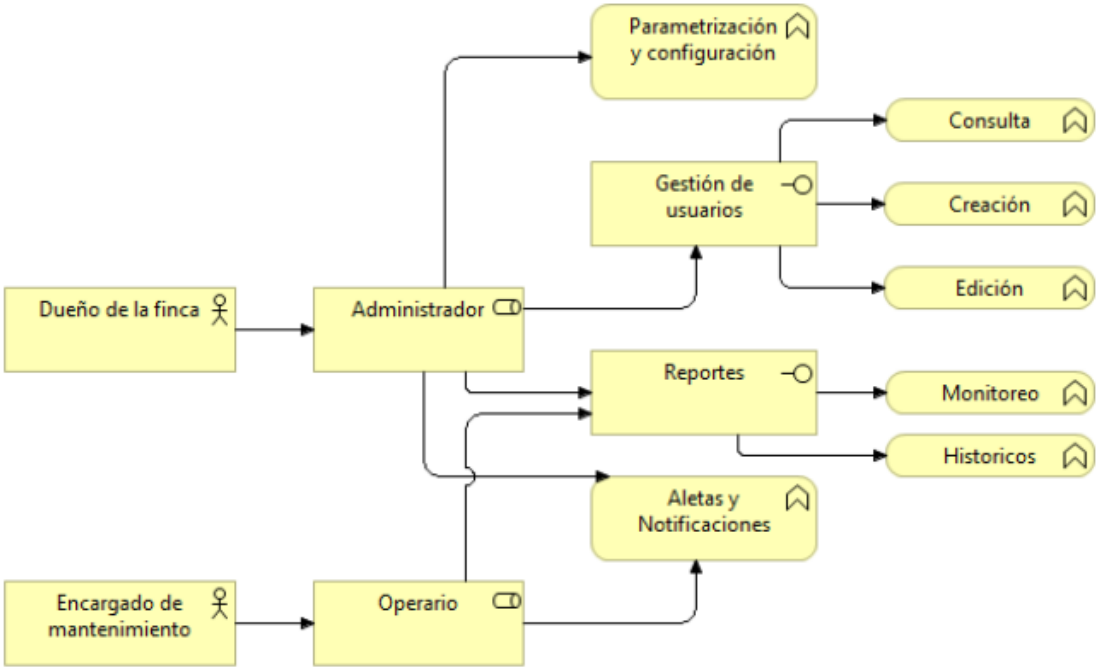


Imagen 15 Punto de Vista de Función Negocio – Fuente: Los Autores

2.3.4 Punto de Vista de Proceso

Este modelo presenta la función principal del monitoreo constante de la calidad del agua utilizando el sistema TilapiApp.

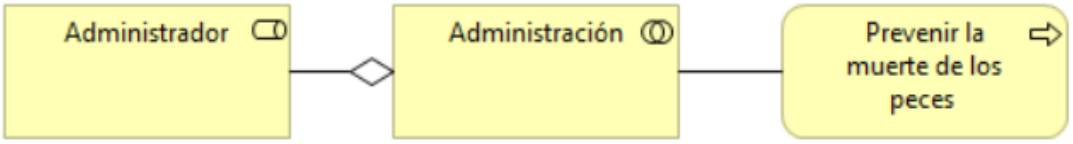


Imagen 16 Punto de Vista de Proceso – Fuente: Los Autores

2.3.5 Punto de Vista de Cooperación de Proceso

El siguiente modelo presenta la función principal de la administración de la finca Allegro la cual es prevenir las muertes de los peces y como el sistema puede generar alertas que generen acciones correctivas.

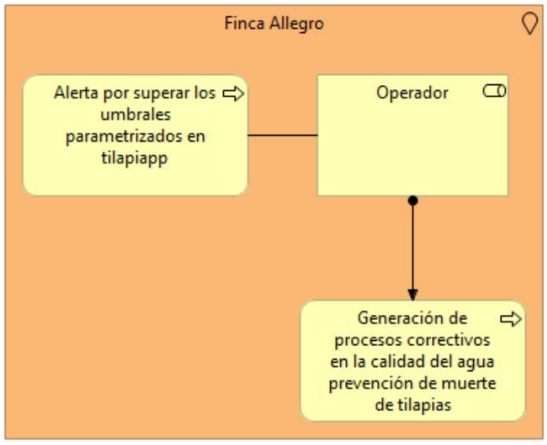


Imagen 17 Punto de Vista de Cooperación de Proceso – Fuente: Los Autores

2.3.6 Punto de Vista de Producto

El modelo de producto representa la funcionalidad general del prototipo de aplicación web Tilapiapp como servicio y su finalidad de monitoreo constante y prevención de muerte de tilapias con alertas a los usuarios.

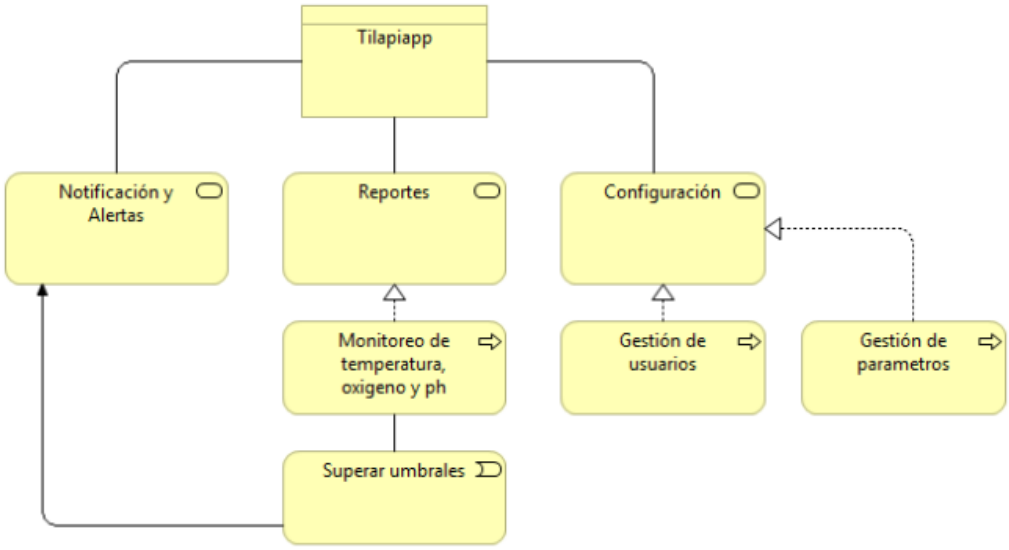


Imagen 18 Punto de Vista de Producto – Fuente: Los Autores

2.4 CAPA DE APLICACIÓN

Este punto de vista contiene las aplicaciones, componentes y sus relaciones entre sí, así como también detallar la estructura que tiene o tendrá la aplicación que soportará la operación del negocio.

Permite tener un panorama sobre interfaces, servicios, procesos que pueden ser reutilizados a lo largo del desarrollo de esta, con el fin de hacer sencilla y simple la aplicación. Es decir, describe la estructura y la interacción de la aplicación.

2.4.1 Punto de Vista de Comportamiento

En este punto de vista identificamos como es la interacción de la aplicación Tilapiapp y cuáles son los componentes principales, en donde se soportará la operación del negocio.

Estos componentes son:

- Administración Usuarios: Gestiona los usuarios del sistema.
- Gestión Alertas: Encargado del envío de alertas por medio de correo.
- Servicio REST: Servicio para comunicación entre base de datos, sensor y aplicación Web.
- Base de Datos: Repositorio de datos de la aplicación Web.
- Reportes: Administración de la información.
- Sensores: Gestión de los sensores ubicados en los estanques.

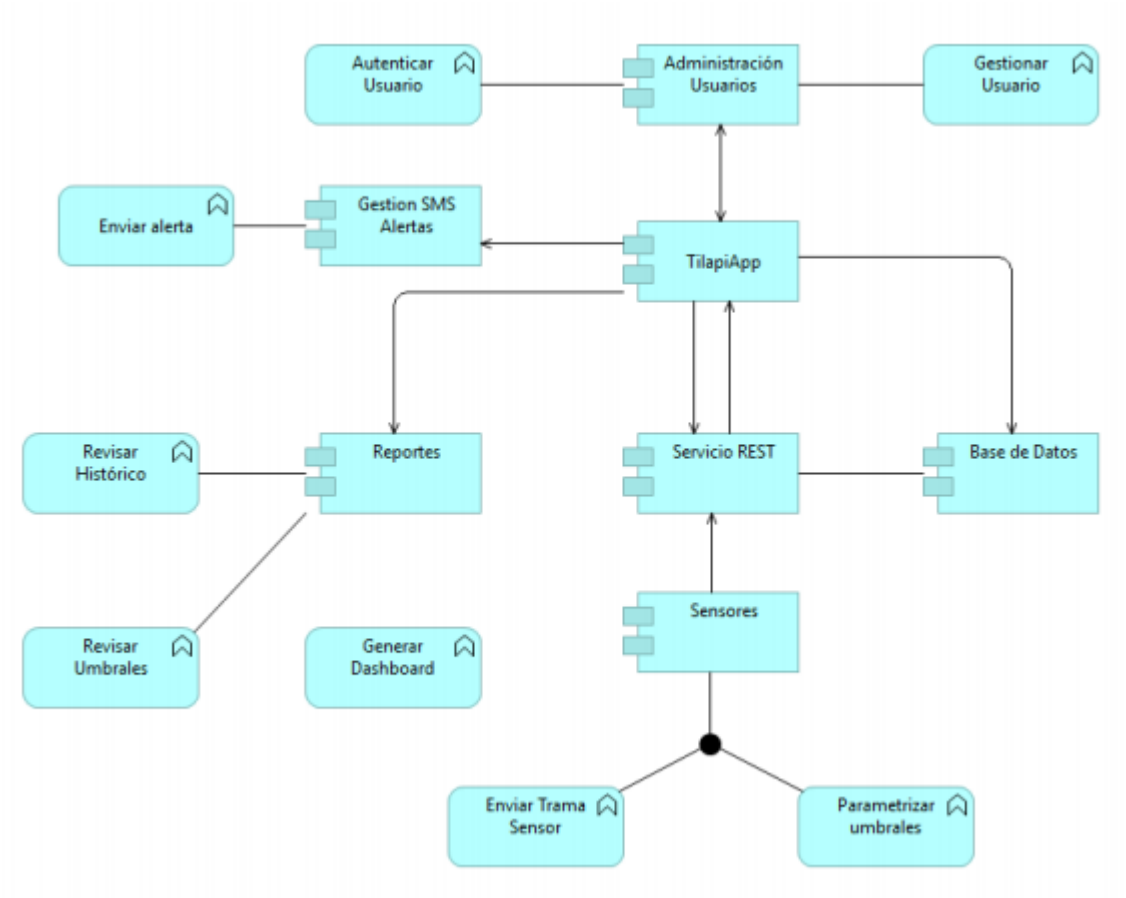


Imagen 19 Punto de Vista de Comportamiento – Fuente: Los Autores

2.4.2 Punto de Vista de Cooperación de Aplicación

Para este punto de vista, se identificaron cuatro ubicaciones lógicas en la aplicación TilapiApp, de forma que haya una alta cohesión de acuerdo a sus funciones.

Se tiene la capa Fron-End en donde el usuario tiene acceso a los componentes de Administración de Usuarios, Alertas Sensores y Gestión de Parámetros, la capa Back-End en donde se tiene Base de Datos, el Servicio REST y los Sensores, la capa externa encargada del manejo de envío de correos de alertas y por último la capa Raspberry que contiene la Base de Datos del sensor.

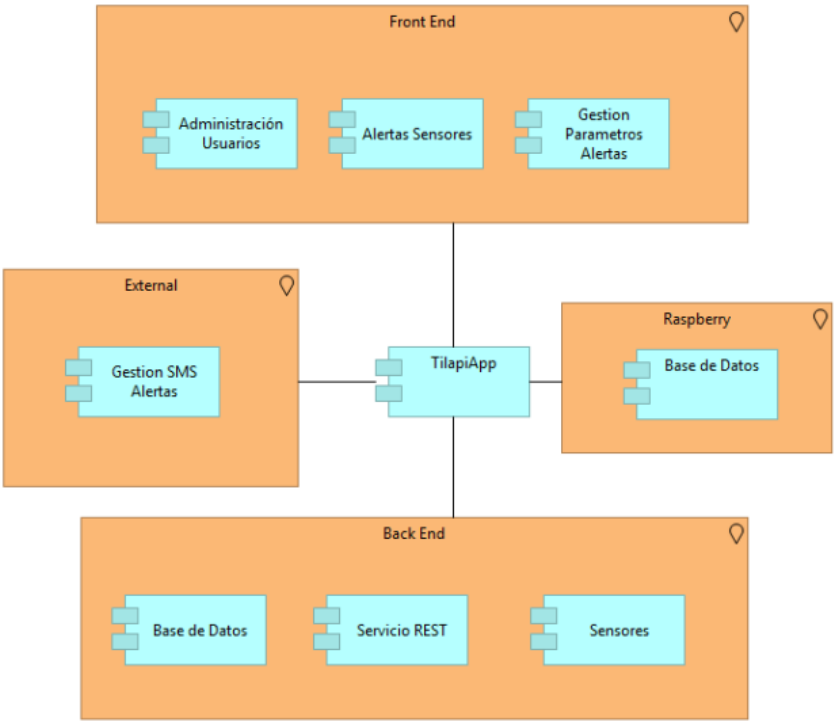


Imagen 20 Punto de Vista de Cooperación de Aplicación

2.4.3 Punto de Vista de Estructura de Aplicación

Este punto de vista muestra la arquitectura lógica desde un alto nivel. En donde, se pueden observar cuatro interfaces principales, encargadas de la comunicación con los componentes. Asimismo, se tienen los componentes que implementan cada una de ellas para el funcionamiento del modelo0020y de la aplicación.

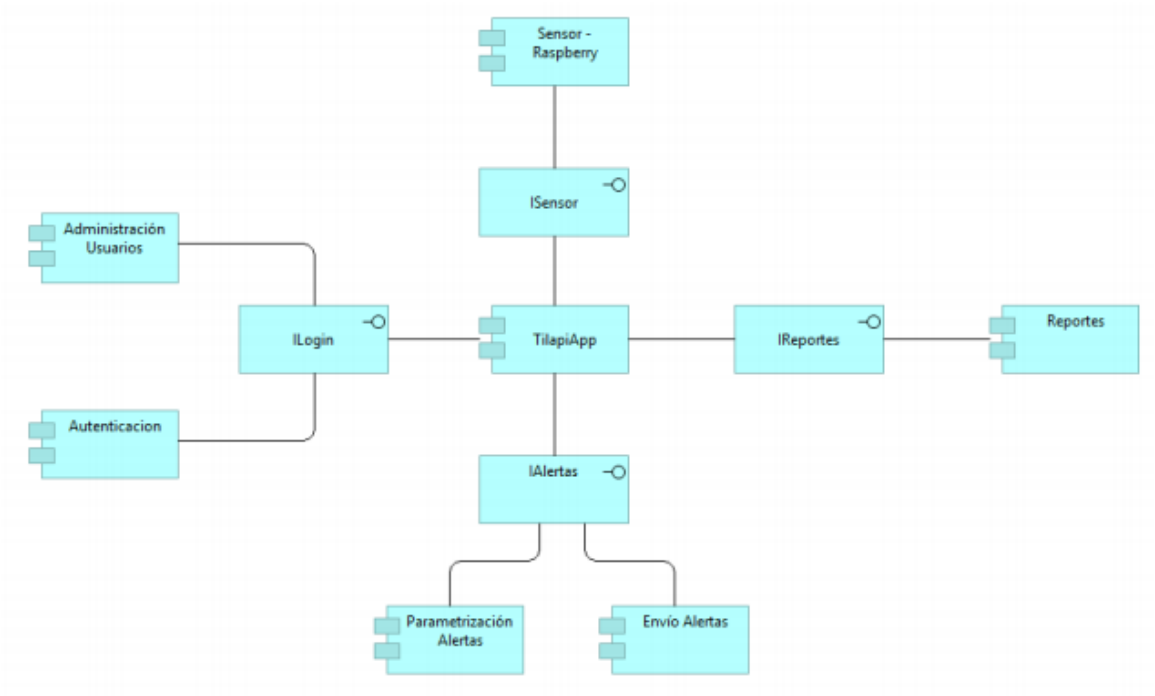


Imagen 21 Punto de Vista de Estructura de Aplicación

2.4.4 Punto de Vista de Uso de Aplicación

Para este punto de vista se identificaron los cuatro procesos de negocio fundamentales, captura de información de sensores, envío de alertas, consulta de información y administración de usuarios, todos ellos asociados a sus respectivos servicios y componentes de aplicación.

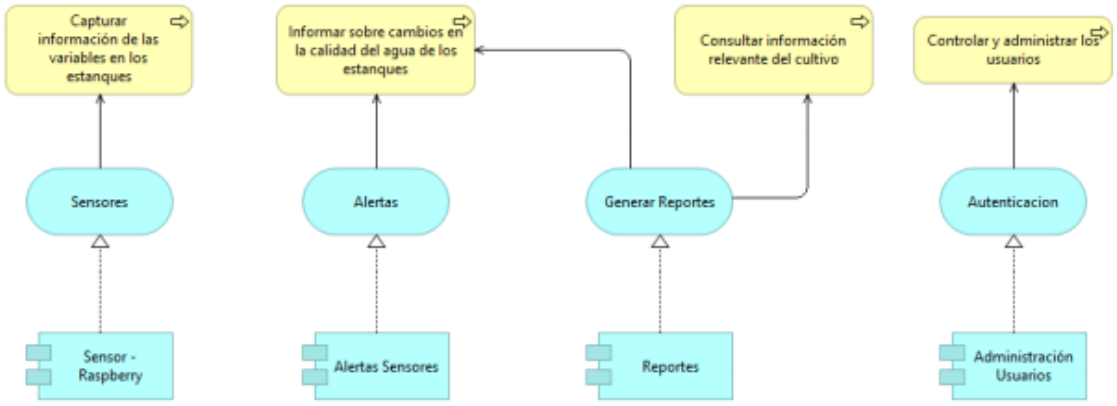


Imagen 22 Punto de Vista de Uso de Aplicación – Fuente: Los Autores

2.5 CAPA DE TECNOLOGÍA

Esta capa muestra cual es la relación entre software y hardware que es necesario para el despliegue y posterior uso de la aplicación

2.5.1 Punto de Vista de Infraestructura

En el siguiente modelo se presenta la infraestructura que tendrá el sistema Tilapiapp desde la descripción de hardware que leer ´ a los sensores hasta el hospedaje de la aplicación web donde se parametrizará y presentarán los resultados.

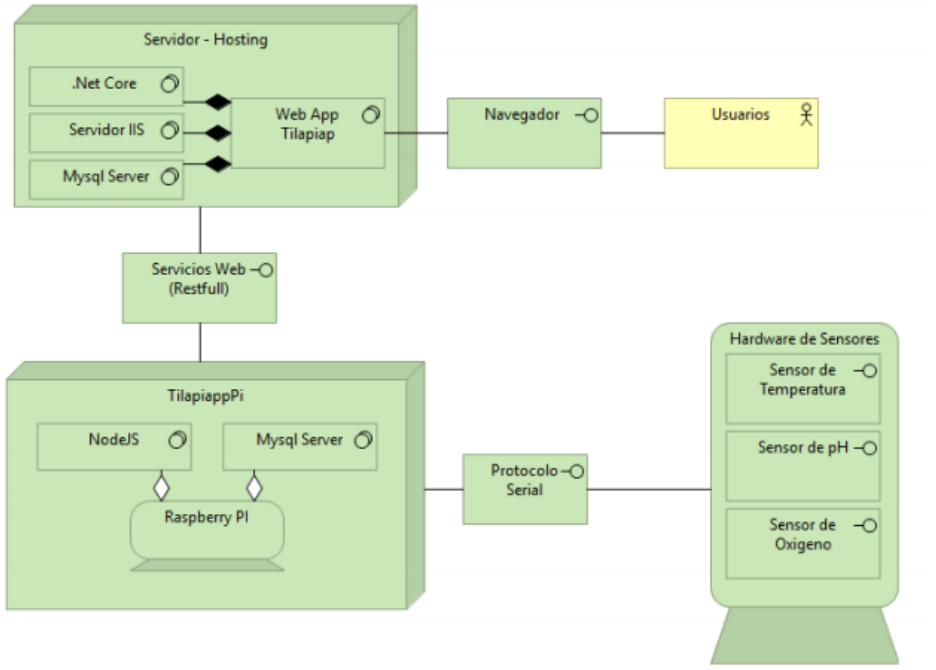


Imagen 23 Punto de Vista de Infraestructura – Fuente: Los Autores

2.5.2 Punto de Vista de Uso de Infraestructura

En el siguiente modelo se presenta como se agrupan las funcionalidades dentro de Tilapiapp y el servidor donde se alojará la aplicación, una interface de comunicación a través de peticiones HTTP entre el sistema de sensores y Tilapiapp.

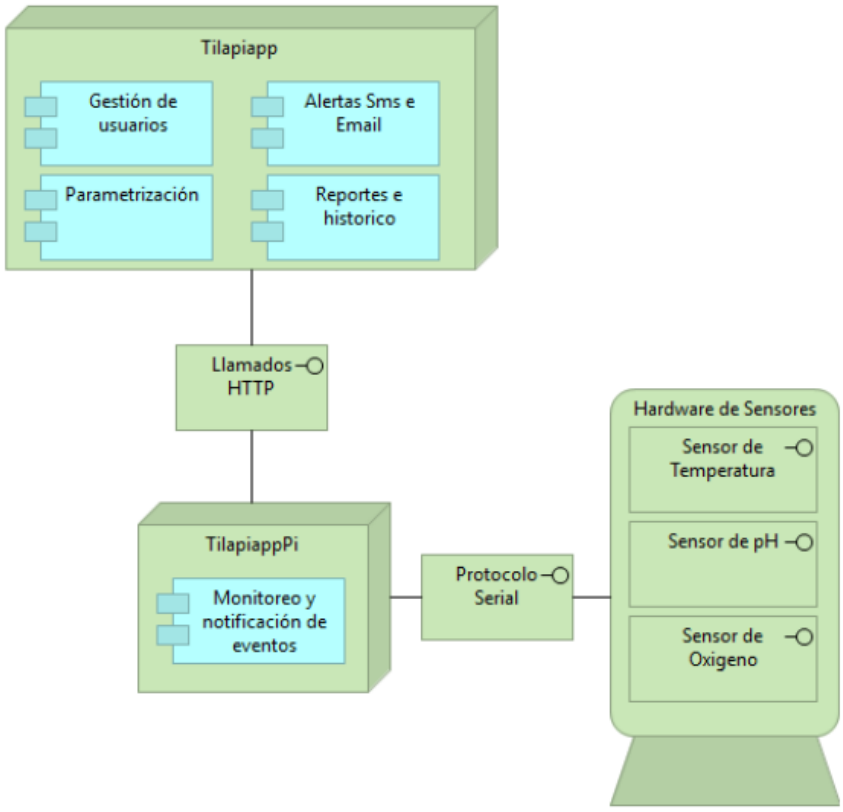


Imagen 24 Punto de Vista de Uso de Infraestructura – Fuente: Los Autores

2.5.3 Punto de Vista de Implementación

En el modelo a continuación se presenta los tres entornos donde el sistema completo se implementará. (Servidor Web, Base de datos, Red Local y Hardware)

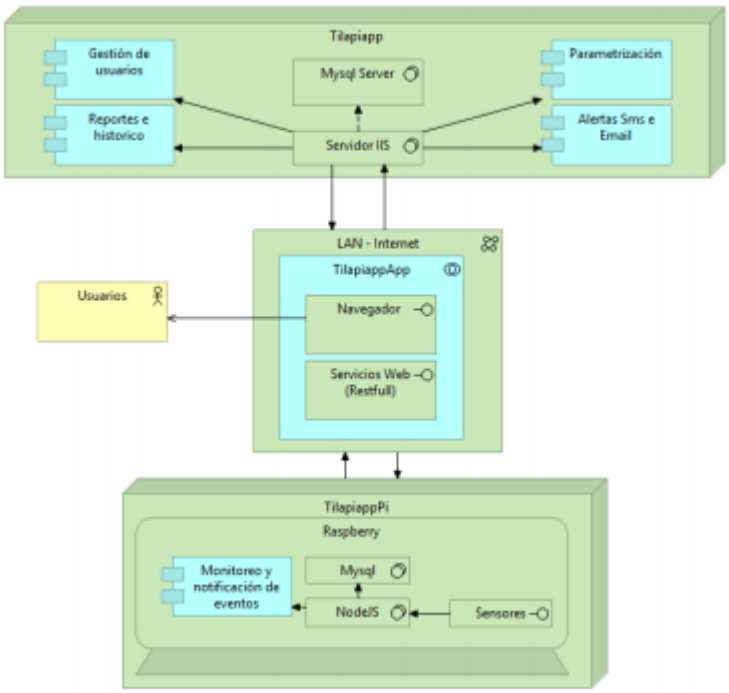


Imagen 25 Punto de Vista de Implementación – Fuente: Los Autores

2.5.4 Punto de Vista de Estructura de la Información

En este modelo se presenta la finalidad principal del sistema la cual es el monitoreo de la calidad del agua, también las funciones propias del aplicativo presentadas con la capa de negocio y las entidades que contendrán la información.

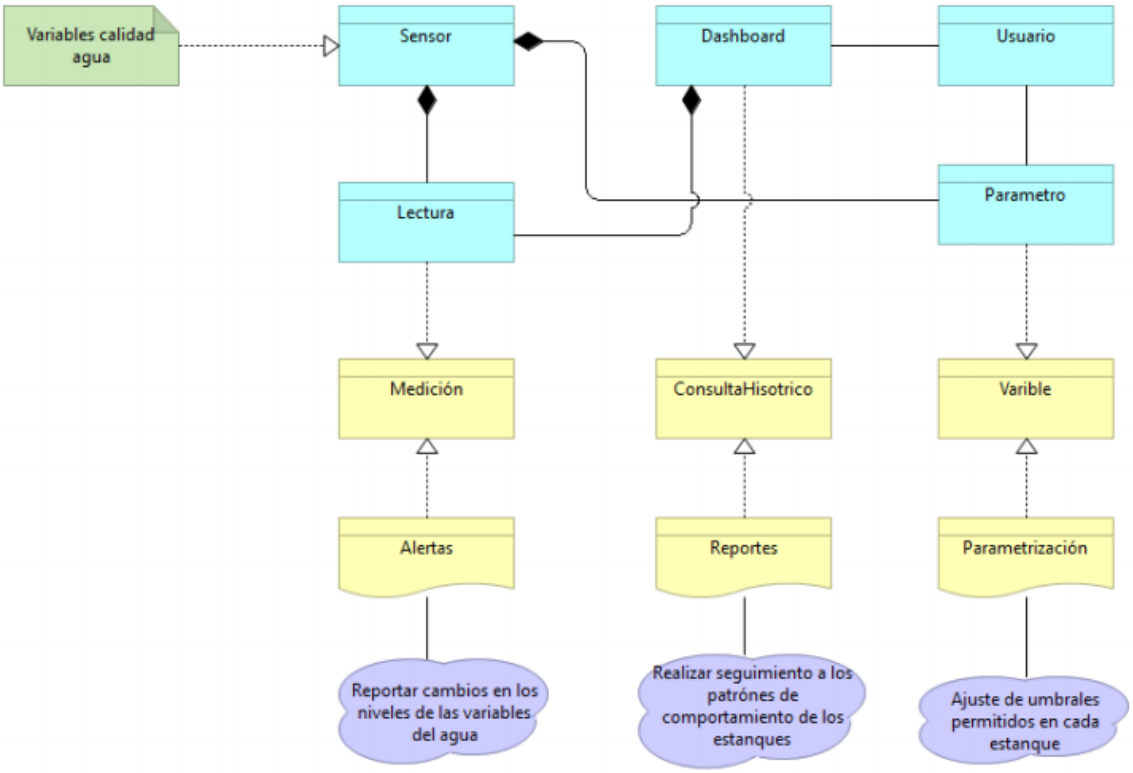


Imagen 26 Punto de Vista de Estructura de la Información – Fuente: Los Autores

2.5.5 Punto de Vista de Realización del Servicio

Se presenta el modelo con respecto a una relación entre el negocio y la base de la aplicación donde se presentan las funcionalidades generales.

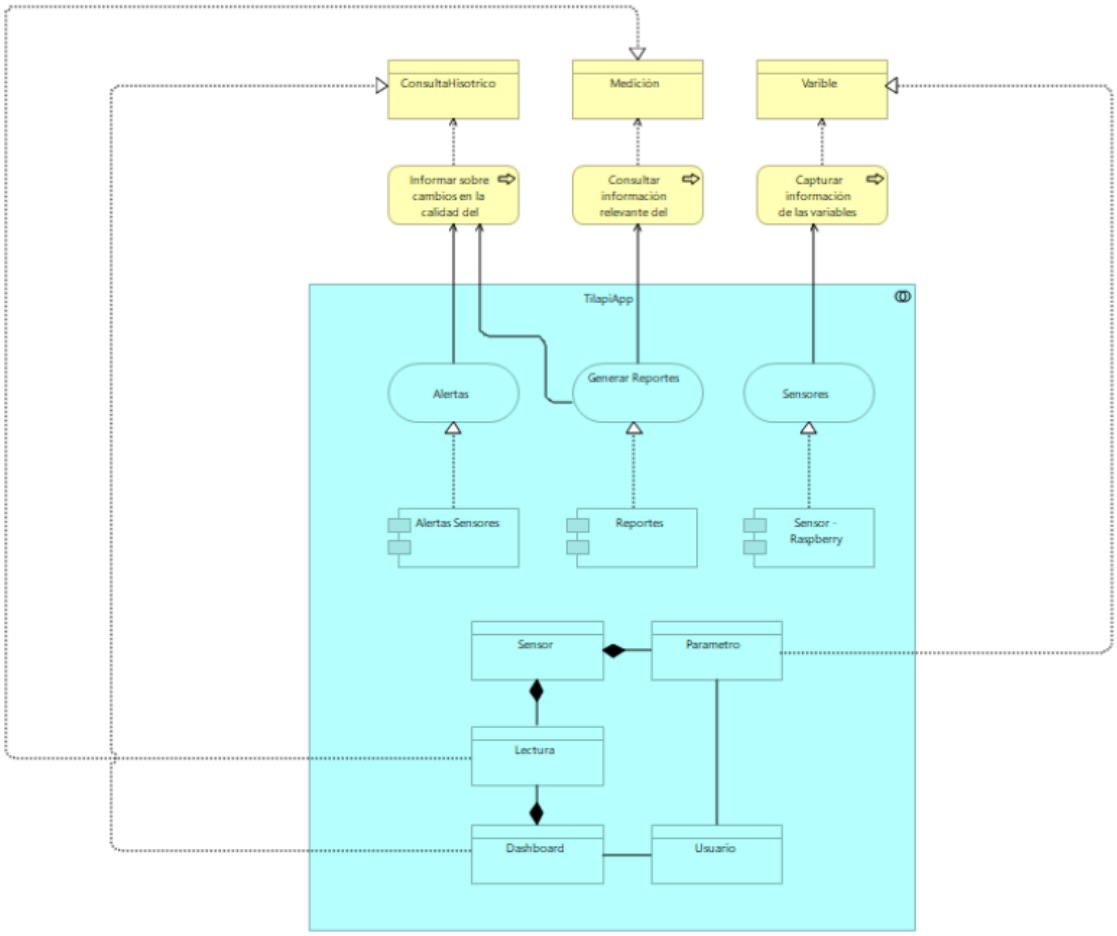


Imagen 27 Punto de Vista de Realización del Servicio – Fuente: Los Autores

2.5.6 Punto de Vista de Capas

En este modelo vemos una vista general de todas las capas de negocio, aplicación e infraestructura que conforman en su conjunto la arquitectura de ArchiMate hasta este punto, se destaca la manera como estas capas se relacionan y como las capas inferiores van dando soporte a las capas superiores.

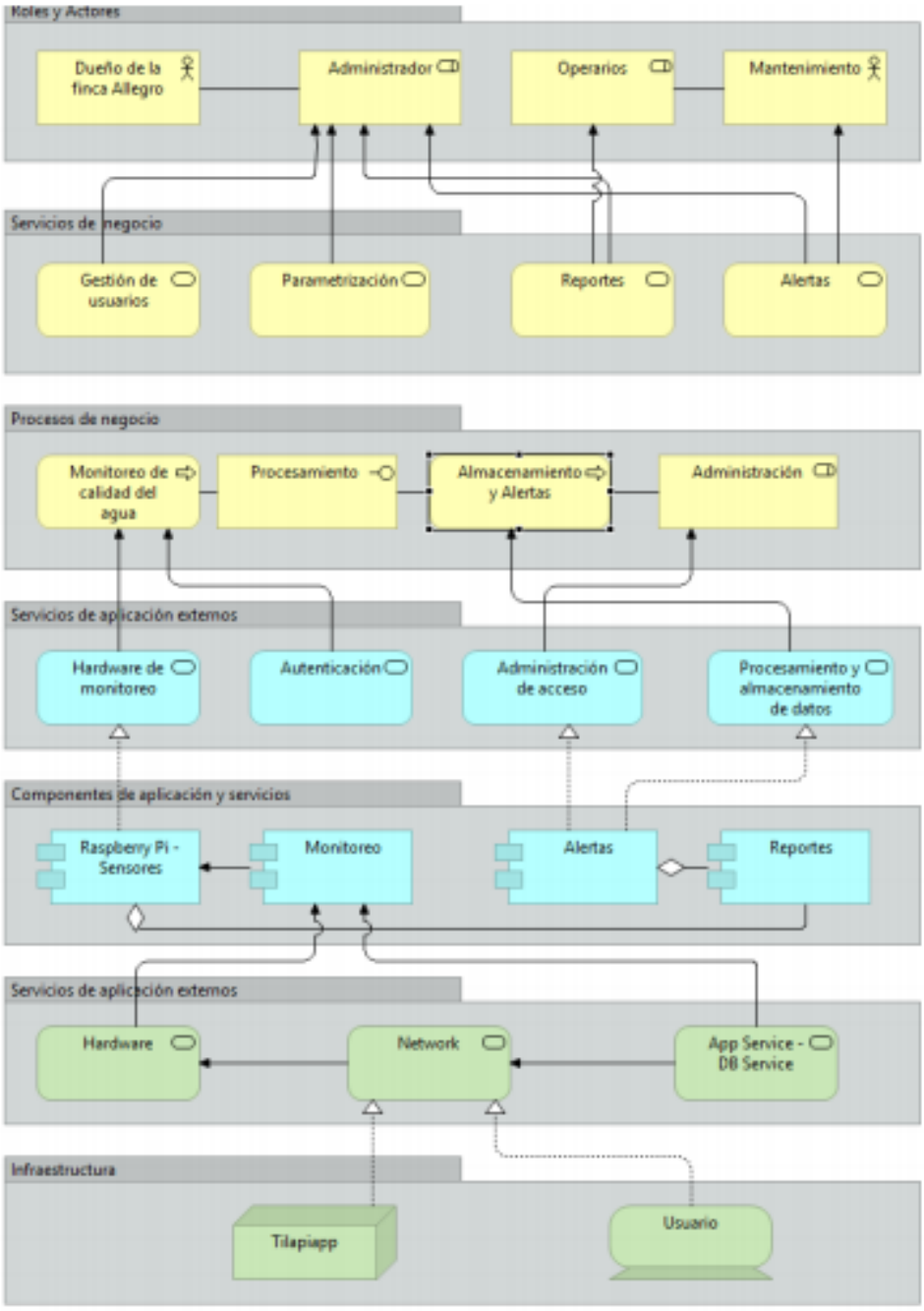


Imagen 28 Punto de Vista de Capas – Fuente: Los Autores

2.6 CAPA DE MOTIVACIÓN

Esta capa permite identificar cuáles son aquellas razones que llevan o desencadenan la necesidad de cambio en la organización. Es importante tener presente que las motivaciones que se den juegan un papel crucial pues son las que influyen, orientan y limitan el diseño del modelo que se desea crear o que se propone para suplir dicha necesidad.

Las motivaciones que se presenten pueden ser generadas tanto por factores interno como externos. Asimismo, aquí se puede tener claridad en cuáles son las partes interesadas en el proyecto o conocidos como *Stakeholders*.

2.6.1 Punto de Vista de Stakeholder

En este punto de vista se observan dos metas motivacionales que son la monitorización de las variables de condiciones del agua y alertar sobre posibles cambios que puedan afectar la vida de los peces. De aquí se identifica que los interesados en que se lleve a cabo estas tareas son el dueño de la finca y los operarios que trabajan en ella.

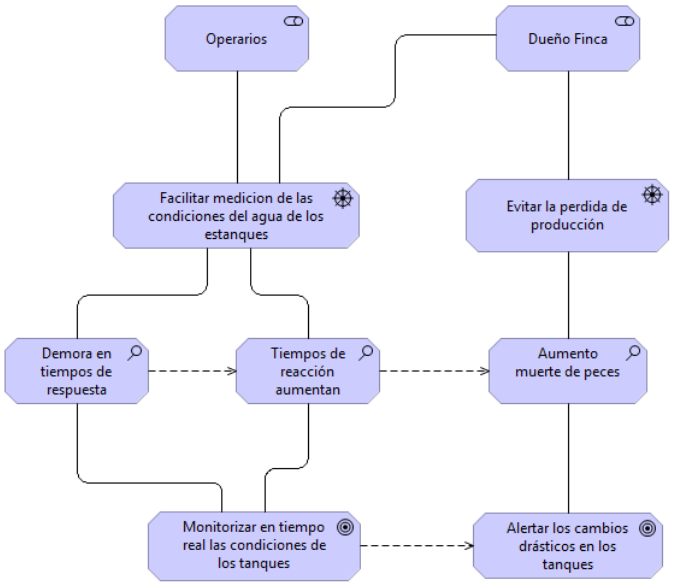


Imagen 29 Punto de Vista de Stakeholder – Fuente: Los Autores

2.6.2 Punto de Vista de Realización del servicio

En este modelo nos centramos en las metas motivacionales, medición de variables y alertas de cambios en condiciones, partiendo del hecho que se tiene un principio común a las dos que es brindar herramientas que permitan efectuar o llegar a realizar las metas antes mencionadas.

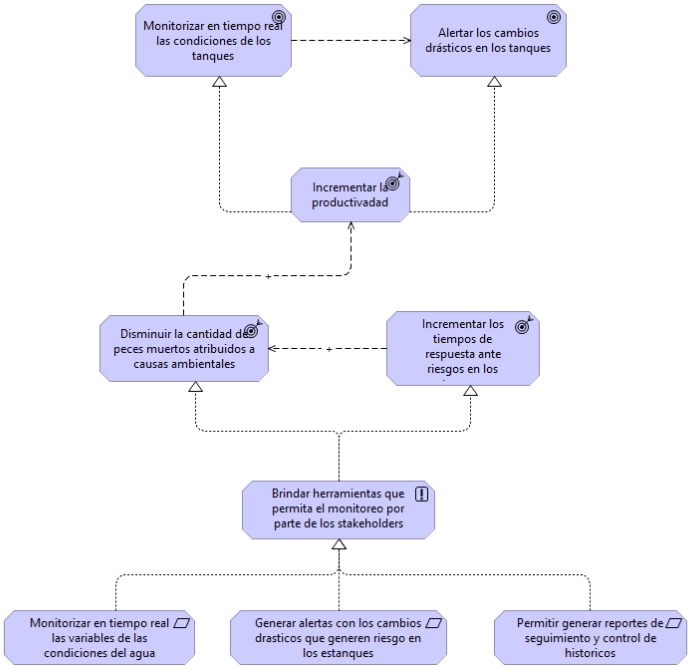


Imagen 30 Punto de Vista de Realización del servicio – Fuente: Los autores

2.6.3 Punto de Vista de Contribución de Objetivos

En este modelo se observa cómo los objetivos o metas motivacionales pueden ser alcanzados haciendo uso de tres actividades puntuales, que son generando alertas, monitorizando en tiempo real y generando protocolos de atención inmediata.

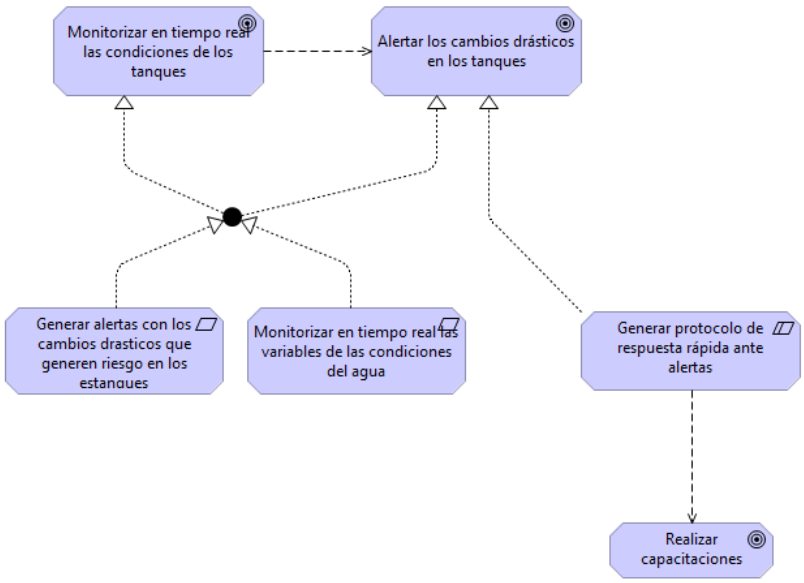


Imagen 31 Punto de Vista de Contribución de Objetivos

2.6.4 Punto de Vista de Principios

En este modelo se tienen definidos los principios que permitirán llegar a cumplir las metas motivacionales propuestas para el proyecto. Estos principios son tres, oportunidad, control y confianza.

En donde,

- Oportunidad: En cuanto a tener los datos inmediatos y en tiempo real
- Control: Parametrización de umbrales máximos y mínimos, reportes de gestión y acceso a estados de los tanques.
- Confianza: Se genera al tener la información fiable y acorde a la necesidad de los *Stakeholders*.

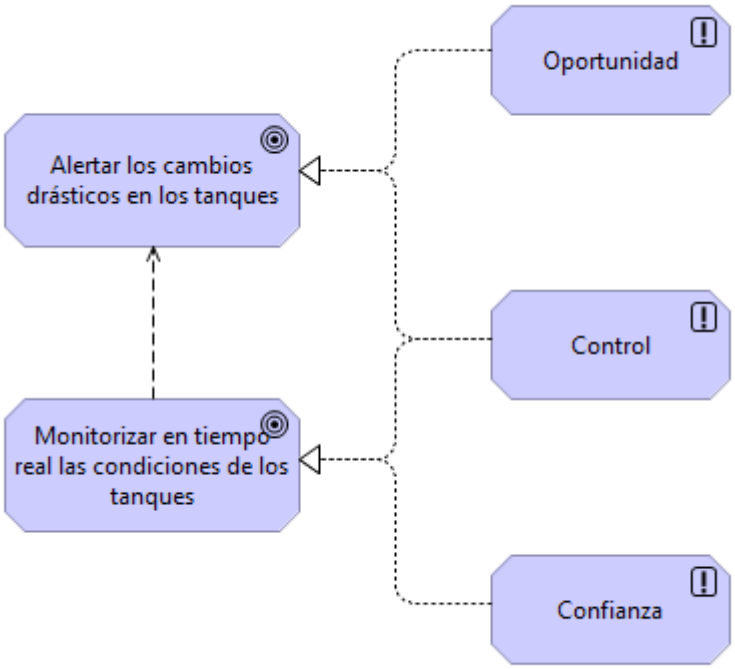


Imagen 32 Punto de Vista de Principios – Fuente: Los Autores

2.6.5 Punto de Vista de Realización de Requerimientos

En este punto de vista, se detallan cuales son los requerimientos que deben cumplirse para llegar a cumplir con las metas motivacionales propuestas anteriormente. Asimismo, se observa que para llegar a cumplir los objetivos se depende tanto del Sensor como de Tilapiapp, pues de la implementación de estos depende el éxito del proyecto.

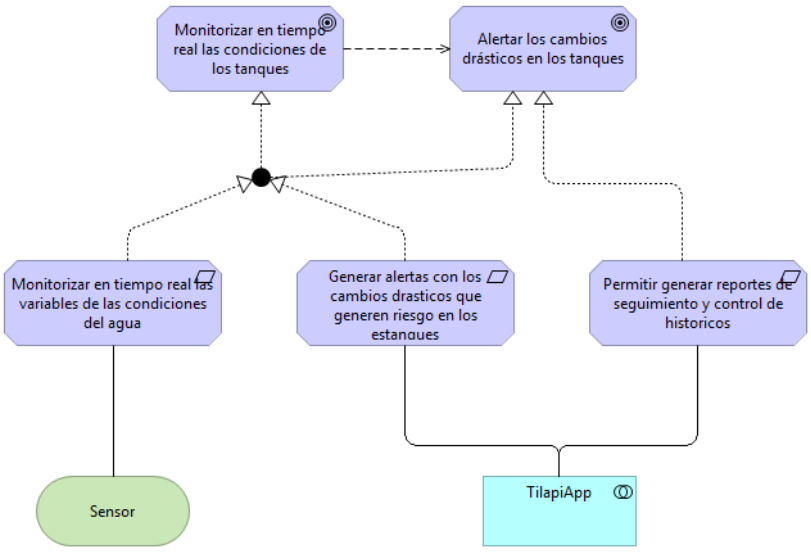


Imagen 33 Punto de Vista de Realización de Requerimientos – Fuente: Los Autores

2.6.6 Punto de Vista de Motivación

Este modelo recopila de manera general todos los elementos que fueron en los puntos de vista anteriores, por lo que puede brindar un panorama de cuál es la motivación, lo que la desencadena o drivers, Stakeholders y demás.

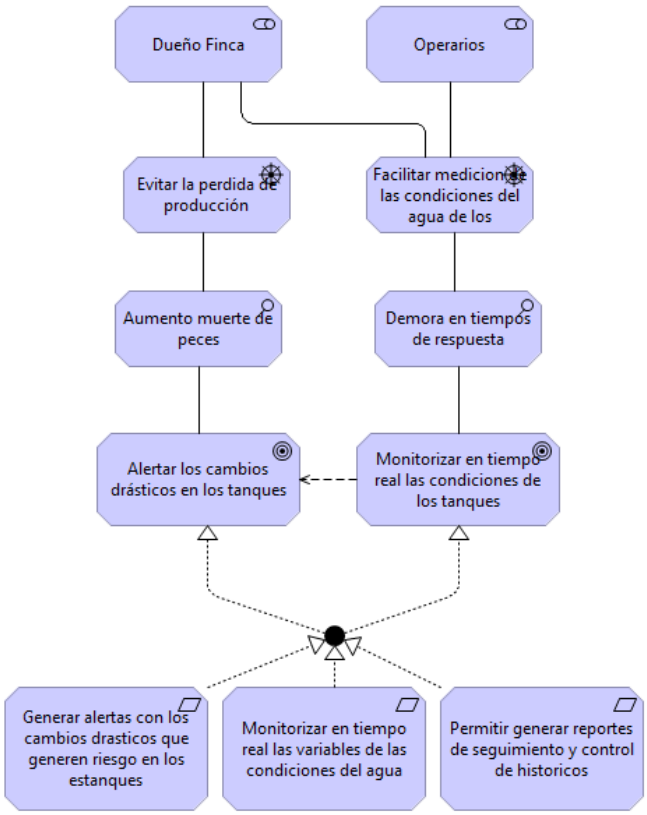


Imagen 34 Punto de Vista de Motivación – Fuente: Los Autores

2.7 CAPA DE MIGRACIÓN Y DESPLIEGUE

Una vez se concluye con los desarrollos de aplicaciones o implementación de nuevas tecnologías, están deben estar publicadas para que puedan ser accedidas o consultadas, pero para ello es vital contar con la información sobre cuál va a ser esa hoja de ruta y de qué manera van a ser desplegados los servicios. Por tal razón, esta capa nos muestra los artefactos y de qué manera van a ser puestos en producción o en el ambiente que se definido para tal fin.

2.7.1 Punto de Vista de Proyecto

En este modelo se puede observar cuales son los entregables del proyecto en términos de artefactos y como son su relación con las metas motivacionales, de tal manera que sean cumplidas y se pueda hacer seguimiento a su ejecución.

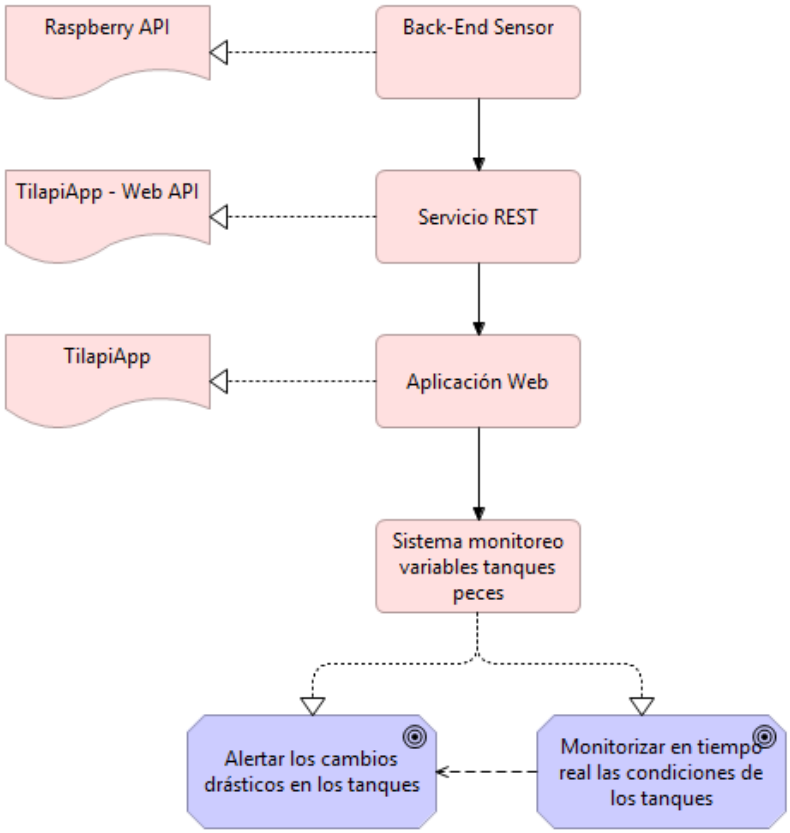


Imagen 35 Punto de Vista de Proyecto – Fuente: Los Autores

2.7.2 Punto de Vista de Migración

Este punto de vista sirve como hoja de ruta para la implementación tanto del Sensor como de Tilapiapp, permitiendo observar cuales son los artefactos de arquitectura y como se encuentran ubicados.

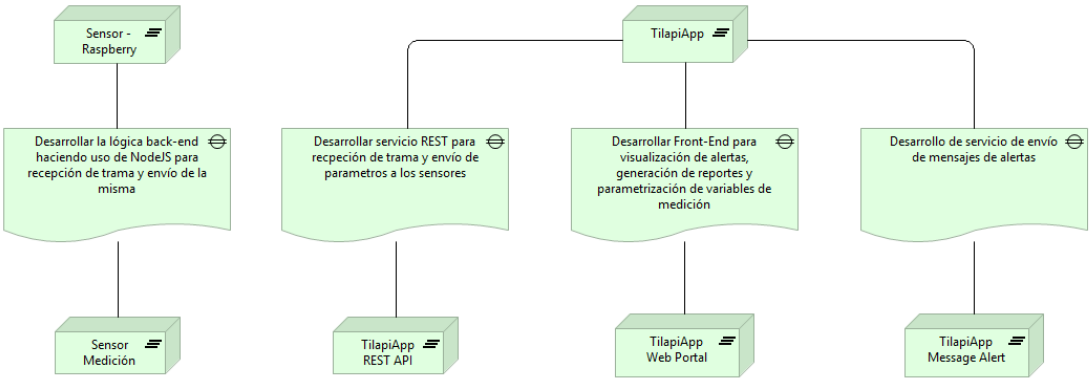


Imagen 36 Punto de Vista de Migración – Fuente: Los Autores

2.7.3 Punto de Vista de migración e implementación

Este punto de vista integra la migración y el proyecto, de tal manera que se puede tener claro cómo es su relación, siempre en pro de cumplir con la meta motivacional propuesta. Esta vista es el resultado de generar entregables y poner en marcha el prototipo propuesto para la Fundación Finca Allegro.

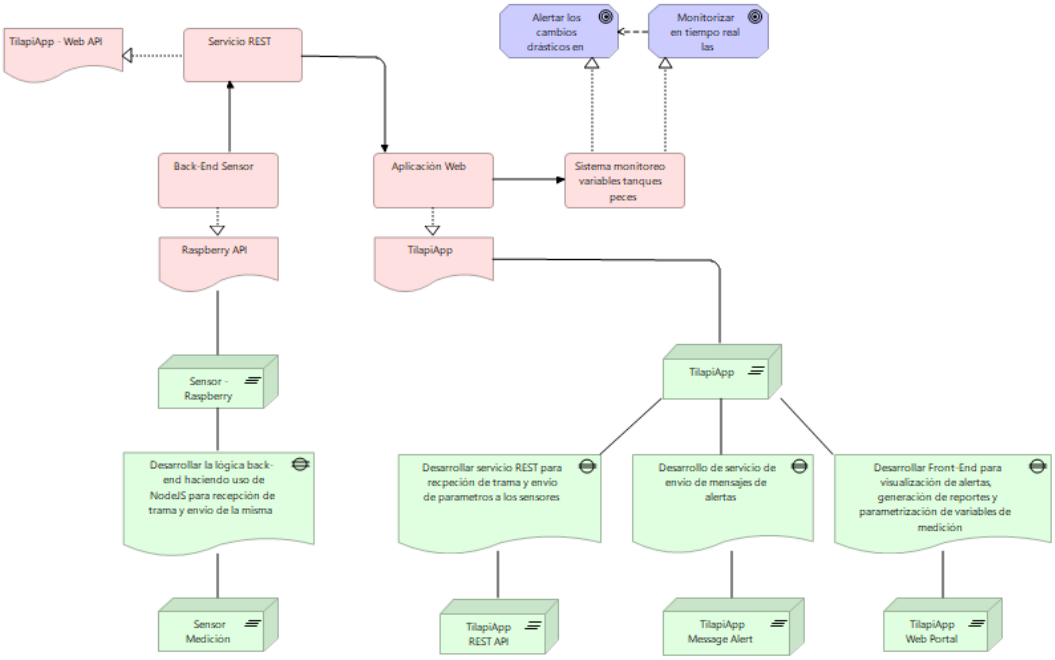
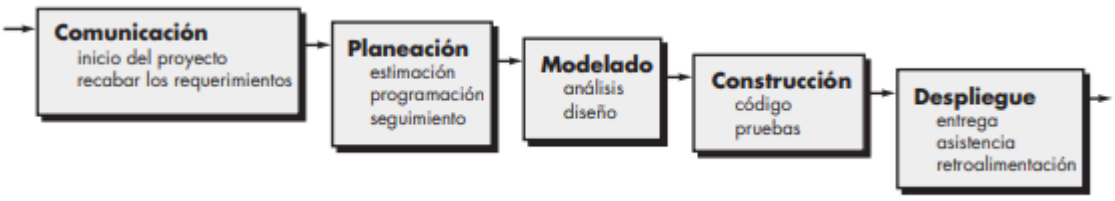


Imagen 37 Punto de Vista de migración e implementación – Fuente: Los Autores

METODOLOGÍA

El desarrollo de sistemas web se realiza bajo estándares o patrones definidos, que permiten a su vez establecer un proceso de control y calidad, siempre con el fin de entregar el mejor producto de cara al cliente. Ahora bien, para el caso del desarrollo del prototipo para la Finca Allegro, se tomó como modelo de desarrollo, el Modelo en Cascada, fundamentado por los siguientes factores:

- La ubicación y desplazamiento del equipo de ingenieros limita en gran parte la aplicación de metodologías ágiles que requieren de reuniones diarias y entregas constantes.
- El tiempo de entrega del prototipo debe realizarse de acuerdo con el cronograma establecido en la formulación del proyecto.



Ahora bien, el Modelo en Cascada se caracteriza por ser lineal, en donde se tienen pasos prestablecidos que requieren del anterior para continuar con su ciclo. El autor Roger Pressman [4] indica que este modelo es considerado clásico y se basa en enfoque sistémico y secuencial.

Así las cosas, el desarrollo del Prototipo se realizará con las siguientes fases:

1. Análisis

2. Diseño y Desarrollo
3. Despliegue

Para la codificación del aplicativo, tanto en la parte web como en la parte del dispositivo de lectura de datos, se implementa un modelo de desarrollo de software basado en capas con el fin de separar o desacoplar los componentes del sistema y permitir el principio de alta cohesión y bajo acoplamiento lo que permite llevar el desarrollo a varios niveles. Para el caso que se realice o se presente algún cambio, solo se afectará la capa o el nivel en donde se encuentre esa lógica. [13].

Las capas que se implementan a lo largo del desarrollo del sistema son 3

- Presentación.
- Negocio.
- Datos.

En este proyecto y teniendo presente cual es objetivo principal y la necesidad que busca suplir, se establecen los siguientes componentes de trabajo:

1. **Hardware:** Integración del sensor desarrollado por la finca Allegro y la tarjeta Raspberry para recepción de los datos.
2. **Portal Web (Tilapiapp Web):** Aplicativo web que contiene las funcionalidades definidas con el usuario y permite su visualización e interacción.

3. **Servicio REST (Tilapiapp API):** Contiene lo lógica de implementación para los servicios que serán expuesto para que sean consumidos por la tarjeta Raspberry.
4. **Modulo API Raspberry (Tilapiapp Pi):** Contiene la lógica para la captura de información del sensor y el consumo del servicio de *Tilapiapp API.*

3.1 ESPECIFICACIÓN

3.1.1 Análisis de requerimientos

En esta primera etapa se utilizó como referencia para la identificación de los requerimientos una serie de encuestas realizadas al personal encargado de la recolección de las muestras de la calidad del agua en la finca Allegro.

En la actualidad las muestras son obtenidas a través de un proceso manual repetitivo utilizando una tarjeta electrónica la cual tiene acoplados 4 sensores de calidad de agua (Temperatura, pH, Oxígeno disuelto y Saturación), la lógica de la tarjeta consiste en leer los valores de los sensores por medio de un ciclo cada 20 segundos y son puestos a disposición del usuario a través de una interfaz serial TTL.

El operario tiene que ir hasta el lugar donde se encuentran los tanques, conectar un computador a la interfaz serial de la tarjeta y capturar los datos para almacenarlos en un archivo .xlsx. El procedimiento anterior se realiza cada vez

que el operario tiene la oportunidad o se programa la revisión de 6 a 8 veces por día.

En el anexo A se presentan los formularios diligenciados por los operarios encargados de la recolección de los datos.

Haciendo un sondeo y una validación estadística sobre los resultados de las encuestas el 100% de los operarios coincidieron que el proceso no es tan eficiente cuando se hace manual mente ya que en los momentos que no se están realizando mediciones es donde se están presentando los cambios en la calidad del agua especialmente en horas de la noche y madrugada.

El administrador de la finca recomienda y sugiere que tener un monitoreo constante que alerte los cambios del agua es indispensable para cualquier tipo de solución que se desee implementar por lo que la generación de alertas es un requerimiento principal en el diseño del proceso de monitoreo automático.

Otro de los puntos importantes en el proceso de medición de la calidad del agua es la edad de los peces, cada tanque alberga un cardumen con una edad específica, por lo tanto, las referencias o valores de alerta son diferentes en cada uno. Por ejemplo, los peces recién nacidos (denominados alevinos [14]) no necesitan que el agua contenga niveles elevados de oxigeno contrario al agua de los tanques que albergan peces de mayor edad.

De acuerdo con lo anterior y como requerimiento del sistema, cada tanque que se parametrice en el sistema debe tener la opción de indicar los niveles de

los umbrales en los que las variables de la calidad del agua pueden afectar la vida de los peces.

Siguiendo con el análisis de los resultados de las encuestas realizadas, es de suma importancia presentar la información de tal manera que el operario pueda identificar y conocer los valores actuales de las variables al igual que los históricos para identificar el comportamiento detallado de la calidad del agua durante un día o un rango de días específicos. Al ser el cultivo un proyecto privado es necesario que el acceso al aplicativo este controlado por usuarios autorizados por ende la gestión de usuarios y roles se contempla como un requerimiento en el sistema de medición.

El siguiente es el diagrama de casos de uso general para el proyecto:



Imagen 38 Diagrama de casos de uso – Fuente: Los Autores

3.1.1.1 Listado de Requerimientos de Tilapiapp Web

Teniendo en cuenta el análisis descrito en la sección anterior se consolida y sintetiza el resultado en el siguiente listado de requerimientos enfocado al aspecto funcional del sistema, los cuales se detallan uno a uno en el anexo B (Casos de uso).

Tabla 3 Listado Requerimientos TilapiappWeb - Fuente: Los atuores

Requerimiento	**Descripción**	**Estimación (horas)**
Gestión de usuarios y acceso	Se requiere realizar un módulo de gestión de usuarios para Tilapiapp web con el fin de gestionar el acceso a la aplicación.	24
Gestión de tanques y umbrales	Se requiere realizar un módulo de gestión de parámetros para Tilapiapp web con el fin de crear y/o editar tanques y los umbrales asociados a las variables de medición.	32
Alertas y procesamiento de trama	Se requiere que el sistema Tilapiapp procese y genere alertas a través de correos electrónicos cuando los niveles capturados en la trama	8

	sobrepasan los umbrales parametrizados para las variables de los tanques.	
Dashboard e historial de datos	Se requiere visualizar la información actual sobre las lecturas obtenidas por el dispositivo de lectura Raspberry a través de un Dashboard en Tilapiapp web.	36

3.1.1.2 Listado de Requerimientos de Tilapiapp Api

El siguiente listado de requerimientos corresponde a una solución de integración entre la aplicación (**TilapiappWeb**) y el dispositivo de adquisición (**TilapiappPi**) a través de servicios web. Los cuales deben estar alojados en el mismo ambiente de **TilapiappWeb** y deben ser visibles a través de la red en la que se encuentre el dispositivo **TilapiappPi**.

Tabla 4 Listado Requerimientos TilapiappPi - Fuente: Los autores

Requerimiento	**Descripción**	**Estimación (horas)**
Servicios web Tilapiapp (Api)	Se requiere realizar una API a través de servicios web RestFull que permita la	24

	integración con los dispositivos de adquisición en cada uno de los tanques.	

3.1.1.3 Listado de Requerimientos de TilapiappPi

De acuerdo con los casos de uso del Anexo B, se obtiene el siguiente listado de requerimientos pertenecientes a la aplicación que hará las veces de interprete de la trama recibida por serial a través de TTL y el envío de esta al sistema ***TilapiappWeb*** por medio del servicio RESTFull ***Tilapiapp Api*****.**

Tabla 5 Listado de requerimientos Tilapiapp Pi - Fuente: Los autores

Requerimiento	**Descripción**	**Estimación (horas)**
Autenticación con el servicio REST	Se requiere que el API del sensor Raspberry se autentique con el servicio Rest de la aplicación Tilapia Web.	8

Obtener parametrización para el sensor, de acuerdo con el tanque en que se encuentre instalado	Se requiere que el API del sensor Raspberry obtenga los parámetros de envío de información y valores máximos y mínimos de las variables de medición	12
Leer trama del sensor serial	Se requiere que el API del Raspberry obtenga la trama del sensor del tanque por medio de protocolo serial.	24
Enviar trama y actualizar estado del sensor	Se requiere que el API del sensor Raspberry envíe la trama a la aplicación Tilapiapp Web para que sea procesada y analizada. Asimismo, se hace la actualización del estado del sensor	12

3.2 DISEÑO Y DESARROLLO

El diseño y desarrollo de las tres etapas que tiene el sistema de monitoreo de la calidad del agua está dividido por componente y a su vez por capas dependiendo la funcionalidad.

3.2.1 TilapiappWeb

3.2.1.1 Capa de datos

La capa de datos del aplicativo web proporciona toda la estructura de persistencia donde se almacena toda la parametrización del aplicativo junto con los datos recolectados por los dispositivos asociados a los tanques.

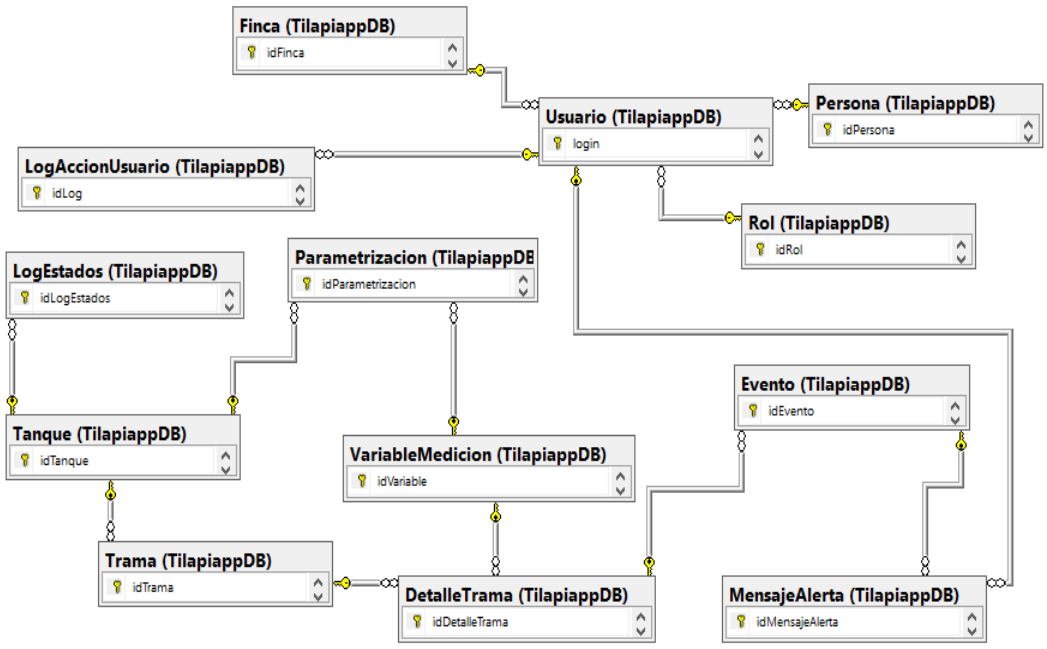


Imagen 39 Modelo relacional Tilapiapp Web – Fuente: Los autores

El diseño del modelo relacional del aplicativo está basado en dependencias funcionales mixtas (DFE -DFNE) ya que la relación entre las entidades está dada por llaves compuestas mixtas que permiten entrelazar varios procesos y funciones del sistema [15].

Este modelo de datos se utiliza para almacenar toda la información de parametrización recolectada directamente en el aplicativo web y toda la información que sea inyectada a través de las consultas que realice el dispositivo **TilapiappPi** a través de **TilapiappApi**.

El motor de base de datos que se utiliza para el desarrollo de este proyecto es *Sql Server Express 2017* el cual posee un licenciamiento gratuito para bases de datos que no superen un tamaño de 10 GB. Para este sistema, esta versión de Sql cumple con lo necesario para la implementación.

Para la creación de la base de datos (tilapiappdb) se generó un Script .sql (ver Anexo C) indicando el nombre de cada una de las tablas, relaciones y atributos correspondientes para ser ejecutado mediante un procedimiento en bloque con el fin de registrar los objetos a través del cliente de la base de datos *Sql Server Management Studio 2017.*

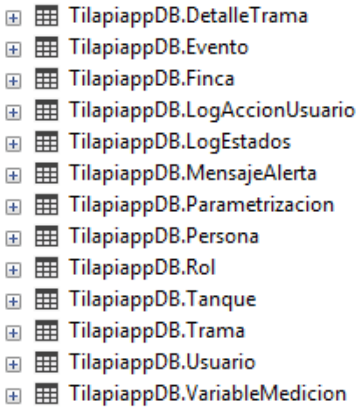


Imagen 40 Listado de tablas directamente en la base de datos - Fuente: Los Autores

La integración con la base de datos de SQL con el entorno de programación *IDE Visual Studio Community 2017* se hace a través del componente de Microsoft *EntityFrameworkCore* el cual permite la conexión directa con la base de datos y realizar un proceso de mapeo de datos en objetos creados como clases los cuales manejan la misma estructura de las tablas creadas en la capa de datos. Para la implementación se configura una clase de contexto de datos la cual está encargada de manipular y convertir los datos directamente a objetos definidos a través de clases.

Mediante la creación de un proyecto basado en la plantilla de librería de clases de .net core 2.1 se realiza el montaje del contexto y las clases que representan las entidades de negocio del sistema web.

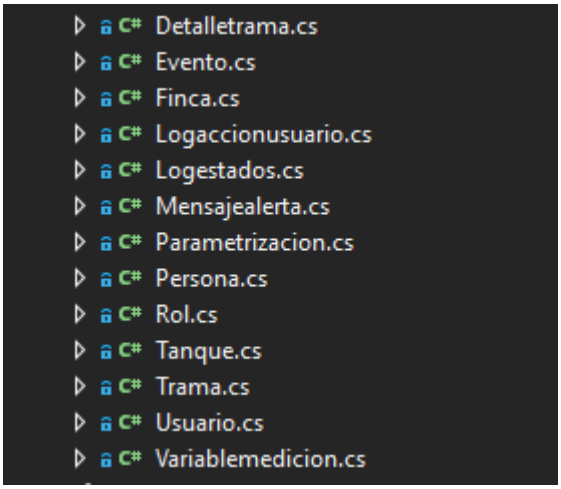


Imagen 41 Listado de clases de Visual Studio correspondientes a la capa de datos – Fuente: Los Autores

3.2.1.2 Capa de negocio

En la capa de negocio se especifica y se plasma el funcionamiento interno del aplicativo y la comunicación indirecta con la capa de datos teniendo en cuenta la lógica directa de cada una de las entidades que se tienen en la base de datos por lo que se genera para cada uno de los procesos de registro de datos un servicio de lógica CRU (Create, Read and Update).

Con el fin de generar informes y ver históricos de datos se opta por no eliminar datos que estén relacionados con información histórica en el aplicativo. Este proceso se realiza mediante la implementación del patrón de diseño *Fabricación* el cual consiste en la creación de clases que puedan ser delegadas a través de subclases o interfaces con el fin de eliminar el proceso de instanciación especifica cada vez que se requiera la comunicación con la capa de datos. [16]

En el siguiente esquema se presenta la distribución de las interfaces de control entre la capa de negocio y la capa de datos haciendo uso del patrón de fabricación.

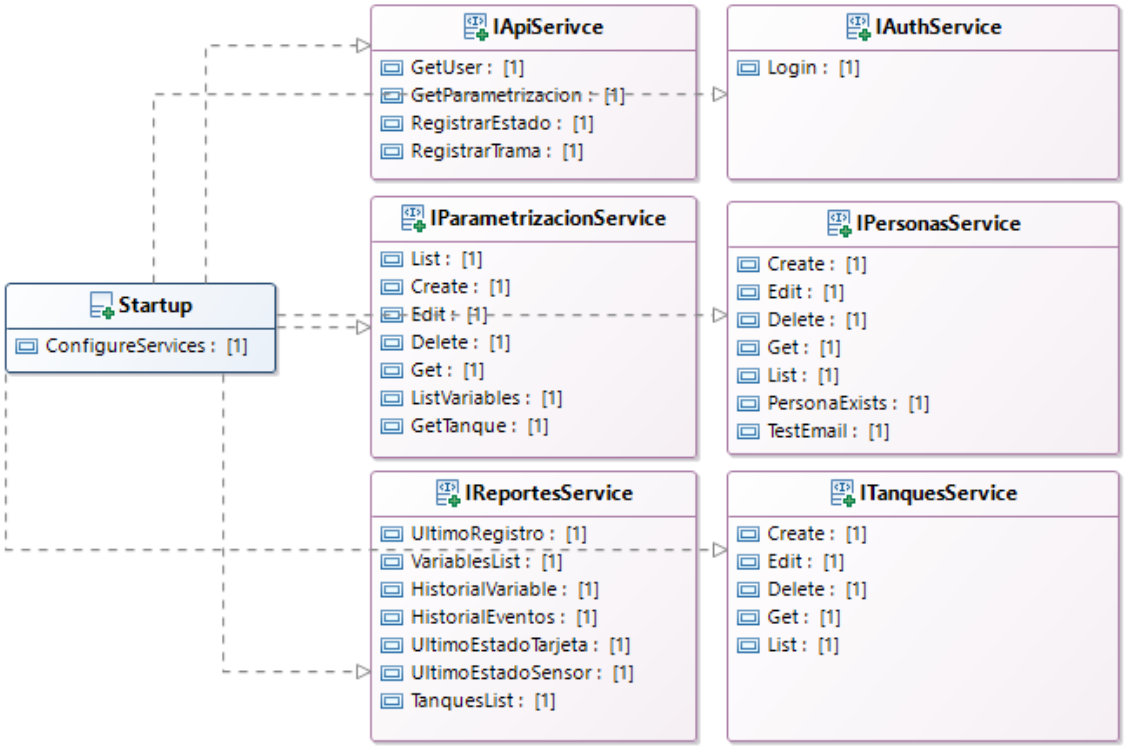


Imagen 42 Diagrama de clases correspondientes a las interfaces en el patrón de fabricación – Fuente: Los Autores

El lenguaje de desarrollo que se utiliza para la creación de este sistema es C# bajo el entorno de programación *Visual Studio Cummunity Edition 2017*, utilizando la tecnología .net core 2.1. Para la creación de la capa de negocio se utiliza una plantilla de proyecto de clases para .net core 2.1 en donde se procede a generar todas las clases e interfaces que estarán relacionadas con la operación

del sistema. El código completo de la solución podrá ser encontrado en el medio magnético adjunto a este documento.

Cada una de las clases relacionadas a las interfaces cumple una función directa almacenando o consultada información en la base de datos utilizando el contexto de la base de datos el cual es instanciado a través del proceso de inyección de dependencias implementado en los constructores de cada una de las clases de servicio de la capa de negocio.

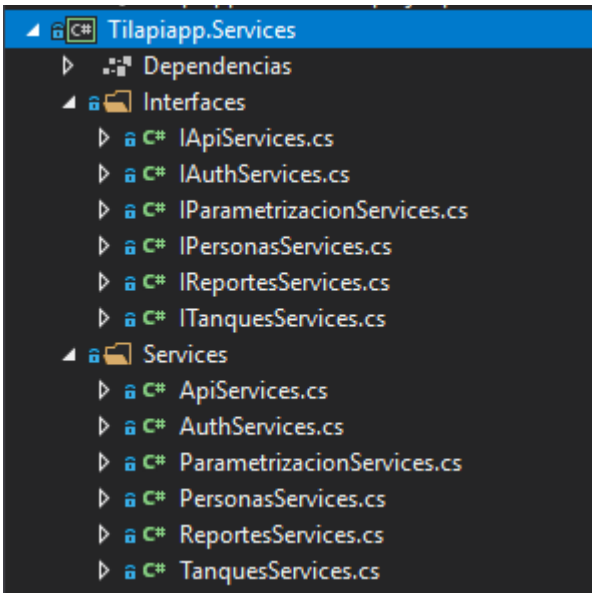


Imagen 43 Listado de clases e interfaces en la capa de negocio de Tilapiapp Web - Fuente : Los Autores.

En la capa de negocio se encuentra la lógica correspondiente para cada una de las siguientes funciones:

- Gestión de usuarios: **IPersonasService**.
- Gestión de acceso y autenticación: **IAuthService**.
- Gestión de parámetros de la aplicación: **IParametrizacionService**.

- Gestión de tanques: **ITanquesService**.
- Reportes e historial: **IReportesService**.
- Api: **IApiService**.

En el caso del servicio del api que es utilizada por el dispositivo de recolección de datos de los tanques, este controla y procesa la información que envían los sensores utilizando la parametrización de umbrales que deben estar previamente configurados por los usuarios a través de la capa de presentación.

Cuando se sobrepasa los umbrales parametrizados la lógica interna del servicio Api consulta la información de contacto de los usuarios y genera una serie de alertas que se envían a través de correo electrónico, el destino del envío de estas notificaciones es la bandeja de entrada de los correos electrónicos asociados a los usuarios registrados en Tilapiapp.

3.2.1.3 Capa de presentación

Teniendo en cuenta los requerimientos definidos en la etapa de análisis se realizan varios ejemplares visuales que permiten a los interesados tener una idea de lo que es la presentación del aplicativo web Tilapiapp durante las diferentes etapas y servicios que presta este sistema.

- Gestión de acceso y autenticación



Imagen 44 Scketch login de usuario – Fuente: Los Autores

- Gestión de usuarios



Imagen 45 Scketch Listado de usuarios del sistema – Fuente: Los Autores

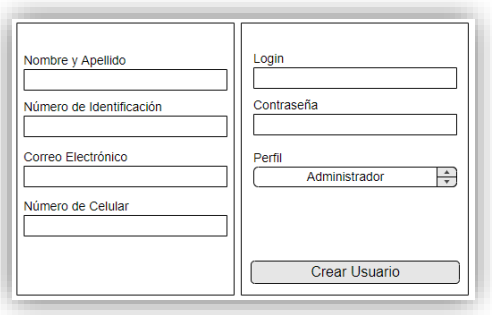


Imagen 46 Formulario de creación / edición de usuarios - Fuente: Los Autores

- Gestión de tanques y parametrización

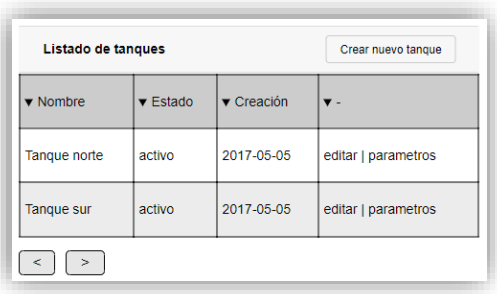


Imagen 47 Scketch Listado de tanques - Fuente: Los Autores

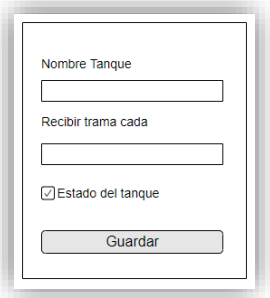


Imagen 48 Scketch Formulario de creación / edición de tanques – Fuente: Los Autores

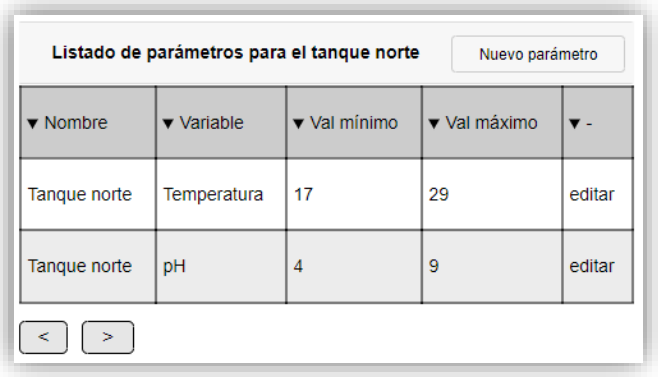


Imagen 49 Scketch Listado de parámetros de tanques - Fuente: Los Autores



Imagen 50 Scketch Formulario de creación / edición de parámetros de tanques

- Reportes e historial



Imagen 51 Scketch Dashboard de tanques - Fuente: Los Autores

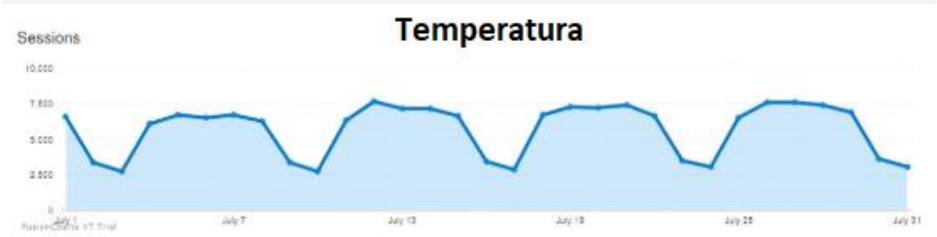


Imagen 52 Scketch representación gráfica - Fuente: Los Autores

#ID	Username	First Name	Last Name
200	janedoe	Yane	Doe
100	zanedoe	Jane	Moe
300	aanedoe	Lane	Doe
100	canedoe	Jane	Boe
100	janedoe	Jane	Doe

Entries per Page: 5 ▾ « 1 2 3 4 5 6 7 »

Imagen 53 Scketch Tabulación de resultados - Fuente: Los Autores

La capa de presentación se desarrolla en el entorno de programación *Visual Studio Community 2017* haciendo uso de una de las plantillas enfocadas a desarrollo web la cual utilizar el patrón de diseño MVC (Model View Controller o Modelo Vista Controlador).

En este nivel, los modelos de esta capa se enlazan directamente a los modelos creados en la capa de datos con el fin de mantener definiciones únicas de los objetos.

Las vistas se realizan en base a los scketch realizados en la etapa de diseño utilizando el framework de front-end Bootstrap 3, la lógica de front – end se realiza utilizando el framework de Javascript JQuery 3.3, al ser un patrón de MVC los métodos de los controladores (llamados acciones), retornan un modelo de datos con la información que se representa en la vista. Las vistas se representan en archivos con extensión *.htmlcs.*

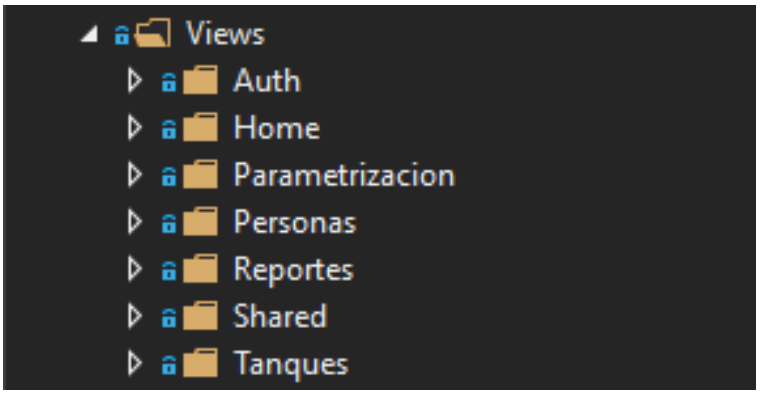


Imagen 54 Scketch Listado de vistas en el proyecto MVC

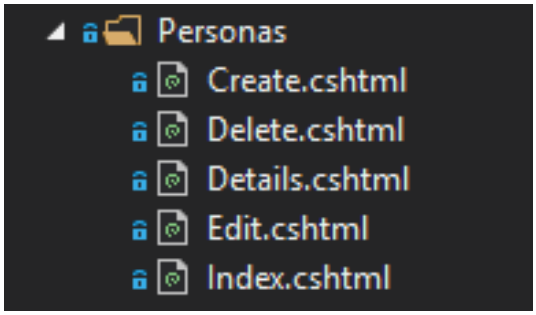


Imagen 55 Scketch Listado de vistas para el controlador personas – Fuente: Los Autores

3.2.2 TilapiappPi

Los sensores se encuentran conectados por medio de una tarjeta Raspberry Pi (3b o Zero). Cada uno de estos tienen como principales tareas tanto la captura de la trama, como el envío de esta para procesamiento posterior.

Debido al hardware del que se hace uso, se tienen restricciones para el uso de tecnologías que pueden usarse. Sin embargo, el abanico de posibilidades se extiende en el mundo de tecnologías Open Source.

A continuación, se muestra el diseño y desarrollo de las capas de Datos y Negocio, así como también el diagrama de la tarjeta y su integración con el sensor.

3.2.2.1 Capa de Datos

Para cada tarjeta Raspberry se define el siguiente modelo relacional el cual permite el almacenamiento de las tramas generadas por cada sensor, la parametrización obtenida de acuerdo con la configuración dada por los

administradores del sistema ***Tilapiapp Web*** obtenidas por medio del servicio REST y el control de envío de tramas al aplicativo Web.

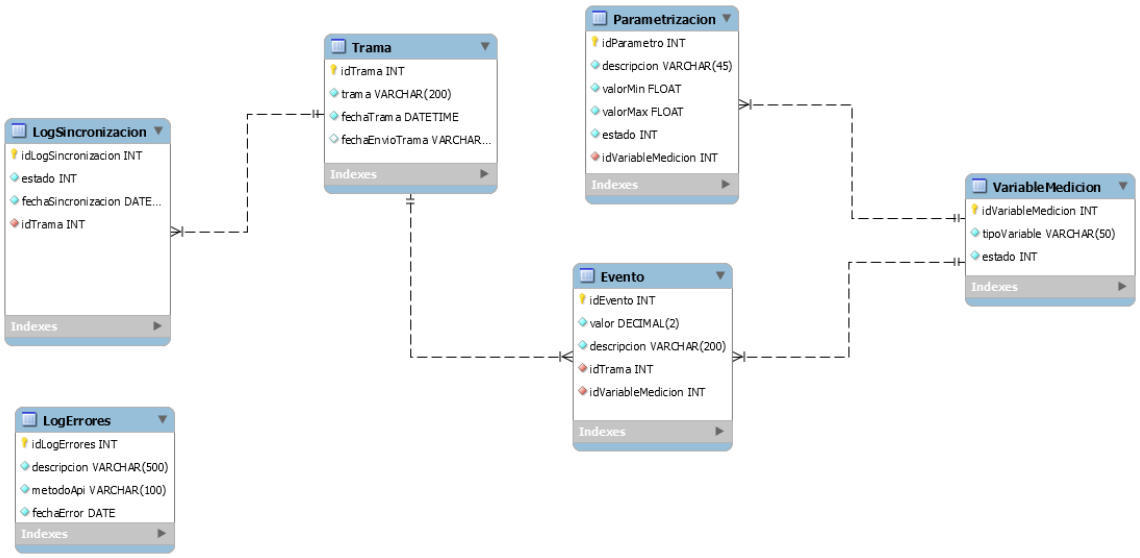


Imagen 2 Modelo relacional TilapiappPI – Fuente: Los autores

Este modelo se basa en un análisis de dependencias funcionales mixtas (DFE – DFNE) [15], pues se da una relación entre varias dependencias del modelo, lo que conlleva a la generación de las llaves primarias y foráneas para mantener su integridad y permitir el acceso a los datos.

El motor de base de datos que se utiliza en las tarjetas Raspberry es *MySQL* en su versión para *ARM* - Advanced RISC Machine -, pues es importante tener presente que estas tarjetas cuentan con un SoC de dicha característica. Adicionalmente, este motor es Open Source, lo que permite su uso sin adquirir licencias.

Los scripts de creación de la base de datos pueden ser consultados en el anexo C de este documento.

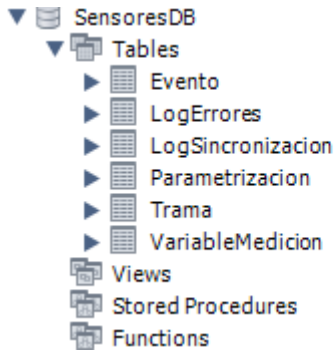


Imagen 56 Árbol de tablas SensoresDB - Fuente: Los Autores

Finalmente, se realizó la creación de Stored Procedures, con el fin de facilitar la inserción y extracción de los datos de la base de manera ágil, y a su vez permitir mejor integración con la lógica de negocio que se explica en la siguiente sección.

3.2.2.2 Capa de Negocio

En esta capa se maneja el control de la persistencia de la información y el envío de esta a la base de datos, la lógica de negocio y el uso del servicio REST con *Tilapiapp Api*, para lo cual se hace uso del siguiente modelo de paquetes.

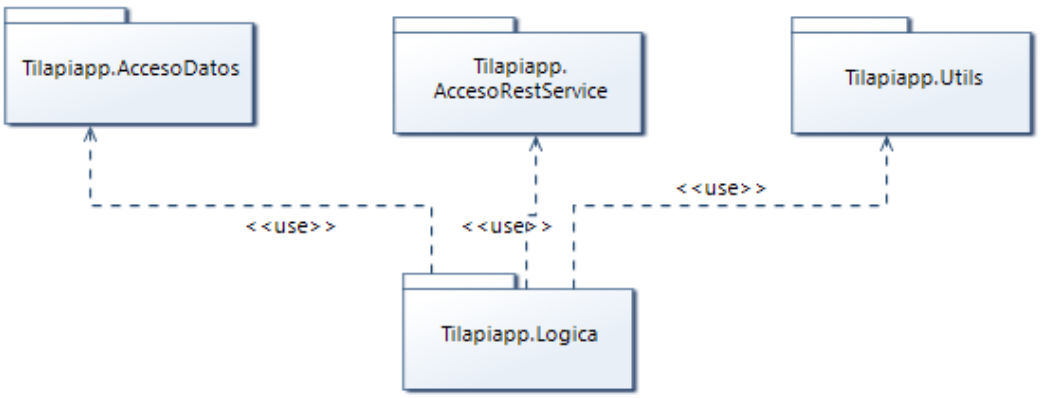


Imagen 57 Modelo de paquetes Tilapiapp PI - Fuente: Los autores

En donde,

- Tilapiapp.AccesoDatos: Contiene todo el acceso a la capa de datos.
- Tilapiapp.AccesoRestService: Contiene las clases de acceso a los servicios expuestos por *Tilapiapp PI.*
- Tilapiapp.Utils: Contiene las clases con utilidades.
- Tilapiapp.Logica: Contiene las clases con la lógica de negocio establecida por los requerimientos funcionales expuestos en el capítulo de Análisis de este documento.

Para el desarrollo de esta capa se hace uso de la tecnología Node.js [17], el cual es un entorno en tiempo de ejecución multiplataforma, que permite el uso de ECMAScript o JavaScript de manera asíncrona, proporcionando que las tareas y procesamiento de información se ejecute por medio de hilos.

Node.js trabaja por medio de paquetes, lo que vendría siendo el equivalente a librerías en lenguajes como Java o C#, las cuales permiten cargar funcionalidades ya creadas por terceros. Para este proyecto se hace uso de los siguientes paquetes:

- Serialport: Permite la trasmisión y recepción de paquetes por medio del protocolo serial TTL.
- Mysql: Permite la conexión a la base de datos MySQL
- Yargs: Librería que permite adicionar variables de entrada en la ejecución del programa desde Terminal o Command Prompt.

- Fs: Permite la manipulación de archivos y todo el sistema de archivos del SO. Para este caso, se hace uso en la creación de log de la aplicación.
- Request: Permite el uso de servicios rest.

Adicionalmente, y como se explicó en el capítulo de Metodología, se hace uso del principio de bajo acoplamiento y alta cohesión para el desarrollo.

En el caso de la conexión a la capa de datos, se hace uso del patrón Singleton, pues se instancia una única clase con los parámetros de conexión haciendo uso de la librería *mysql* una sola vez, y esta puede ser utilizada por toda la aplicación en cualquier momento sin tener que ser instanciada nuevamente.

El siguiente es el árbol del explorador de soluciones para el código de la aplicación Tilapiapp PI.

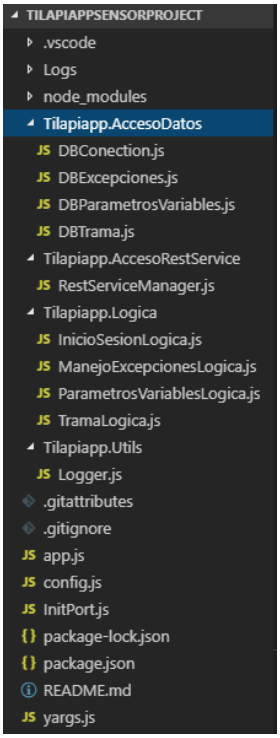


Imagen 58 Arbol explorador de solución Tilapiapp Pi - Fuente: Los Autores

El código completo de la solución puede ser consultado desde el medio magnético asociado a este documento.

3.2.2.3 Hardware de la tarjeta Raspberry y Sensor

Teniendo presente que la Finca Allegro cuenta con un sensor desarrollado especialmente por ellos para la medición de temperatura, oxígeno disuelto, saturación de oxígeno y pH en los tanques de tilapias, se hizo necesario implementar una solución a nivel de hardware capaz de permitir la interfaz con los servicios REST y a su vez obtener las tramas de medición, para lo cual se muestra a continuación el esquema del hardware utilizado para esta integración. En este

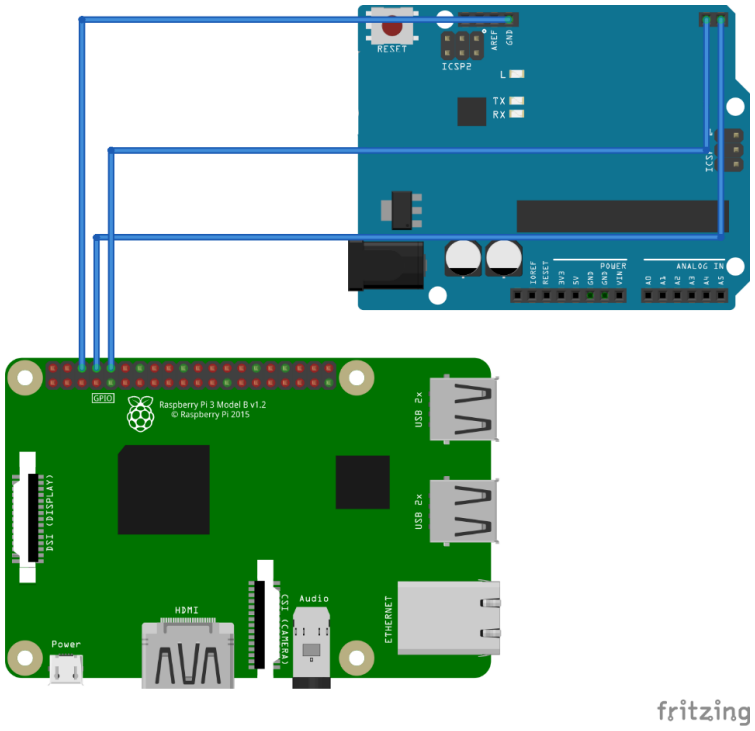


Imagen 59 Diagrama de conexión sensor y tarjeta - Fuente: Los Autores

3.3 DESPLIEGUE

3.3.1 Tilapiapp Web

Después de haber finalizado el desarrollo completo de TilapiappWeb se procede a realizar el proceso de despliegue y publicación del aplicativo. Para este proceso es necesario realizar una publicación de la aplicación Web a través de Visual Studio 2017.

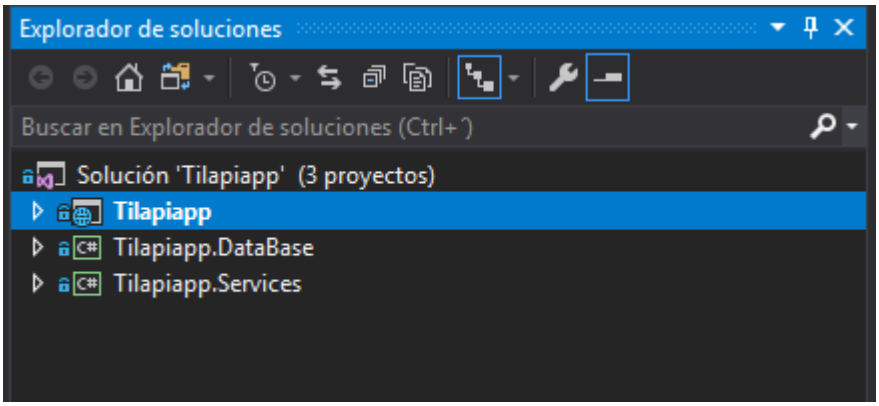


Imagen 60 Listado de capas del proyecto TilapiappWeb - Fuente: Los Autores

3.3.1.1 Publicación del aplicativo web

La publicación se realiza desde la capa de presentación del proyecto la cual, a través de la configuración .net core, tiene los componentes necesarios para el despliegue.

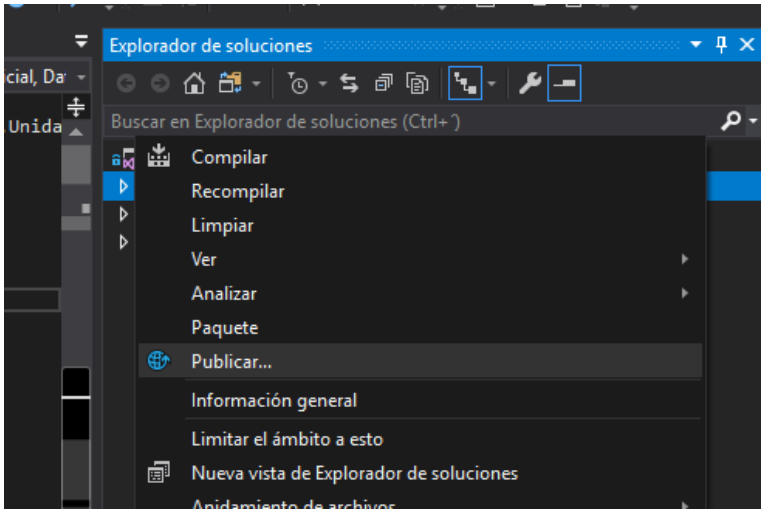


Imagen 61 Menú de opciones de la capa de presentación de Tilapiapp web – Fuente: Los autores

Dentro del menú que se despliega a través de la capa de presentación al presionar el botón derecho del mouse, aparece “publicar” o “publish”, esta opción permite abrir la sección de publicación de Visual Studio 2017, se debe seleccionar un perfil configurado para generar los archivos compilados en una ruta específica del computador de desarrollo.

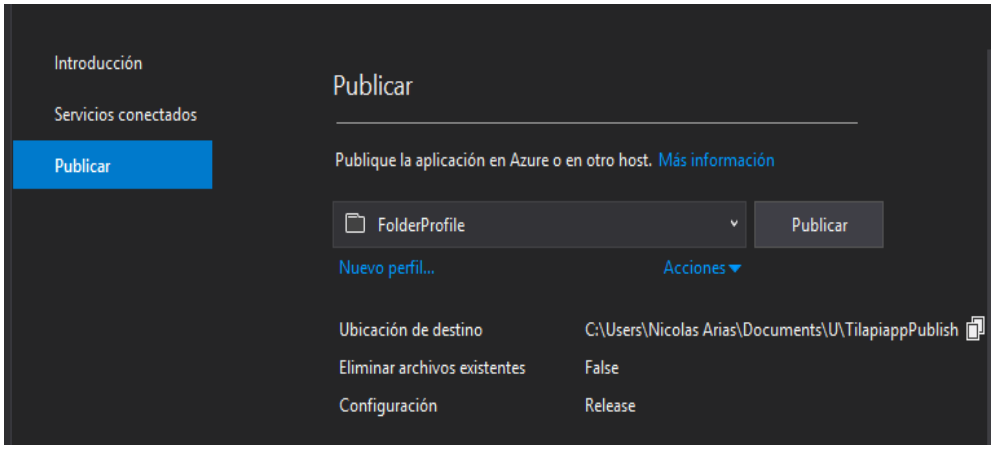


Imagen 62 Archivo de publicación en Visual Studio 2017- Fuente: Los autores

Se procede a ejecutar el procedimiento presionando el botón Publicar ubicado a la derecha de la sección de perfiles.

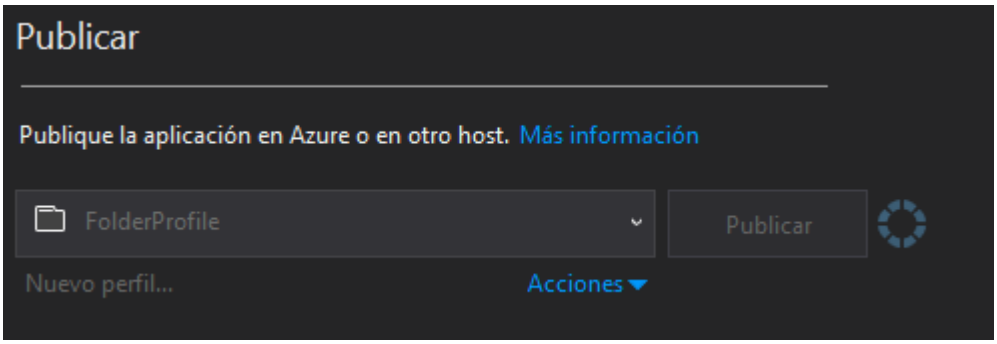


Imagen 63 Proceso de publicación en ejecución- Fuente: Los autores

Una vez este proceso finalice se podrán apreciar los archivos directamente en la ruta configurada para el perfil de publicación.

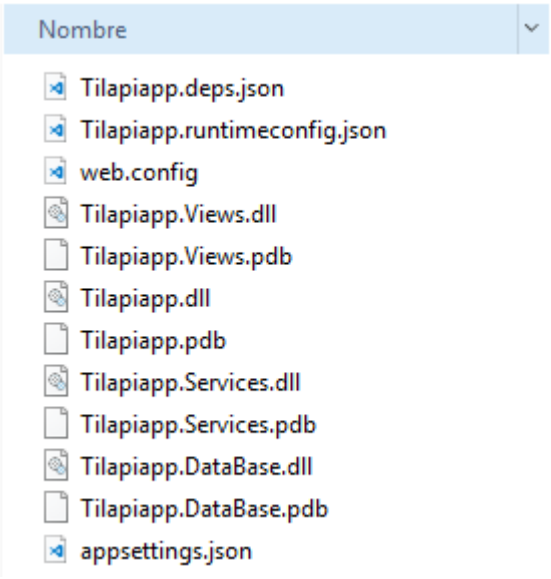


Imagen 64 Listado de archivos generados por el proceso de publicación - Fuente: Los autores.

3.3.1.2 Configuración del servidor web

Para este caso se utiliza el servidor de aplicaciones web propio y nativo de Windows Internet Information Services (IIS).

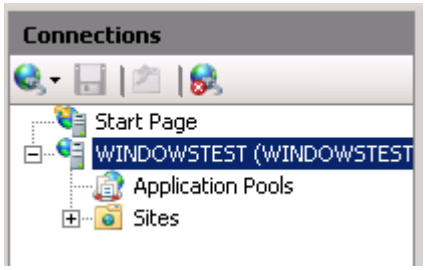


Imagen 65 Internet Information Services - Fuente: Los Autores

Se crea un nuevo sitio y se asigna como fuente de contenido la publicación realizada en el paso anterior.

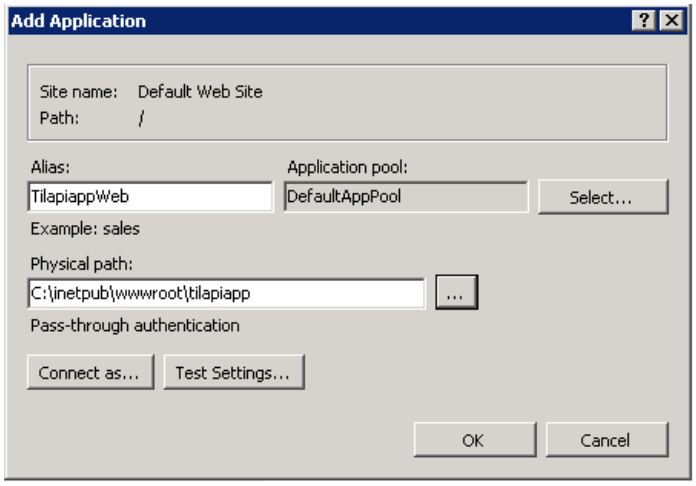


Imagen 66 Formulario de creación de nuevo sitio web en IIS - Fuente: Los autores

Una vez se encuentre el sitio web activo se puede ingresar a través del Alias configurado.

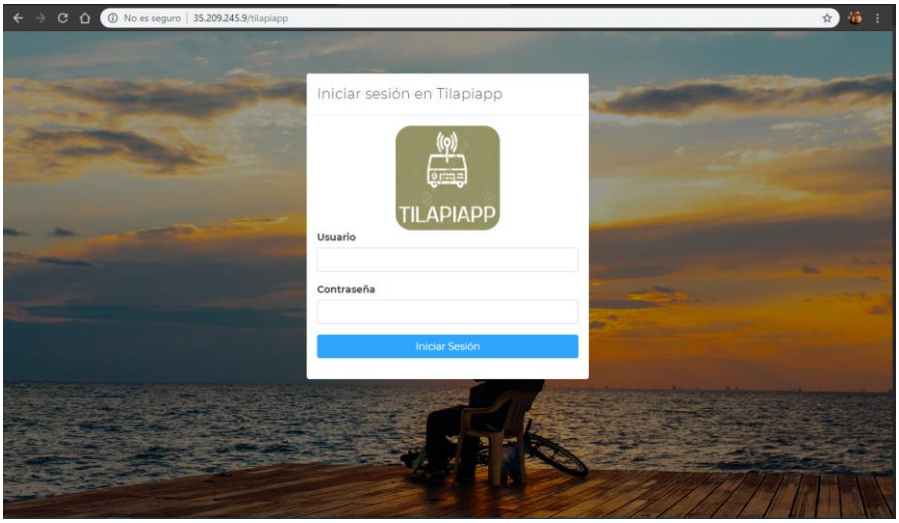


Imagen 67 Pantalla e Login de Tilapiapp Web – Fuente: Los Autores.

3.3.1.3 Configuración de la base de datos en SQL Server.

La tecnología utilizada en el desarrollo del aplicativo web contiene el complemento para el manejo y control de la base de datos directamente desde la inicialización del aplicativo web, este componente (EntityFrameworkCore) crea automáticamente los objetos y la base de datos la primera vez que se realiza una petición a través del navegador. Lo anterior facilita el proceso de implementación ya que solo es necesario la configuración de la cadena de conexión al servidor de base de datos. Esta configuración se realiza en el archivo appsettings.json.

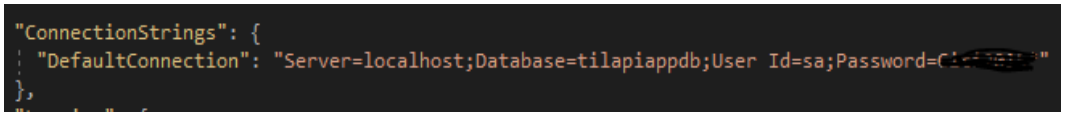


Imagen 68 Cadena de conexión de TilapiappWeb a la base de datos SQL Server - Fuente: Los Autores

Como instancia final en el proceso de implementación de TilapiappWeb se crea la base de datos automáticamente con la ayuda de EntityFrameworkCore en el servidor configurado.

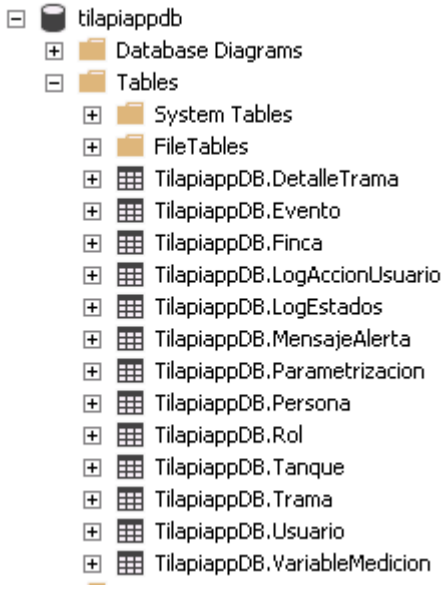


Imagen 69 Listado de tablas y objetos creados en la base de datos de Tilapiapp a través de Entity Framework Core - Fuente: Los Autores

3.3.2 Tilapiapp PI

Una vez se concluido el desarrollo de la API que se ejecutará desde las tarjetas Raspberry Pi, esta debe ser instalada en la misma, para lo cual como prerrequisitos son necesarios los siguientes:

- Habilitar acceso SSH y comunicación serial
- Instalar motor de base de datos MySQL.
- Instalar servicio de Node.js.

3.3.2.1 Habilitar acceso SSH y comunicación serial

Para habilitar estos servicios, debe realizarse teniendo la tarjeta conectada a un monitor (VGA – HDMI) y seguir los siguientes pasos:

1. Dar clic en el botón de Inicio → Preferencias → Configuración de Raspberry Pi

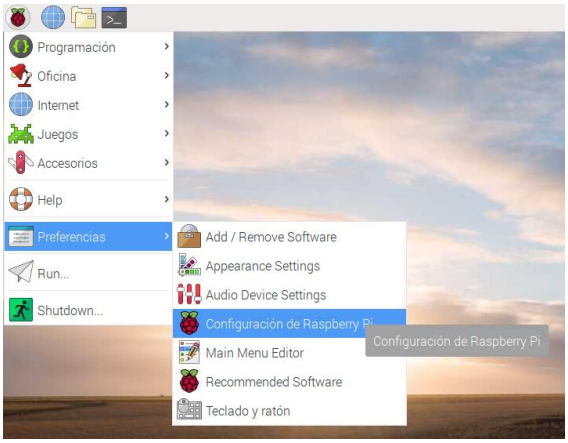


Imagen 70 Configuración Raspberry Pi 01 - Fuente: Los Autores

2. En la pestaña *Interfaz* deben estar en estado *Activo* las opciones Serial Port y SSH:

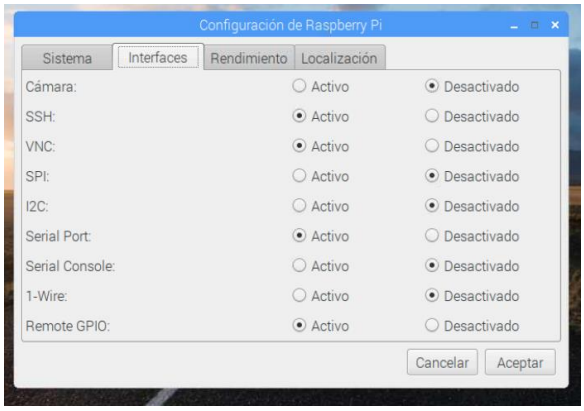


Imagen 71 Configuración Raspberry Pi 02 - Fuente: Los Autores

3. Dar clic en Aceptar y reiniciar la Raspberry si así lo solicita la configuración.

4. Para probar la conexión SSH puede hacerse uso del aplicativo *PuTTY* [18]. Ingresando la IP que tenga la Raspberry Pi y el puerto 22 y clic en el botón Open.

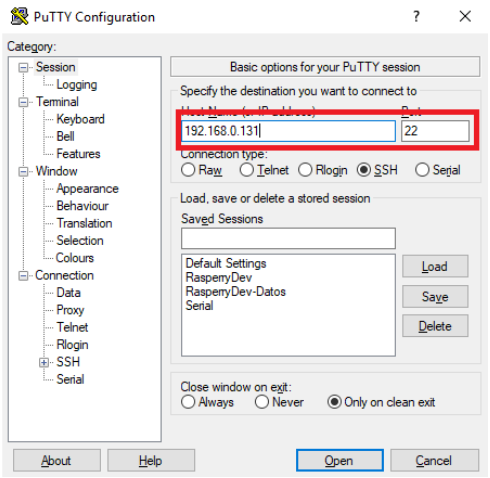


Imagen 72 Prueba conexión SSH Raspberry Pi 01 - Fuente: Los Autores

5. Ingresar usuario y contraseña, lo que permitirá el acceso a la Terminal de Linux para realizar las demás tareas.

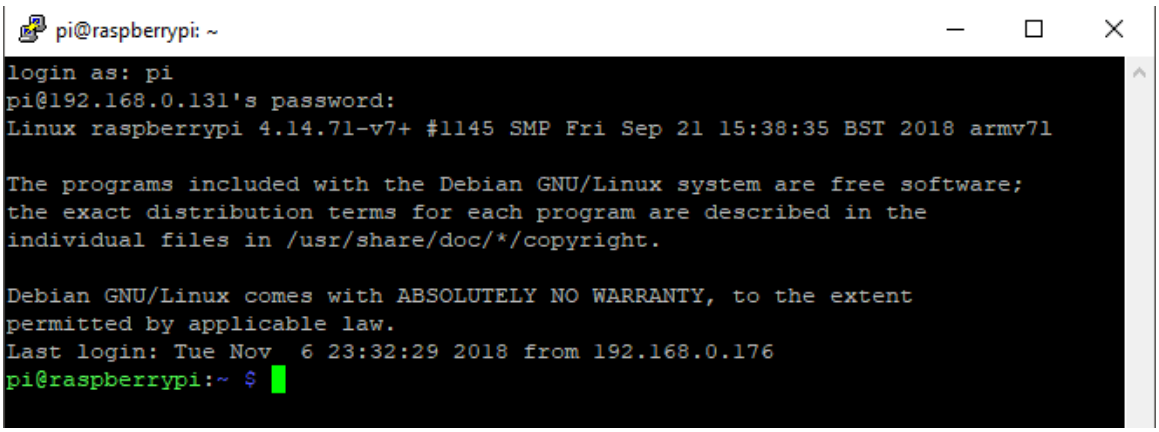


Imagen 73 Prueba conexión SSH Raspberry Pi 02 - Fuente: Los Autores

3.3.2.2 Instalación y Configuración del Motor de Base de Datos MySQL

Para realizar la instalación de MySQL en el Raspberry deben seguirse los siguientes pasos. Ejecutar desde el cliente SSH o en Terminal desde el SO de la tarjeta.

1. Ejecutar las siguientes líneas de código para actualizar el SO de la tarjeta.

```
sudo apt-get update && sudo apt-get upgrade
```

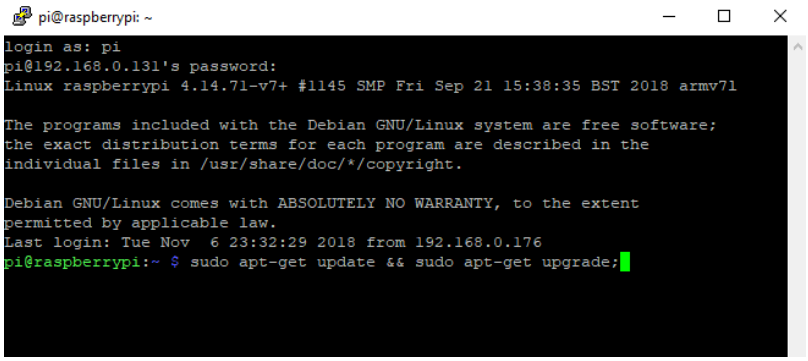


Imagen 74 Configuración MySQL Raspberry Pi 01 - Fuente: Los Autores

2. Ejecutar la siguiente línea para instalar el motor de DB.

```
sudo apt-get install mysql-server --fix-missing
```

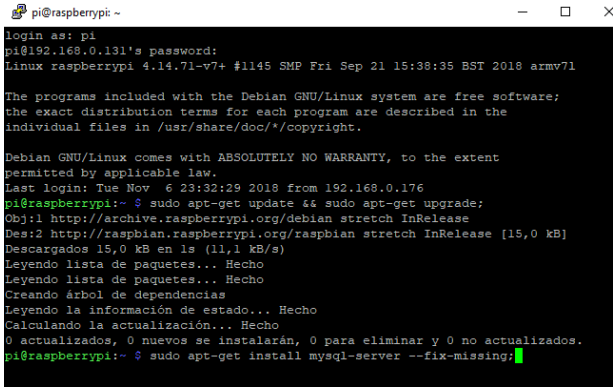


Imagen 75 Configuración MySQL Raspberry Pi 02 - Fuente: Los Autores

3. Ingresar al motor y crear el usuario pi con acceso para localhost y remoto:

```
sudo mysql -u root -p
```

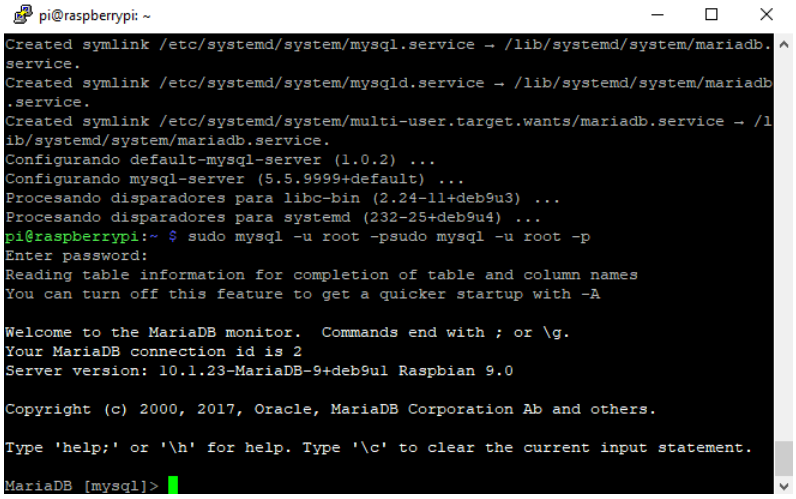


Imagen 76 Configuración MySQL Raspberry Pi 03 - Fuente: Los Autores

4. Ejecutar los siguientes scripts:

```
CREATE DATABASE `SensoresDB`;
CREATE USER 'pi' IDENTIFIED BY 'Tilapi@ppPi';
GRANT USAGE ON *.* TO 'pi'@'%' IDENTIFIED BY 'Tilapi@ppPi';
GRANT ALL privileges ON `SensoresDB`.* TO 'pi'@'%';
FLUSH PRIVILEGES;
```

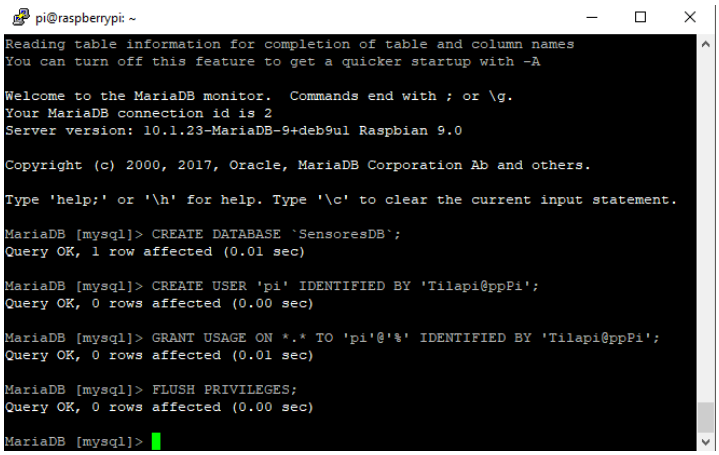


Imagen 77 Configuración MySQL Raspberry Pi 04 - Fuente: Los Autores

5. Ejecutar el código de creación de la base de datos SensoresDB, los procedimientos almacenados y variables iniciales (Estos scripts pueden ser consultados en el medio magnético adjunto a este documento). Para la ejecución se puede hacer uso de la Terminal o de algún entorno como MySQL Workbench [19]

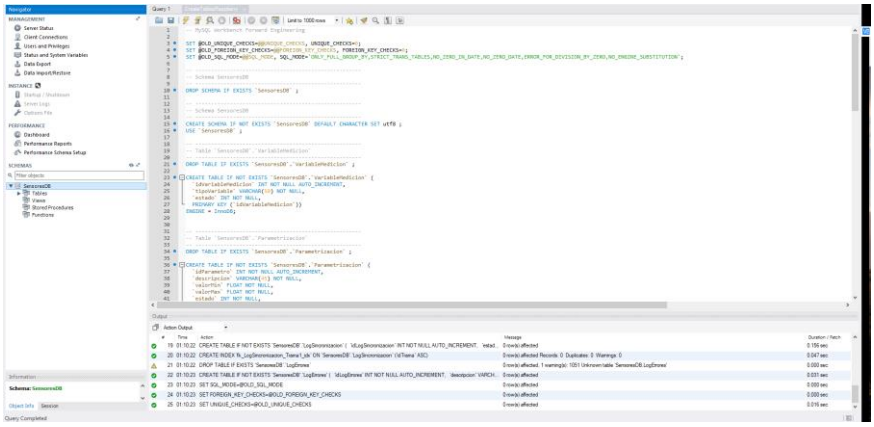


Imagen 78 Configuración DB SensoresDB 01 - Fuente: Los Autores

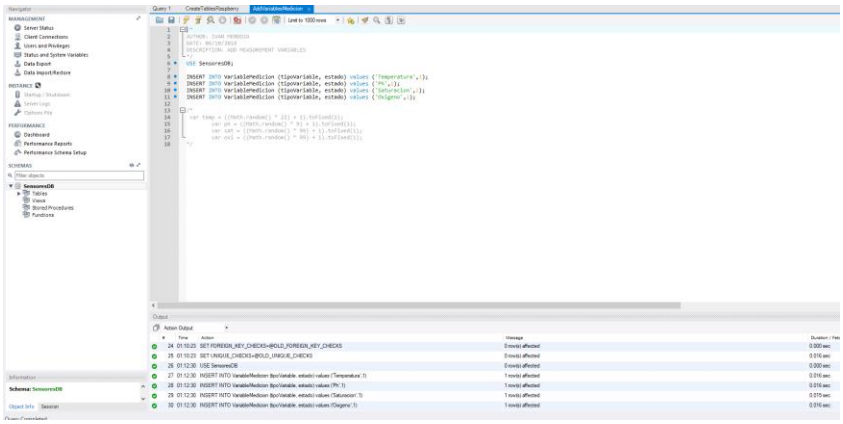


Imagen 79 Configuración DB SensoresDB 02 - Fuente: Los Autores

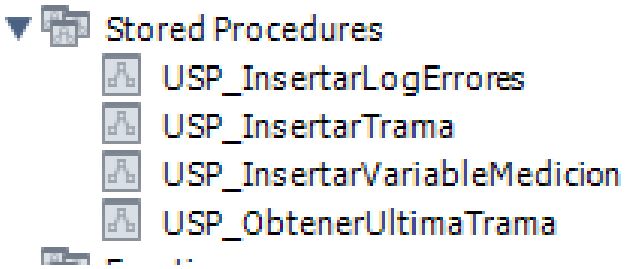


Imagen 80 Configuración DB SensoresDB 03 - Fuente: Los Autores

3.3.2.3 Instalación servicio Node.js

Para realizar la instalación de Node.js en el Raspberry deben seguirse los siguientes pasos. Ejecutar desde el cliente SSH o en Terminal desde el SO de la tarjeta.

1. Ejecutar los siguientes comandos en Terminal

```
curl -sL https://deb.nodesource.com/setup_11.x | sudo -E bash -
sudo apt install -y nodejs
```



Imagen 81 Instalación Node.js 01 - Fuente: Los Autores

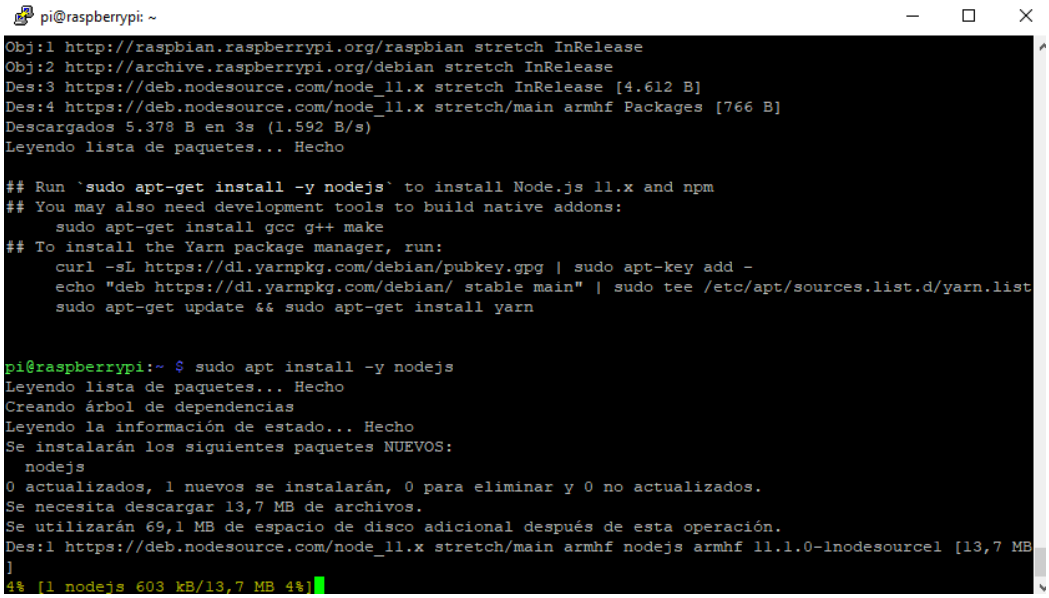


Imagen 82 Instalación Node.js 02 - Fuente: Los Autores

2. Ejecutar el siguiente comando para validar la instalación del servicio de Node.js

```
node -v
```

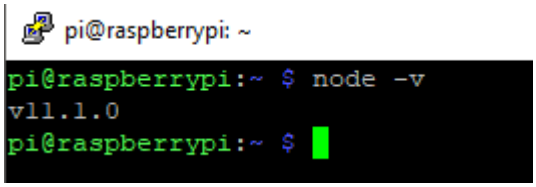
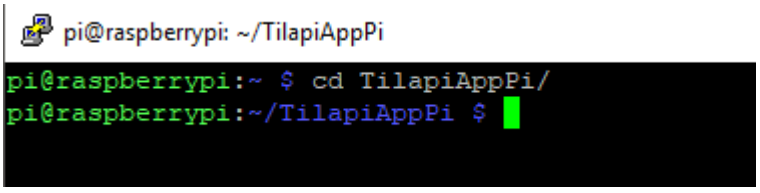


Imagen 83 Instalación Node.js 03 - Fuente: Los Autores

3.3.2.4 Instalación de la aplicación Desarrollada

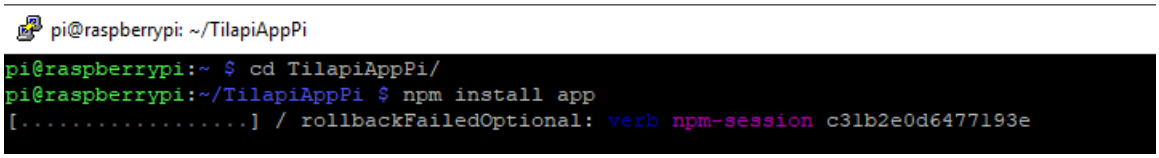
Una vez se tienen instalados los prerrequisitos de los puntos 3.3.2.1 al 3.3.2.4 es posible continuar con la instalación de la aplicación, para lo cual previamente se debe haber realizado la copia de la carpeta con los archivos en la tarjeta.

1. Ingresar por SSH y ubicar la carpeta con la aplicación ***TilapiAppPi***



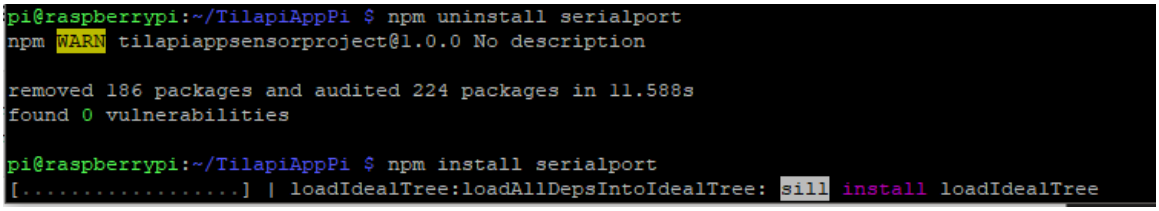
2. Ejecutar el siguiente comando, el cual instalará la aplicación y las dependencias o paquetes de Node.js requeridos para la ejecución del aplicativo

```
npm install app
```



3. Debido a que la aplicación fue desarrollada en Windows, se debe reinstalar el paquete **serialport** con los siguientes comandos:

```
npm uninstall serialport
npm install serialport
```



4. Para iniciar la aplicación se debe ejecutar el siguiente comando:

```
node app id idTanque us usuarioApp ps passUsarioApp
```

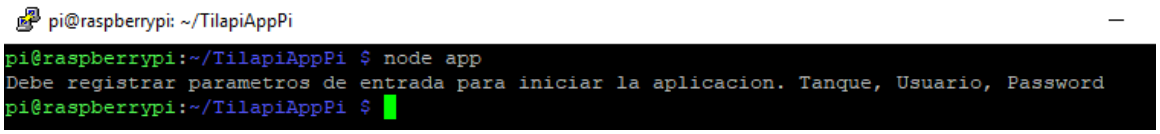
En donde,

idTanque: Identificador del tanque al que se encuentra asociado el sensor.

usuarioApp: Usuario creado en Tilapiapp Web.

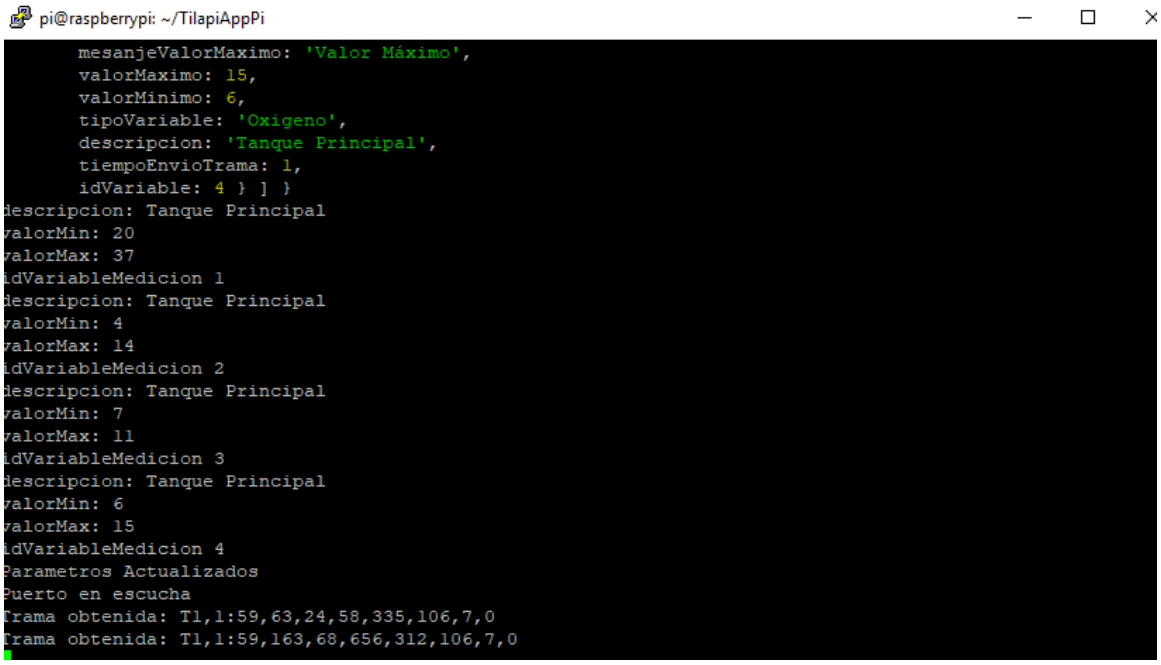
passUsarioApp: Contraseña del usuario de Tilapiapp Web.

Estos datos son importantes, ya que de no ser ingresados la aplicación no iniciará correctamente.





5. La aplicación queda con el puerto en escucha de acuerdo a los parámetros establecidos.



3.3.2.5 Archivos de configuración

La aplicación Tilapiapp Pi cuenta con dos archivos donde se ingresan los datos de conexión al servicio Rest y a la Base de Datos y son los siguientes:

- **/pi/TilapiAppPi/config.js:** Contiene la siguiente línea que debe modificarse únicamente si la url del servicio ha cambiado.

```
global.urlServicioWeb = 'http://192.168.0.176:5000/Api/';
```

- **/pi/TilapiAppPi/Tilapiapp.AccesoDatos/DBConection.js:** En la siguiente variable Json se encuentra información y únicamente debe ser cambiado si la configuración de la base de datos es modificada.

```
let DbConection = sql.createConnection({
    host: "localhost",
    user: "pi",
    password: "Tilapi@ppPi",
    database: "SensoresDB"
});
```

3.4 PRUEBAS

3.4.1 Información base – estado previo a la implementación.

La finca Allegro se encontraba realizando sus mediciones de manera manual, en donde un operario cada 4 horas se dirigía a los estanques para registrar y observar el resultado de las mediciones de los sensores.

Este proceso semi manual impedía tener la información en tiempo real sobre el estado de la calidad del agua de los estanques.

En el anexo C se presenta un ejemplo de planilla de datos obtenidos por los sensores y registrados por los operarios.



Imagen 84 Visualización de los sensores de medición en un estanque en la finca Allegro – Fuente: Finca Allegro

En la anterior imagen se presenta un ejemplo del sistema de medición de la calidad del agua propio de la finca Allegro y una presentación alterna de la data capturada.

En la fecha del 22 de octubre de 2018 en horas de la madrugada se presentó un decremento en el nivel de oxígeno disuelto en el agua, lo que causó la muerte de más de 2000 peces en uno de los estanques.



Imagen 85 Evidencia de muerte de peces debido a la falta de oxígeno – Fuente: Finca Allegro

La imagen anterior presenta una evidencia de peces muertos a raíz del inconveniente con el oxígeno en el tanque. Este tipo de eventos desafortunados en el proceso de cultivo de tilapias han sido repetitivos en la finca Allegro y pudieron haber sido evitados si se hubiera tomado el respectivo correctivo.

3.4.2 Pruebas de funcionalidad

Las pruebas que se realizaron directamente en la finca Allegro tuvieron cabida entre las 00:00 Am del 27 de octubre de 2018 hasta las 11:59 Pm del 27 de octubre de 2018.

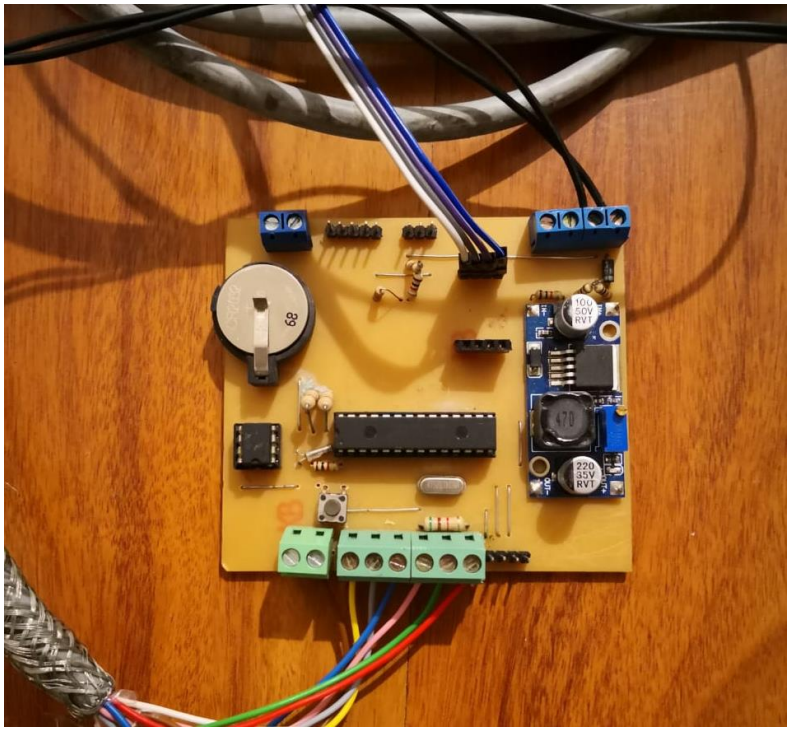


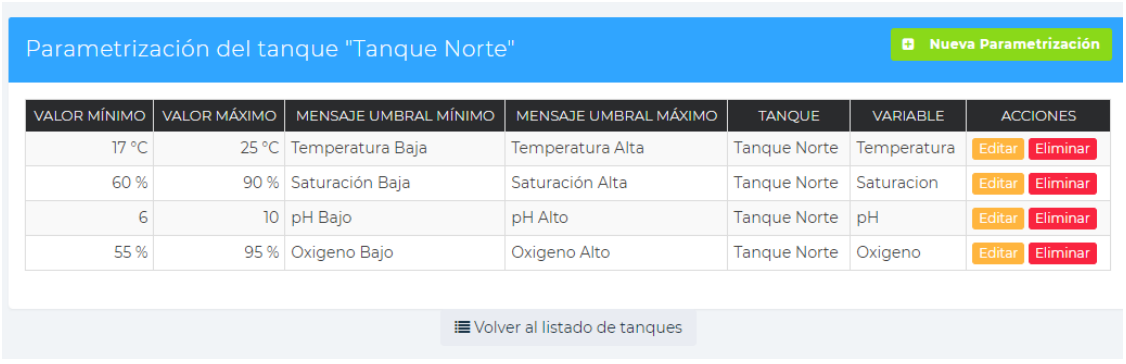
Imagen 86 Conexión tarjeta de sensores con RaspberryPi – Fuente: Los Autores

La imagen mostrada anteriormente presenta la conexión interna de la tarjeta de recolección y lectura de los sensores, la cual es adaptada a la tarjeta Raspberry Pi a través de la conexión serial. Para las pruebas fue necesaria una conexión directa al servidor de aplicaciones de Tilapiapp a través de internet. Se utilizó la misma estructura física de los sensores conectando directamente el dispositivo TilapiappPi compuesto por la tarjeta Raspberry Pi y la conexión a la tarjeta de los sensores utilizando el protocolo de comunicación serial.



Imagen 87 Cardumen de tilapias en el Tanque norte - Fuente: Finca Allegro

Se realizó la parametrización de umbrales permitidos como se muestra en la siguiente imagen, para cada una de las variables medidas por los sensores para el tanque norte, teniendo en cuenta lo indicado por el personal de la finca Allegro.



VALOR MÍNIMO	VALOR MÁXIMO	MENSAJE UMBRAL MÍNIMO	MENSAJE UMBRAL MÁXIMO	TANQUE	VARIABLE	ACCIONES
17 °C	25 °C	Temperatura Baja	Temperatura Alta	Tanque Norte	Temperatura	Editar Eliminar
60 %	90 %	Saturación Baja	Saturación Alta	Tanque Norte	Saturacion	Editar Eliminar
6	10	pH Bajo	pH Alto	Tanque Norte	pH	Editar Eliminar
55 %	95 %	Oxigeno Bajo	Oxigeno Alto	Tanque Norte	Oxigeno	Editar Eliminar

Imagen 88 Listado de parámetros asociados al tanque norte en Tilapiapp - Fuente: Los Autores

En la siguiente imagen se presenta el listado de usuarios creados a los cuales se les dio acceso a Tilapiapp para que pudieran monitorear desde su computador, las mediciones de los sensores.

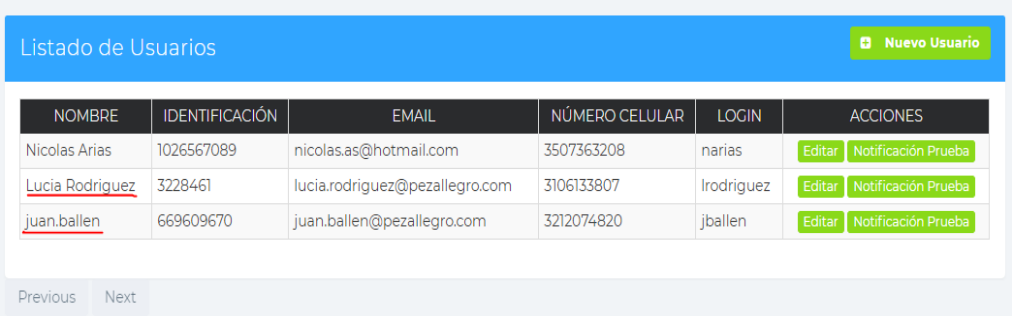


Imagen 89 Listado de usuarios creados en Tilapiapp

En la siguiente imagen se puede evidenciar la presentación del Dashboard donde se muestran las mediciones del último registro recibido por el dispositivo TilapiappPi.



Imagen 90 Dashboard Tilapiapp presentación de datos del sensor - Fuente: Los Autores

Durante la sesión de pruebas programada por 24 horas, se registraron una serie de alertas a los usuarios (presentadas en la siguiente imagen directamente desde Tilapiapp), correspondientes a niveles bajos de oxígeno en el agua lo que

les permitió ejecutar el proceso correctivo el cual consiste en prender un circulador y bombeador de agua para introducir oxígeno en el agua.

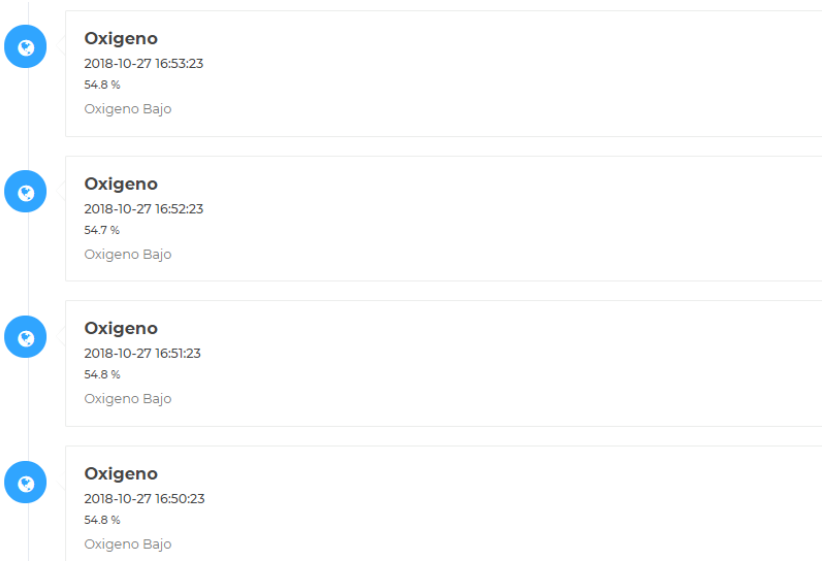


Imagen 91 Informe de eventos en Tilapiapp – Fuente: Los Autores

Como evidencia de recepción de las alertas a través del correo electrónico configurado para cada usuario se presentan las dos siguientes imágenes.



Imagen 92 Email de notificación de Tilapiapp por nivel bajo de oxigeno - Fuente: Los Autores

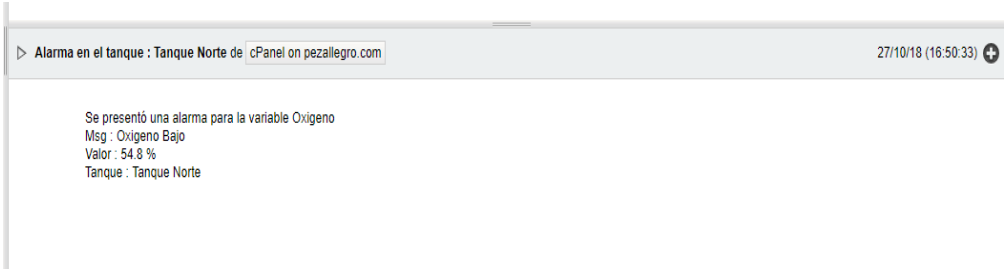


Imagen 93 Email de notificación de Tilapiapp por nivel bajo de oxigeno (2) - Fuente: Los Autores

Una vez los encargados recibieron la notificación vía email, se dirigieron al tanque norte y activaron el sistema de bombeo y oxigenación como se muestra a continuación, con el fin de aumentar el nivel de oxígeno en el agua.



Imagen 94 Tanque Norte con los sistemas de aireación activados - Fuente: Los autores

A continuación, se presenta como ejemplo los registros de los datos obtenidos correspondientes a la variable de oxígeno disuelto en el agua del tanque norte.

Tabla 6 Registros de Oxigeno promediados por hora del tanque norte el día 27 de octubre 2018 - Fuente: Los Autores

Variable	Valor	Fecha
Oxigeno	72%	2018-10-27 00
Oxigeno	72.3%	2018-10-27 01
Oxigeno	70%	2018-10-27 02
Oxigeno	70%	2018-10-27 03
Oxigeno	71.9%	2018-10-27 04
Oxigeno	69.2%	2018-10-27 05
Oxigeno	62%	2018-10-27 06
Oxigeno	62%	2018-10-27 07
Oxigeno	62%	2018-10-27 08
Oxigeno	62.4 %	2018-10-27 09
Oxigeno	60.7 %	2018-10-27 10
Oxigeno	59.1 %	2018-10-27 11

Oxigeno	60.3 %	2018-10-27 12
Oxigeno	61.5 %	2018-10-27 13
Oxigeno	60.4 %	2018-10-27 14
Oxigeno	59.5 %	2018-10-27 15
Oxigeno	54.8 %	2018-10-27 16
Oxigeno	56.8 %	2018-10-27 17
Oxigeno	65.2 %	2018-10-27 18
Oxigeno	71.9 %	2018-10-27 19
Oxigeno	70.2 %	2018-10-27 20
Oxigeno	70.4 %	2018-10-27 21
Oxigeno	72.6 %	2018-10-27 22
Oxigeno	71.3 %	2018-10-27 23

Utilizando la herramienta de historial de Tilapiapp se obtuvieron los anteriores datos y a su vez se presenta la gráfica del historial de mediciones del día 27 de octubre de 2018, en la siguiente imagen es notorio el comportamiento de las mediciones registradas por los sensores.

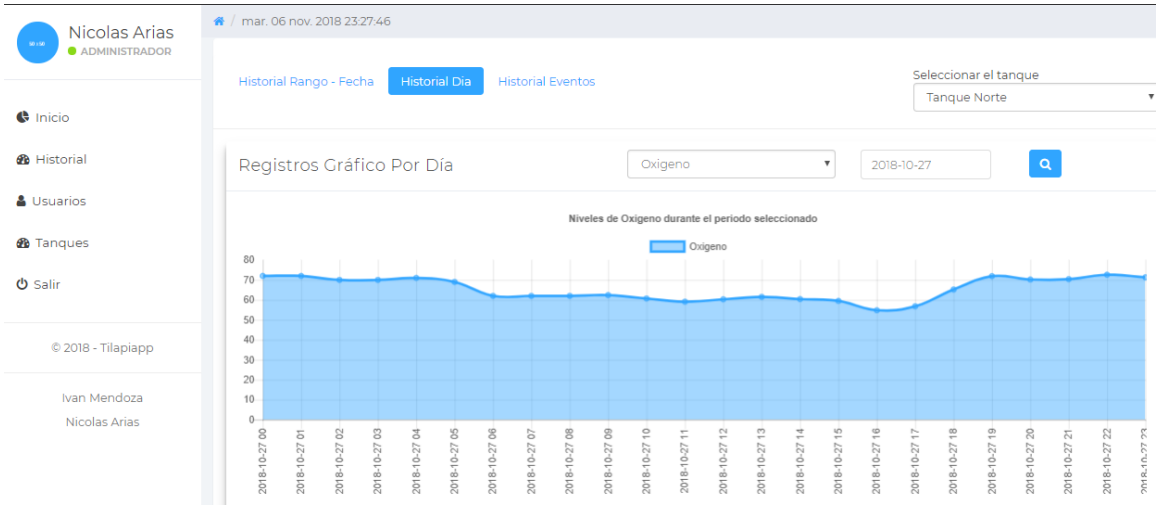


Imagen 95 Histórico de medicines de oxígeno en el tanque norte - 27 de octubre

Fuente: Los Autores

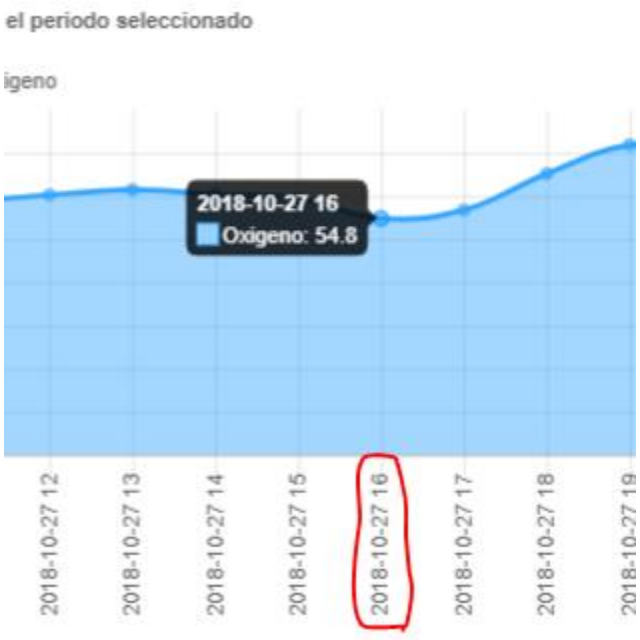


Imagen 96 Visualización de nivel inferior al umbral parametrizado en para el tanque norte. Fuente: Los Autores

Las imágenes 95 y 96 corresponden a la gráfica del histórico para la medición de la variable oxígeno en el tanque norte, en donde se puede observar cómo fue disminuyendo el oxígeno hasta un pico mínimo a las 4 pm y cómo aumento una vez realizadas las actividades correctivas por parte de los operarios.

De acuerdo con los resultados de las pruebas, es posible decir que se cumplió con notificar los cambios presentados en los umbrales parametrizados para el tanque norte, en donde fue posible actuar de manera oportuna, evitando así, que se presentaran situaciones desafortunadas como puede llegar a ser la muerte de los peces.

PARTE III. CIERRE DE LA INVESTIGACIÓN

CONCLUSIONES, TRABAJOS, FUTUROS APORTES, Y CONTRASTACIÓN

4.1 CONCLUSIONES

- La investigación permitió identificar las variables que influyen en la calidad del agua y que permiten el correcto desarrollo de la tilapia roja, así como también, tener una claridad sobre los procesos que ejecuta la Fundación Finca Allegro en el desarrollo de sus actividades, lo que sirvió de insumo clave para el desarrollo de Tilapiapp Web e integración con los sensores que utilizan.
- El uso de los microservicios permite mantener un bajo acoplamiento de los componentes, lo que hace que su implementación y mantenimiento sea más simple y eficiente, por lo que fue posible realizar un diseño web y un diseño local en Raspberry en lenguajes diferentes y que a su vez pudieran permanecer conectados.
- Los sistemas embebidos enfocados al internet de las cosas permiten a los usuarios no solo la automatización de procesos sino también la captura y almacenamiento de datos que pueden ser utilizados para

entender el comportamiento de un fenómeno o predecir eventos futuros.

- La implementación del prototipo del sistema Tilapiapp Web generó una reducción en el tiempo de acción frente a problemas que afectan la vida de los peces y se traduce en pérdidas económicas para el dueño, de tal manera que se garantiza la viabilidad del proyecto y su futura implementación en la totalidad de los tanques.

4.2 TRABAJOS FUTUROS

4.2.1 Ejecutar acciones remotamente

Partiendo del uso del Internet de las Cosas, es posible implementar un servicio en doble vía que permita ejecutar acciones automáticamente o a demanda para corregir alguna anomalía que se presente en el tanque, como por ejemplo activar un calentador de agua, agregar sustancias para nivelar pH u oxígenos, de tal manera que los riesgos puedan ser corregidos inmediata y automáticamente.

4.2.2 Alertas Visuales – Tilapiapp VA (Visual Alert)

Con el fin de tener un control aún mayor de los riesgos que se puedan llegar a presentar en cada uno de los tanques, se plantea la idea de generar alertas de tipo visual o auditiva directamente en la misma posición donde se encuentra el sensor, esto con el fin de tener una herramienta de contingencia en caso de que

se presenten fallas con el suministro eléctrico a la tarjeta Raspberry o perdida conexión a internet en el momento.

4.2.3 Alertar por medio de SMS

Implementar el reporte de las alertas generadas en los tanques por medio de mensajes de texto SMS [20]. Para esto, el mercado actual ofrece soluciones de pago a bajo costo que pueden ser utilizadas. Esta opción permite que las alertas lleguen a dispositivos que no cuentan con acceso a internet.

4.3 APORTES

Colombia al ser un país agrícola puede tener a su alcance para uso y beneficio el Internet de las Cosas, pues la implementación de estos sistemas en cultivos, criaderos y demás, puede permitir un mayor control de estos, así como también obtener datos importantes que pueden llegar a servir para hacer minería y analítica que puede llevar a generar información valiosa como por ejemplo al momento de querer conocer las razones por las cuales se puede ver afectado un cultivo. Es por esto, que este proyecto da un paso a la implementación de estos sistemas en pequeños cultivos y abre la puerta a que se generen nuevas soluciones.

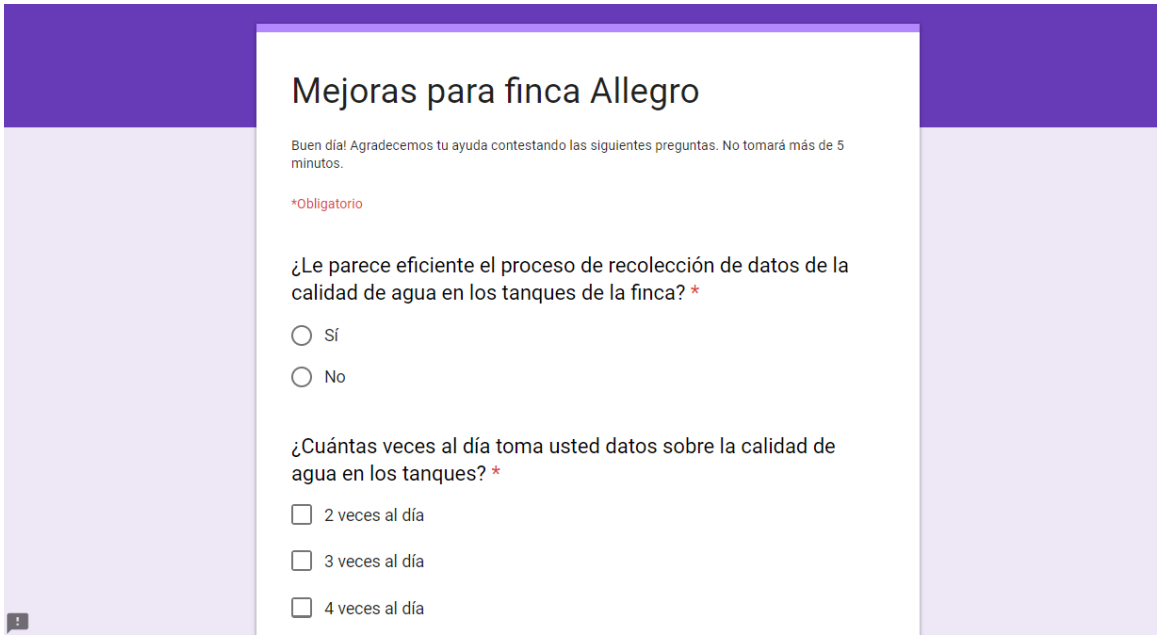
4.4 CONTRASTACIÓN

Si bien al tener la información de cada una de las variables que afectan al cultivo de tilapias, es necesario que la Fundación Finca Allegro mejore continuamente sus procedimientos para tareas correctivas cuando se presente una alerta en alguno de los tanques. Asimismo, es importante la capacitación constante de los operarios, pues son ellos los encargados de llevar las acciones correctivas y tener presente la importancia del uso de la tecnología en el desarrollo de sus labores.

ANEXOS

A. ENTREVISTAS

Formulario realizado en Google Docs.



Mejoras para finca Allegro

Buen día! Agradecemos tu ayuda contestando las siguientes preguntas. No tomará más de 5 minutos.

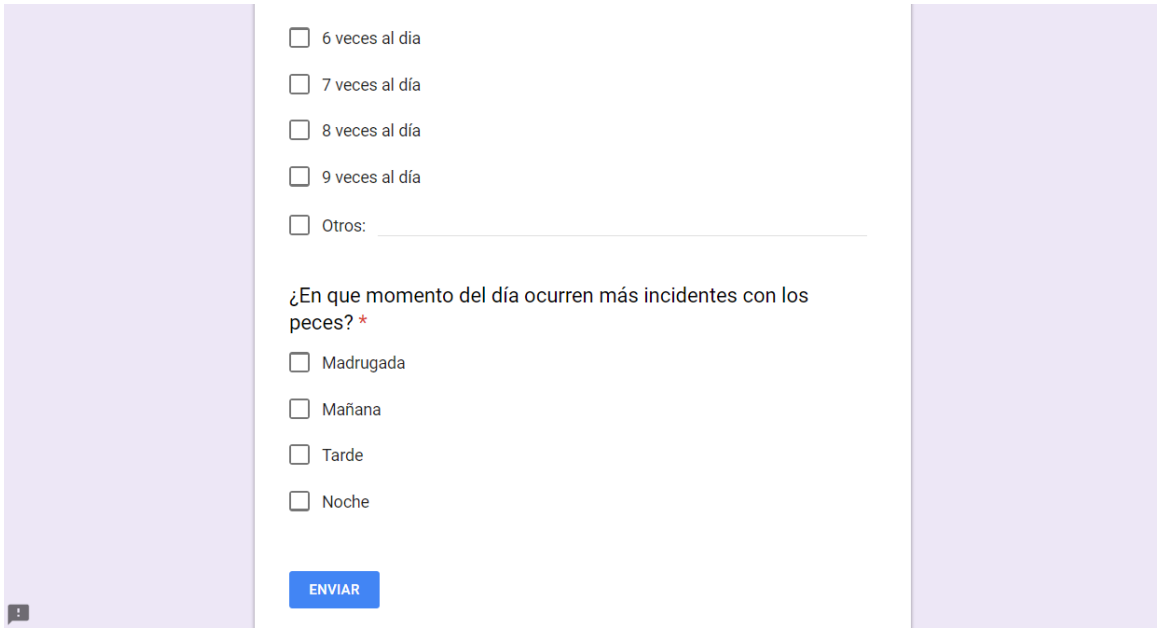
*Obligatorio

¿Le parece eficiente el proceso de recolección de datos de la calidad de agua en los tanques de la finca? *

- ○ Sí
- ○ No

¿Cuántas veces al día toma usted datos sobre la calidad de agua en los tanques? *

- ☐ 2 veces al día
- ☐ 3 veces al día
- ☐ 4 veces al día



- ☐ 6 veces al dia
- ☐ 7 veces al día
- ☐ 8 veces al día
- ☐ 9 veces al día
- ☐ Otros:

¿En que momento del día ocurren más incidentes con los peces? *

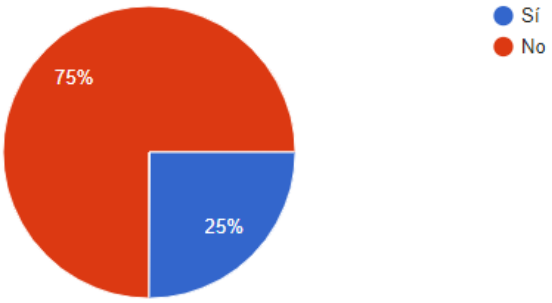
- ☐ Madrugada
- ☐ Mañana
- ☐ Tarde
- ☐ Noche

ENVIAR

Resultados:

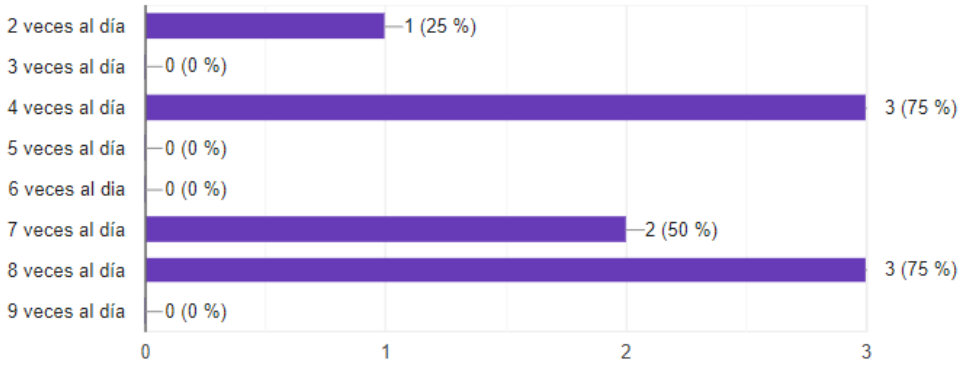
¿Le parece eficiente el proceso de recolección de datos de la calidad de agua en los tanques de la finca?

4 respuestas



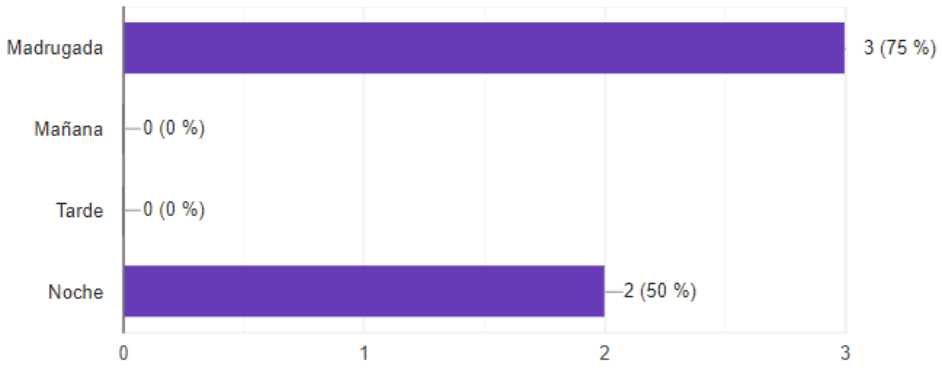
¿Cuántas veces al día toma usted datos sobre la calidad de agua en los tanques?

4 respuestas



¿En que momento del día ocurren más incidentes con los peces?

4 respuestas

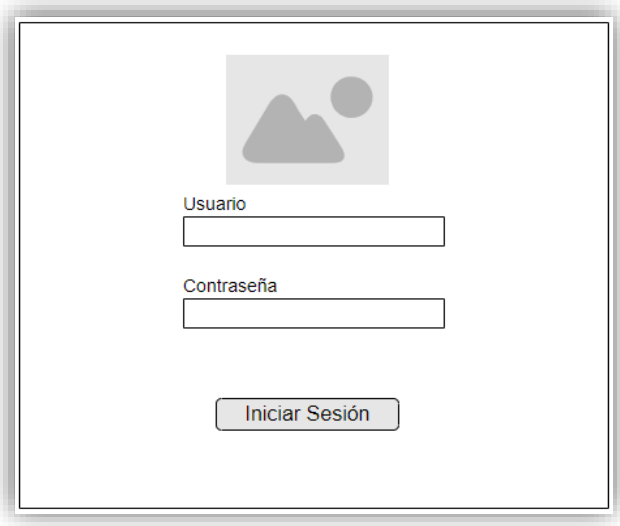


B. CASOS DE USO

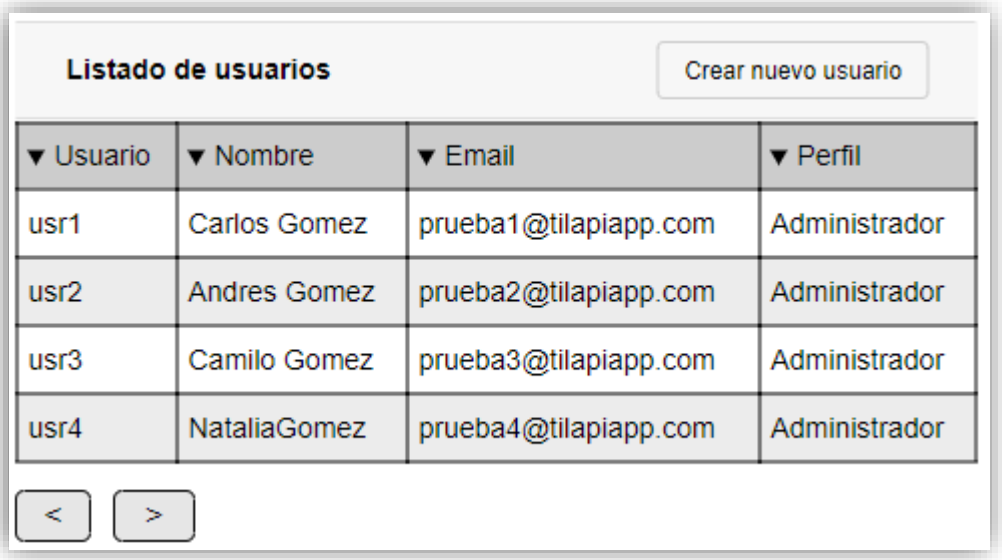
FORMATO DESCRIPCIÓN CASOS DE USO PARA ECS								
Información Básica		Aprobado:	SI	X	NO		Horas:	24
Proyecto:	**Tilapiapp Web**					ID Requisito:	1	
Nombre caso de uso:	Gestión de usuarios y acceso					ID Caso Uso:	1	
Solicitado por:	Finca Allegro		Revisado por:	Iván Mendoza				
Diligenciado por:	Nicolas Arias		Aprobado por:	Finca Allegro				
Detalle y Alcance								
1.	Descripción:	Se requiere realizar un módulo de gestión de usuarios para Tilapiapp web con el fin de gestionar el acceso a la aplicación.						
2.	Pre-condiciones:	Como precondiciones debe estar definida la estructura de la entidad usuarios en la base de datos de Tilapiapp.						
3.	Pos–condiciones:	El aplicativo tendrá la opción de consultar, crear y editar usuarios.						
4.	Restricciones:	• El aplicativo tendrá un usuario perfil administrador por defecto que se utilizará para crear nuevos usuarios. • Las vistas para creación y edición de usuarios deben ir asociadas a la entidad usuarios de la base de datos de Tilapiapp (Ver numeral 6). • Los perfiles de usuario deben ser “Administrador” y “Estándar”. • Los usuarios perfil administrador podrán crear, editar usuarios y parámetros de la aplicación. • Los usuarios perfil estándar podrán consular informes y ver el estado actual de la información de los tanques. • El acceso a Tilapiapp debe realizarse a través de una vista de login, mediante el usuario y la contraseña definida.						
5.	Referencias: *Especifique todos los archivos y casos de uso relacionados.*	Las referencias gráficas se describen en el numeral 6.						
6.	Especificaciones Gráficas (Si aplica)							

Descripción Gráfica.

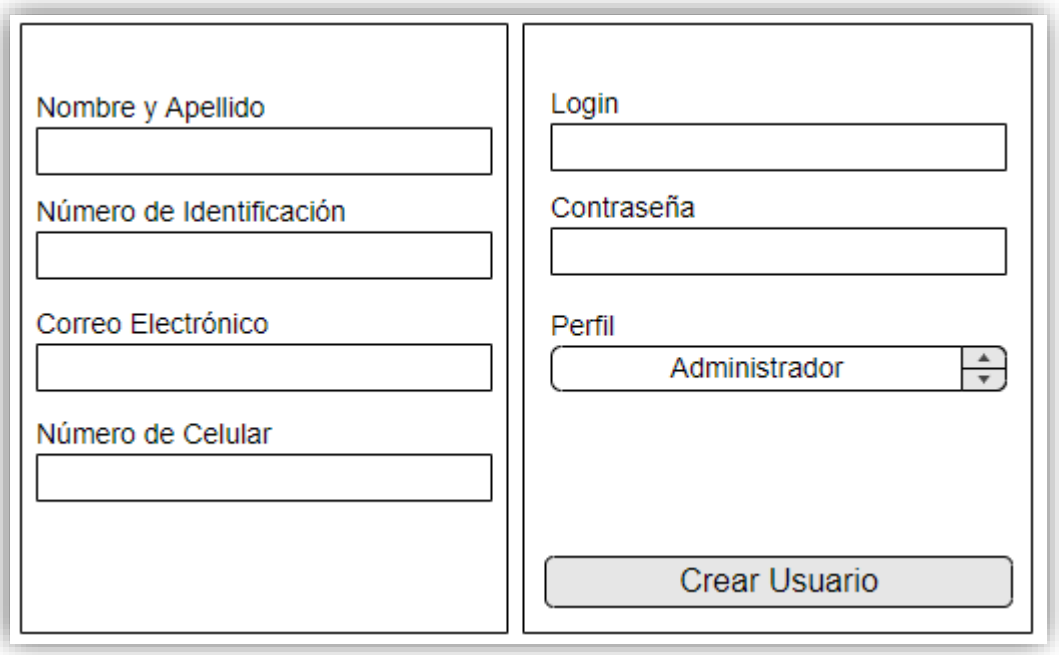
Inicio de sesión



Listado de usuarios



Creación y edición de usuario



7.	Flujo Normal y Flujos Alternos		
No.	Nombre del flujo	Acción	Respuesta Esperada
1	Normal	El usuario ingresa a Tilapiapp con las credenciales iniciales, ingresa a la sección de gestión de usuarios.	Se presenta el listado de usuarios indicando la información principal de cada registro.
2	Normal	El usuario presiona el botón Crear nuevo usuario. Se presenta el formulario y el usuario digita la información.	Se crea correctamente el usuario.
3	Alterno	El usuario presiona el botón Crear nuevo usuario. Se presenta el formulario y el usuario digita la información.	El usuario que intenta crear ya se encuentra registrado, se informa que debe revisar la información.
4	Normal	El usuario selecciona un registro del listado. Se presenta el formulario de edición y el usuario digita la información.	Se actualiza la información correctamente.
5	Alterno	El usuario selecciona un registro del listado. Se presenta el formulario de edición y el usuario digita la información.	La información no es correcta o está incompleta, se informa que debe revisar la información.
			Finaliza el flujo

FORMATO DESCRIPCIÓN CASOS DE USO PARA ECS								
Información Básica		Aprobado:	SI	X	NO		Horas:	32
Proyecto:	**Tilapiapp Web**					ID Requisito:	1	
Nombre caso de uso:	Gestión de parámetros y tanques					ID Caso Uso:	2	
Solicitado por:	Finca Allegro		Revisado por:	Iván Mendoza				
Diligenciado por:	Nicolas Arias		Aprobado por:	Finca Allegro				

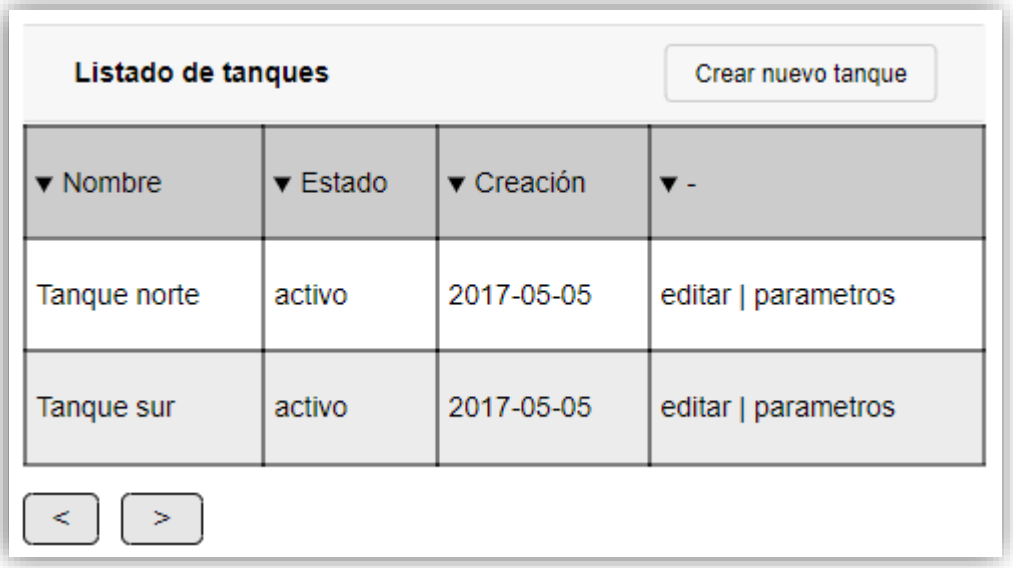
Detalle y Alcance		
1.	Descripción:	Se requiere realizar un módulo de gestión de parámetros para Tilapiapp web con el fin de crear y/o editar tanques y los umbrales asociados a las variables de medición.
2.	Pre-condiciones:	Como precondiciones debe estar definida la estructura de la entidad parámetros, tanques y variables de medición en la base de datos de Tilapiapp.
3.	Pos–condiciones:	El aplicativo tendrá la opción de consultar, crear y editar tanques y sus respectivos parámetros.

4.	Restricciones:	• Solo los usuarios perfil administrador tendrán acceso a la sección de parámetros de los tanques.
		• Las variables de medición para el aplicativo Tilapiapp serán 4.
		  ○ Temperatura.
		  ○ pH.
		  ○ Oxigeno.
		  ○ Saturación.
		• Las vistas para creación y edición de tanques y sus parámetros deben ir asociadas a la entidad Tanques y parámetros de la base de datos de Tilapiapp.
		• Los parámetros por variable de medición asociados a los tanques deben ser:
		  ○ Valor mínimo.
		  ○ Valor máximo.
		  ○ Mensaje alerta valor mínimo.
		  ○ Mensaje alerta valor máximo.
		• Los valores de los umbrales parametrizados servirán para procesar y generar alertas.
		• Los tanques deben tener un parámetro que indique el tiempo en segundos en el que recibirán las lecturas para su procesamiento.

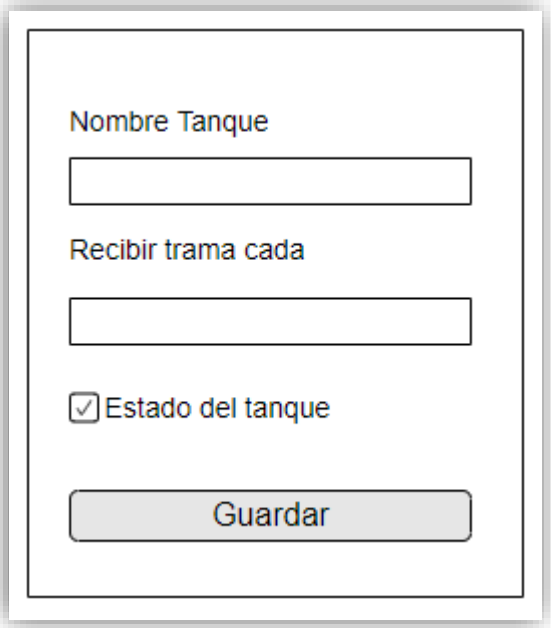
5.	Referencias: *Especifique todos los archivos y casos de uso relacionados.*	Las referencias gráficas se describen en el numeral 6.
6.	Especificaciones Gráficas (Si aplica)	

Descripción Gráfica.

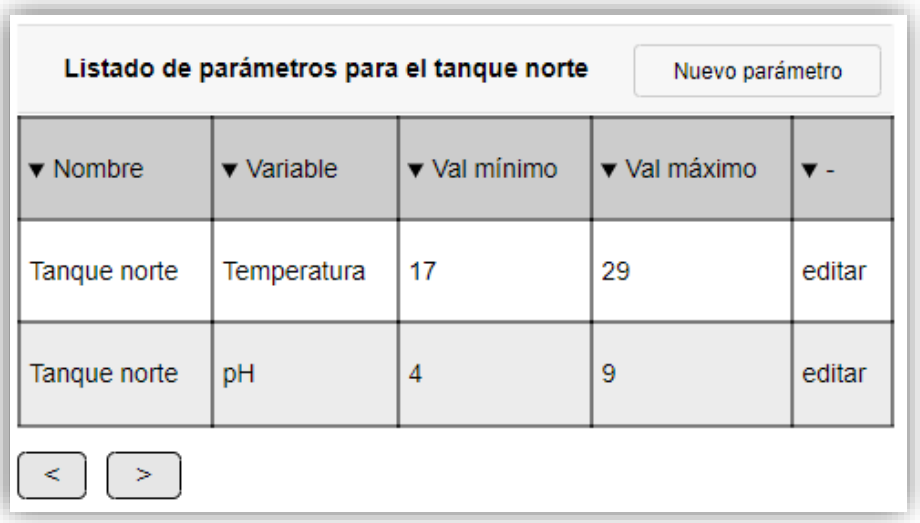
Listado de tanques.



Creación y edición de tanques



Listado de parametrización de tanques



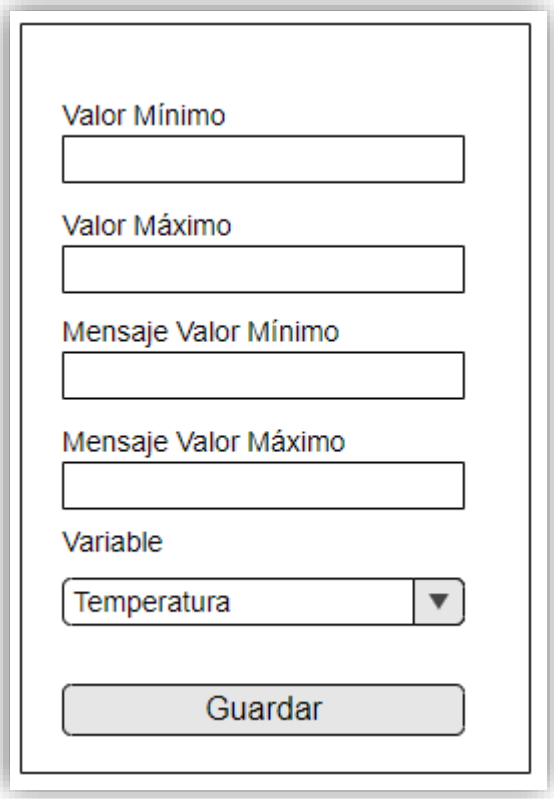
Listado de parámetros para el tanque norte

Nuevo parámetro

▼ Nombre	▼ Variable	▼ Val mínimo	▼ Val máximo	▼ -
Tanque norte	Temperatura	17	29	editar
Tanque norte	pH	4	9	editar

< >

Listado de parametrización de tanques



7.	Flujo Normal y Flujos Alternos		
No.	Nombre del flujo	Acción	Respuesta Esperada
1	Normal	El usuario ingresa a Tilapiapp con las credenciales iniciales, ingresa a la sección de tanques.	Se presenta el listado de tanques creados en el sistema.
2	Normal	El usuario presiona el botón Crear nuevo tanque. Se presenta el formulario y el usuario digita la información.	Se crea correctamente el tanque.
3	Alterno	El usuario presiona el botón Crear nuevo tanque. Se presenta el formulario y el usuario digita la información.	El tanque que intenta crear ya se encuentra registrado, se informa que debe revisar la información.
4	Normal	El usuario selecciona un registro del listado. Se presenta el formulario de edición y el usuario digita la información.	Se actualiza la información correctamente.
5	Alterno	El usuario selecciona un registro del listado. Se presenta el formulario de edición y el usuario digita la información.	La información no es correcta o está incompleta, se informa que debe revisar la información.

6	Normal	El usuario selecciona la opción parámetros de un registro del listado de tanques.	Se presenta el listado de parámetros asociados al tanque seleccionado.
7	Normal	El usuario presiona el botón nuevo parámetro. Se presenta el formulario y el usuario digita la información.	Se crea correctamente el parámetro.
8	Alterno	El usuario presiona el botón nuevo parámetro. Se presenta el formulario y el usuario digita la información.	El parámetro que intenta crear ya se encuentra registrado, se informa que debe revisar la información.
			Finaliza el flujo

FORMATO DESCRIPCIÓN CASOS DE USO PARA ECS								
Información Básica		Aprobado:	SI	X	NO		Horas:	24
Proyecto:	**Tilapiapp Web**					ID Requisito:	1	
Nombre caso de uso:	Servicios web Tilapiapp (Api)					ID Caso Uso:	3	
Solicitado por:	Finca Allegro		Revisado por:	Iván Mendoza				
Diligenciado por:	Nicolas Arias		Aprobado por:	Finca Allegro				
Detalle y Alcance								
1.	Descripción:	Se requiere realizar una API a través de servicios web RestFull que permita la integración con los sensores de cada tanque.						
2.	Pre-condiciones:	El dispositivo encargado de la lectura de los sensores (Raspberry) debe estar parametrizado y capturando las tramas de los sensores.						
3.	Pos–condiciones:	La API permitirá registrar los eventos e información proporcionada por el dispositivo de lectura.						
4.	Restricciones:	• La autenticación de los servicios web de Tilapiapp se realizará a través de JWT (Javascript Web Token) utilizando un usuario y contraseña que se encuentre parametrizado en Tilapiapp web. • La estructura de los métodos que tendrá la API se describe en el numeral 6.						
5.	Referencias: *Especifique todos los archivos y casos de uso relacionados.*	No aplica.						
6.	Especificaciones Gráficas (Si aplica)							

Métodos y estructura.

iniciarSesion

Realiza el proceso de autenticación en Tilapiapp Web.

Dirección	Parámetro	Tipo	Descripción
[In]	**userName**	**String**	Usuario en Tilapiapp.
[In]	**passWord**	**String**	Contraseña del usuario.
[Out]	**token**	**String**	JWT.
[Out]	**expiration**	**Datetime**	Fecha de expiración del Token.

obtenerParametros
Obtiene la parametrización de umbrales y lecturas para el tanque asociado al dispositivo Raspberry.

Dirección	Parámetro	Tipo	Descripción
[In]	**idTanque**	**Int**	Identificación del tanque.
[Out]	**Result**	**Boolean**	Resultado de la consulta.
[Out]	**Message**	**String**	Mensaje de la consulta.
[Out]	**Data**	**List<object>**	Listado de parámetros.

registrarTrama
Realizar el registro y procesamiento de la trama capturada por el dispositivo Raspberry.

Dirección	Parámetro	Tipo	Descripción
[In]	**Trama**	**String**	Trama capturada por el sensor.
[In]	**IdTanque**	**Int**	Identificación del tanque.
[Out]	**Result**	**Boolean**	Resultado de la consulta.
[Out]	**Message**	**String**	Mensaje de la consulta.

estadoSensorAsync
Almacena un registro cuando se presenta algún inconveniente con la lectura de la información de los sensores.

Dirección	Parámetro	Tipo	Descripción
[In]	**idTanque**	**Int**	Identificación del tanque.
[In]	**Estado**	**Int**	Estado del sensor.
[Out]	**Result**	**Boolean**	Resultado de la consulta.
[Out]	**Message**	**String**	Mensaje de la consulta.

estadoTarjetaAsync

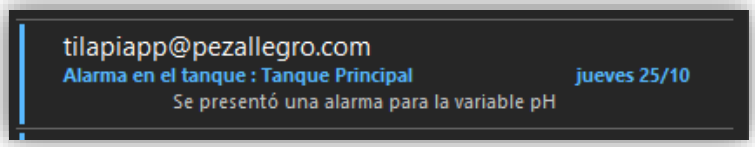
Almacena un registro cuando se presenta un cambio en el estado de la tarjeta (dispositivo Raspberry).

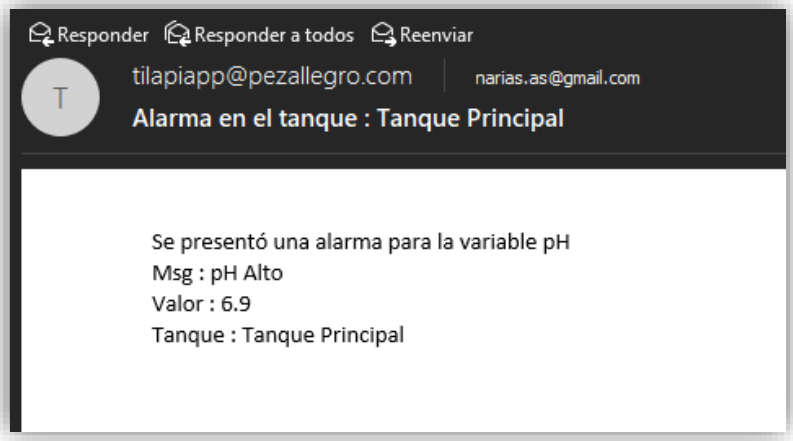
Dirección	Parámetro	Tipo	Descripción
[In]	**IdTanque**	**Int**	Identificador del tanque
[Out]	**Result**	**Boolean**	Resultado de la consulta.
[Out]	**Message**	**String**	Mensaje de la consulta.

7.	Flujo Normal y Flujos Alternos		
No.	Nombre del flujo	Acción	Respuesta Esperada
1	Normal	El dispositivo Raspberry realiza la conexión.	Se retorna el JWT para futuras consultas y peticiones.
2	Normal	El dispositivo Raspberry realiza el consumo de cualquiera de los métodos del API que requieren autenticación.	Se retorna la respuesta especifica de cada método indicando el resultado.
3	Alterno	El dispositivo Raspberry realiza el consumo de cualquiera de los métodos del API que requieren autenticación.	Si se presenta un fallo se almacena un Log interno y se retorna la respuesta.

FORMATO DESCRIPCIÓN CASOS DE USO PARA ECS								
Información Básica		Aprobado:	SI	X	NO		Horas:	8
Proyecto:	**Tilapiapp Web**					ID Requisito:	1	
Nombre caso de uso:	Alertas y procesamiento de trama					ID Caso Uso:	4	
Solicitado por:	Finca Allegro		Revisado por:	Iván Mendoza				
Diligenciado por:	Nicolas Arias		Aprobado por:	Finca Allegro				
Detalle y Alcance								
1.	Descripción:	Se requiere que el sistema Tilapiapp procese y genere alertas a través de correos electrónicos cuando los niveles capturados en la trama sobrepasan los umbrales parametrizados para las variables de los tanques.						
2.	Pre-condiciones:	Cada tanque debe tener parametrizados los umbrales para cada variable que captura el sensor.						
3.	Pos–condiciones:	Se generarán alertas cuando los umbrales se superen.						
4.	Restricciones:	• Las notificaciones se realizarán a todos los usuarios registrados en Tilapiapp. • Las notificaciones se enviarán a través de correos electrónicos. • La parametrización de la cuenta de correo que envía las notificaciones se realiza directamente en el aplicativo web.						
5.	Referencias: *Especifique todos los archivos y casos de uso relacionados.*	No aplica.						
6.	Especificaciones Gráficas (Si aplica)							

Ejemplo de notificación.





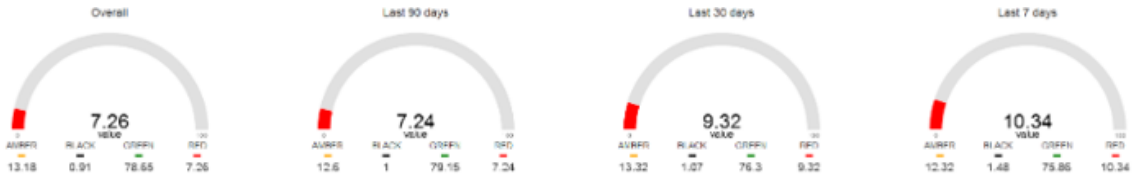
7.	Flujo Normal y Flujos Alternos		
No.	Nombre del flujo	Acción	Respuesta Esperada
1	Normal	Tilapiapp recibe la trama capturada por el dispositivo Raspberry y realiza el procesamiento.	Se almacena la trama obtenida en la base de datos, los niveles no sobrepasan los umbrales no se envía notificación.
2	Alterno	Tilapiapp recibe la trama capturada por el dispositivo Raspberry y realiza el procesamiento.	Se almacena la trama obtenida en la base de datos, los niveles sobrepasan los umbrales se envía notificación a cada usuario registrado en Tilapiapp.

FORMATO DESCRIPCIÓN CASOS DE USO PARA ECS								
Información Básica		Aprobado:	SI	X	NO		Horas:	36
Proyecto:	**Tilapiapp Web**					ID Requisito:	1	
Nombre caso de uso:	Dashboard e Histórico					ID Caso Uso:	5	
Solicitado por:	Finca Allegro		Revisado por:	Iván Mendoza				
Diligenciado por:	Nicolas Arias		Aprobado por:	Finca Allegro				

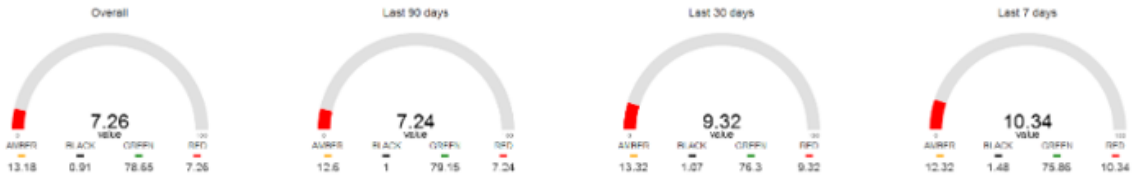
Detalle y Alcance		
1.	Descripción:	Se requiere visualizar la información actual sobre las lecturas obtenidas por el dispositivo de lectura Raspberry a través de un Dashboard en Tilapiapp web.
2.	Pre-condiciones:	El dispositivo de lectura debe consumir el Web service y entregar la información obtenida de los sensores.
3.	Pos–condiciones:	El usuario que se encuentre con una sesión activa en Tilapiapp podrá consultar la información actual y visualizar los históricos de la información recibida.
4.	Restricciones:	• La información se actualizará cada 5 segundos automáticamente. • En el Dashboard se presentará una fila por cada tanque activo en el sistema. • Los históricos se podrán generar por fecha, tanque y variable de medición.
5.	Referencias: *Especifique todos los archivos y casos de uso relacionados.*	No aplica.
6.	Especificaciones Gráficas (Si aplica)	

Representación Gráfica del Dashboard

Tanque Principal



Tanque Secundario



Representación Gráfica del Histórico



7.	Flujo Normal y Flujos Alternos		
No.	Nombre del flujo	Acción	Respuesta Esperada
1	Normal	El usuario ingresa a Tilapiapp, por defecto se direcciona al Dashboard.	El usuario visualiza la información actual del sistema y los tanques.
2	Alterno	El usuario ingresa a Tilapiapp, por defecto se direcciona al Dashboard.	No se encuentran registros de tanques, se informa al usuario.
3	Normal	El usuario ingresa a los históricos, selecciona los filtros y busca la información.	Se obtienen los resultados y se grafica la información obtenida.
4	Alterno	El usuario ingresa a los históricos, selecciona los filtros y busca la información.	No se encuentran registros de históricos para los filtros seleccionados, se informa al usuario.
5			Finaliza el flujo

FORMATO DESCRIPCIÓN CASOS DE USO PARA ECS								
Información Básica		Aprobado:	SI	X	NO		Horas:	8
Proyecto:	**Tilapiapp Sensor**					ID Requisito:	1	
Nombre caso de uso:	Obtener token para uso servicios REST Tilapiapp Web					ID Caso Uso:	1	
Solicitado por:	Finca Allegro		Revisado por:	Nicolas Arias				
Diligenciado por:	Iván Mendoza		Aprobado por:	Finca Allegro				

Detalle y Alcance		
1.	Descripción:	Se requiere que el API del sensor Raspberry se autentique con el servicio Rest de la aplicación Tilapia Web.
2.	Pre-condiciones:	Se debe implementar y exponer el servicio REST de la aplicación Tilapiapp Web
3.	Pos–condiciones:	El servicio REST devuelve la clave privada para autenticación a los servicios
4.	Restricciones:	• El usuario y contraseña deben estar registrados en la base de datos de Tilapiapp Web. • El envío de los datos debe realizarse a través del formato JSON, tanto para recepción como para envío de datos. • El token debe actualizarse cada hora.
5.	Referencias: *Especifique todos los archivos y casos de uso relacionados.*	Para este caso no es necesario el uso de referencias gráficas.
6.	Especificaciones Gráficas (Si aplica)	

Estructuras de datos

La información enviada al servicio debe contar con la siguiente estructura:

Dirección	Parámetro	Tipo	Descripción
[Out]	**userName**	**String**	Usuario en Tilapiapp.
[Out]	**passWord**	**String**	Contraseña del usuario.
[In]	**token**	**String**	JWT.
[In]	**expiration**	**Datetime**	Fecha de expiración del Token.

7.	Flujo Normal y Flujos Alternos		
No.	Nombre del flujo	Acción	Respuesta Esperada
1	Normal	La aplicación se ejecuta con los parámetros de identificación del tanque, usuario y contraseña.	Inicio de la aplicación en el sensor.
2	Normal	La aplicación consume el servicio de autenticación enviado el usuario y contraseña.	Si el usuario y contraseña son correctos se recibe el token generado por la aplicación y la fecha de expiración de este. Se almacena en el sensor.
3	Alterno		Si el usuario o la contraseña no son correctos, se recibe error de

			autenticación y se almacena en log. La aplicación no inicia.
4	Alterno	Pasada una hora, la aplicación solicitará nuevamente un token, enviado usuario y contraseña	Retorno del token y almacenamiento del mismo en el sensor.
			Finaliza el flujo

FORMATO DESCRIPCIÓN CASOS DE USO PARA ECS								
Información Básica		Aprobado:	SI	X	NO		Horas:	12
Proyecto:	**Tilapiapp Sensor**					ID Requisito:	1	
Nombre caso de uso:	Obtener parámetros del tanque					ID Caso Uso:	2	
Solicitado por:	Finca Allegro		Revisado por:	Nicolas Arias				
Diligenciado por:	Iván Mendoza		Aprobado por:	Finca Allegro				

Detalle y Alcance		
1.	Descripción:	Se requiere que el API del sensor Raspberry obtenga los parámetros de envío de información y valores máximos y mínimos de las variables de medición
2.	Pre-condiciones:	Se debe implementar y exponer el servicio REST de la aplicación Tilapiapp Web.
3.	Pos–condiciones:	El servicio REST retorna la parametrización para el tanque solicitado.
4.	Restricciones:	• El servicio API debe haber realizado el login previamente, para obtener el token de autenticación. • El retorno de los datos debe ser en formato JSON. • La actualización de parámetros se realiza cada hora
5.	Referencias: *Especifique todos los archivos y casos de uso relacionados.*	Para este caso no es necesario el uso de referencias gráficas.
6.	Especificaciones Gráficas (Si aplica)	

Estructuras de datos

La información enviada al servicio debe contar con la siguiente estructura:

Dirección	Parámetro	Tipo	Descripción
[Out]	**idTanque**	**Int**	Identificación del tanque.
[In]	**Result**	**Boolean**	Resultado de la consulta.
[In]	**Message**	**String**	Mensaje de la consulta.
[In]	**Data**	**List<object>**	Listado de parámetros.

7.	Flujo Normal y Flujos Alternos		
No.	Nombre del flujo	Acción	Respuesta Esperada
1	Normal		Aplicación en ejecución.
2	Normal	La aplicación consume el servicio de actualización de parámetros enviando el identificados del tanque.	Se recibe objeto JSON con el listado de parámetros por variable establecidos por el usuario a través del sitio Tilapiapp Web

3	Alterno		Si el token no es válido, se genera error de autenticación y se guarda la excepción en archivo log y en base de datos del sensor
4	Alterno		Si el identificador del tanque no se encuentra parametrizado, se genera excepción y se guarda en archivo log y en base de datos.
5	Normal	Se almacenan los parámetros en la base de datos del sensor. Si ya existen parámetros, estos se inactivan. No se sobrescriben datos.	Actualización de los parámetros en la base de datos.
6	Alterno		Si se presenta error en el almacenamiento, se almacena en archivo log y en la base de datos.
			Finaliza el flujo

FORMATO DESCRIPCIÓN CASOS DE USO PARA ECS								
Información Básica		Aprobado:	SI	X	NO		Horas:	24
Proyecto:	**Tilapiapp Sensor**					ID Requisito:	1	
Nombre caso de uso:	Leer trama del sensor serial					ID Caso Uso:	3	
Solicitado por:	Finca Allegro		Revisado por:	Nicolas Arias				
Diligenciado por:	Iván Mendoza		Aprobado por:	Finca Allegro				
Detalle y Alcance								
1.	Descripción:	Se requiere que el API del Raspberry obtenga la trama del sensor del tanque por medio de protocolo serial.						
2.	Pre-condiciones:	Se debe tener una interfaz serial. Para este caso la Finca Allegro la proporciona.						
3.	Pos–condiciones:	La trama se almacena en la base de datos						
4.	Restricciones:	• El intervalo de recepción de la trama depende de la programación del sensor por parte de la Finca Allegro. • La escucha por el puerto serial de la trama se hace en bucle, hasta que la aplicación sea detenida.						
5.	Referencias: *Especifique todos los archivos y casos de uso relacionados.*	Para este caso no es necesario el uso de referencias gráficas.						
6.	Especificaciones Gráficas (Si aplica)							

Estructuras de datos

La trama recibida por protocolo serial tiene la siguiente estructura

>Tx,hh:mm,Temp,Oxi,Sat,Ph,Vol,CodAl,CodErr

Parámetro	Descripción
>Tx	Identificador tanque donde x es número
hh:mm	Hora y minutos en formato 24 horas
Temp	Variable de temperatura
Oxi	Variable de oxígeno
Sat	Variable de saturación de oxígeno
Ph	Variable de pH

Vol	Variable de voltaje de batería del sensor
CodAl	Código alarma
CodErr	Código Error

7.	Flujo Normal y Flujos Alternos		
No.	Nombre del flujo	Acción	Respuesta Esperada
1	Normal		Aplicación en ejecución.
2	Normal	La aplicación abre el puerto serial de escucha de la Raspberry	Puerto puesto en escucha
3	Alterno		Si se genera error, se almacena en la base de datos y en archivo log
4	Normal		De acuerdo con el tiempo t parametrizado por la Finca Allegro del sensor en el tanque, se recibe la trama con la estructura del punto 6.
5	Normal	Una vez recibida la trama, se almacena en la base de datos. (No se debe procesar)	Datos almacenados
6	Alterno		Si se presenta error en el almacenamiento, se almacena en archivo log y en la base de datos.
			Finaliza el flujo

FORMATO DESCRIPCIÓN CASOS DE USO PARA ECS								
Información Básica		Aprobado:	SI	X	NO		Horas:	12
Proyecto:	**Tilapiapp Sensor**					ID Requisito:	1	
Nombre caso de uso:	Enviar trama y actualizar estado del sensor.					ID Caso Uso:	4	
Solicitado por:	Finca Allegro		Revisado por:	Nicolas Arias				
Diligenciado por:	Iván Mendoza		Aprobado por:	Finca Allegro				
Detalle y Alcance								
1.	Descripción:	Se requiere que el API del sensor Raspberry envíe la trama a la aplicación Tilapiapp Web para que sea procesada y analizada. Asimismo, se hace la actualización del estado del sensor						
2.	Pre-condiciones:	Se debe contar con información de trama en la base de datos.						
3.	Pos–condiciones:	Trama enviada y procesada por Tilapiapp Web						
4.	Restricciones:	• El servicio API debe haber realizado el login previamente, para obtener el token de autenticación. • El formato de transmisión de datos es JSON. • El tiempo de envío de la trama esta parametrizado por el usuario en la aplicación Tilapiapp Web						
5.	Referencias: *Especifique todos los archivos y casos de uso relacionados.*	Para este caso no es necesario el uso de referencias gráficas.						
6.	Especificaciones Gráficas (Si aplica)							

Estructuras de datos

La información enviada con la trama al servicio debe contar con la siguiente estructura:

Dirección	Parámetro	Tipo	Descripción
[Out]	**Trama**	**String**	Trama capturada por el sensor.
[Out]	**IdTanque**	**Int**	Identificación del tanque.
[In]	**Result**	**Boolean**	Resultado de la consulta.
[In]	**Message**	**String**	Mensaje de la consulta.

La información enviada con el estado del sensor al servicio debe contar con la siguiente estructura:

Dirección	Parámetro	Tipo	Descripción
[Out]	**idTanque**	**Int**	Identificación del tanque.
[Out]	**Estado**	**Int**	Estado del sensor.
[In]	**Result**	**Boolean**	Resultado de la consulta.
[In]	**Message**	**String**	Mensaje de la consulta.

7.	Flujo Normal y Flujos Alternos		
No.	Nombre del flujo	Acción	Respuesta Esperada
1	Normal		Aplicación en ejecución.
2	Normal	La aplicación en un intervalo configurado por la parametrización del tanque remite la última trama almacenada en la base de datos y el identificador del tanque.	Se consume el servicio de envío de trama de Tilapiapp Web
3	Alterno		Si el token no es válido, se genera error de autenticación y se guarda la excepción en archivo log y en base de datos del sensor
4	Alterno		Si el identificador del tanque no se encuentra parametrizado, se genera excepción y se guarda en archivo log y en base de datos.
5	Alterno	Si la última trama almacenada ya había sido enviada se informa que el sensor puede tener problemas de lectura. Se consume el servicio de Estado Tarjeta de aplicación Tilapiapp Web, enviando el estado en 0 para el tanque indicado.	Información del sensor actualizada en el sitio Web.
6	Normal	Se actualiza el estado del sensor del tanque, enviando el estado en 1 para el tanque indicado.	Información del sensor actualizada en el sitio Web.
6	Alterno		Si se presenta error en el almacenamiento, se almacena en archivo log y en la base de datos.
			Finaliza el flujo

FORMATO DESCRIPCIÓN CASOS DE USO PARA ECS								
Información Básica		Aprobado:	SI	X	NO		Horas:	6
Proyecto:	**Tilapiapp Sensor**					ID Requisito:	1	
Nombre caso de uso:	Enviar estado de la tarjeta (Raspberry)					ID Caso Uso:	5	
Solicitado por:	Finca Allegro		Revisado por:	Nicolas Arias				
Diligenciado por:	Iván Mendoza		Aprobado por:	Finca Allegro				

Detalle y Alcance		
1.	Descripción:	Se debe estar notificando cada 10 segundo el estado de la tarjeta (Raspberry)
2.	Pre-condiciones:	Se debe implementar y exponer el servicio REST de la aplicación Tilapiapp Web.
3.	Pos–condiciones:	El servicio REST actualiza el estado de la tarjeta.
4.	Restricciones:	• El servicio API debe haber realizado el login previamente, para obtener el token de autenticación. • La actualización de parámetros se realiza cada 10 segundos
5.	Referencias: *Especifique todos los archivos y casos de uso relacionados.*	Para este caso no es necesario el uso de referencias gráficas.
6.	Especificaciones Gráficas (Si aplica)	

Estructuras de datos

La información enviada al servicio debe contar con la siguiente estructura:

Dirección	Parámetro	Tipo	Descripción
[Out]	**Trama**	**String**	Trama capturada por el sensor.
[In]	**Result**	**Boolean**	Resultado de la consulta.
[In]	**Message**	**String**	Mensaje de la consulta.

7.	Flujo Normal y Flujos Alternos		
No.	Nombre del flujo	Acción	Respuesta Esperada
1	Normal		Aplicación en ejecución.
2	Normal	La aplicación consume el servicio de actualización del estado de la tarjeta enviado el identificador del tanque.	Se recibe estado de confirmación de la actualización.
3	Alterno		Si el token no es válido, se genera error de autenticación y se guarda la excepción en archivo log y en base de datos del sensor

4	Alterno		Si el identificador del tanque no se encuentra parametrizado, se genera excepción y se guarda en archivo log y en base de datos.
			Finaliza el flujo

C. CARTA AUTORIZACIÓN USO SENSONRES

Noviembre 27, 2018

San Francisco, Cundinamarca

Señores

Nicolas Arias, Iván Mendoza,

Asunto: Autorización

Con la presente autorizó la utilización de los sensores y tarjeta electrónica desarrollada en la finca Allegro. Esta autorización va dirigida al proyecto de investigación **"Diseño de un prototipo de sistema web para el monitoreo y control de la calidad del agua en los estanques de cultivo de tilapias en la finca Allegro en San Francisco Cundinamarca."** Desarrollado en la especialización de ingeniería de software en la universidad distrital Francisco José de Caldas.



Alonso Lewing

Dueño de la fina Allegro

CEL: 321 207 4860

REFERENCIAS

[1] DANE, «Insumos y factores asociados a la producción agropecuaria,» DANE, Bogotá, 2014.

[2] M. A. Saavedra Martinez, «Manejo del cultivo de tilapias,» Universidad Centroamericana, Managua, 2006.

[3] R. A. Española, «RAE,» 15 04 2018. [En línea]. Available: http://dle.rae.es/?id=YErlG2H.

[4] R. Pressman S., Igeniería de Sofrware. Un enfoque práctico, 7ma ed., New York: McGraw Hill, 2010, p. 805.

[5] AseSoftware, «AseSoftware,» 10 Mayo 2018. [En línea]. Available: http://asesoftware.com/site/blog/2016/03/08/la-arquitectura-microservicios/.

[6] Microsoft, «Microsot Azure,» 10 Mayo 2018. [En línea]. Available: https://docs.microsoft.com/es-es/azure/architecture/guide/architecture-styles/microservices.

[7] M. Morris Mano, Arquitectura de Computadoras, 3ra ed., Mexico: Pearson Education, 1994.

[8] L. Da Xu, W. He y S. Li, «Internet of things in industries: A survey,» *IEEE Transactions on industrial informatics,* p. 10, 2013.

[9] Milacron Mexicana Sales S.A., «¿Por qué es importante el Oxígeno Disuelto?,» CIMCOOL, Querétaro, Mexico, 2004.

[10] W. C. Raymond Chan, Quimica, Séptima Edición ed., McGraw-Hill, Ed., Mexico D.F.: McGraw-Hill, 2002, p. 1004.

[11] H. d. Peces, «Hablemos de Peces,» 12 04 2018. [En línea]. Available: http://hablemosdepeces.com/pez-tilapia/.

[12] T. O. Group, «Open Group,» [En línea]. Available: http://pubs.opengroup.org/architecture/archimate3-doc/apdxd.html. [Último acceso: 09 10 2018].

[13] C. d. l. t. llorente, Guía de arquitectura N-Capas orientada al domino con .net 4.0, España: Krasis Consulting, 2010.

[14] M. C. M. Archilla, «Guía Práctica de Piscicultura en Colombia,» *INCODER,* 2006.

[15] J. J. L. Pérez, Modelo seudomatemático para el diseño de las bases de datos relacionales, Bogotá: MAGA Imagen Estratégica, 2016.

[16] L. Welicki, «The Dynamic Factory Pattern,» 2008.

[17] Node.js, «Node.js,» [En línea]. Available: https://nodejs.org/en/about/. [Último acceso: 20 10 2018].

[18] «PuTTY.org,» PuTTY Project, [En línea]. Available: https://www.putty.org. [Último acceso: 15 10 2018].

[19] O. Corp., «MySQL,» Oracle, [En línea]. Available: https://www.mysql.com/products/workbench/. [Último acceso: 20 10 2018].

[20] IETF, «IETF,» [En línea]. Available: https://www.ietf.org/rfc/rfc5724.txt. [Último acceso: 01 11 2018].

[21] American Psychological Association, Manual de Publicaciones de la American Psychological Association, 6 ed., México: El Manual Moderno, 2010.

[22] Opensource.com, «Opensource.com,» 12 04 2018. [En línea]. Available: https://opensource.com/resources/raspberry-pi.