


| | | | | |
|---|---|---------------------|-------------------|----------|
|  Universidad Francisco de Paula Santander Ocaña - Colombia Vigesima Reimpresión | UNIVERSIDAD FRANCISCO DE PAULA SANTANDER OCAÑA | | | |
| | Documento | Código | Fecha | Revisión |
| | FORMATO HOJA DE RESUMEN PARA TRABAJO DE GRADO | F-AC-DBL-007 | 10-04-2012 | A |
| Dependencia | Aprobado | Pág. | | |
| DIVISIÓN DE BIBLIOTECA | SUBDIRECTOR ACADEMICO | i(137) | | |

RESUMEN – TRABAJO DE GRADO

| | | | |
|--|--|----------------|-----------|
| AUTORES | BRAYAN SANCHEZ TORRES | | |
| FACULTAD | INGENIERÍA | | |
| PLAN DE ESTUDIOS | INGENIERÍA DE SISTEMAS | | |
| DIRECTOR | ALVEIRO ALONSO ROSADO GOMEZ | | |
| TÍTULO DE LA TESIS | DESARROLLO DE UN PROTOTIPO PARA MONITOREAR LA TEMPERATURA Y EL PH EN LA ESTACIÓN PISCÍCOLA DE LA UNIVERSIDAD FRANCISCO DE PAULA SANTANDER OCAÑA | | |
| RESUMEN | | | |
| (70 palabras aproximadamente) | | | |
| <p>EN EL PRESENTE TRABAJO SE DESCRIBE EL PROCEDIMIENTO Y RESULTADO DEL DESARROLLO DE UN SISTEMA PARA MONITOREAR LA TEMPERATURA Y PH DE ESTANQUES DE PECES, CON CARACTERÍSTICAS DE UN SISTEMA DE INTERNET DE LAS COSAS, QUE PUEDA NOTIFICAR AL USUARIO SOBRE LAS INCIDENCIAS QUE SE PRESENTEN EN ESTAS, CON RESPECTO A LAS VARIABLES PLANTEADAS EN EL PROYECTO PARA ANALIZAR.</p> | | | |
| CARACTERÍSTICAS | | | |
| PÁGINAS: 137 | PLANOS: | ILUSTRACIONES: | CD-ROM: 1 |

**DESARROLLO DE UN PROTOTIPO PARA MONITOREAR LA TEMPERATURA Y
EL PH EN LA ESTACIÓN PISCÍCOLA DE LA UNIVERSIDAD FRANCISCO DE
PAULA SANTANDER OCAÑA**

Autores

BRAYAN SANCHEZ TORRES – 191018

Trabajo de grado para presentado para optar por el título de ingeniero de sistemas

Director

ALVEIRO ALONSO ROSADO GOMEZ

© Magister en gestión, aplicación y desarrollo de software

UNIVERSIDAD FRANCISCO DE PAULA SANTANDER OCAÑA

FACULTAD DE INGENIERIAS

INGENIERIAS DE SISTEMAS

Ocaña – Norte de Santander, Colombia

Noviembre, 2018

Índice

| | |
|--|----|
| Capítulo 1. Desarrollo de un prototipo para monitorear la temperatura y el pH en la estación piscícola de la Universidad Francisco de Paula Santander Ocaña.....1 | |
| 1.1 Planteamiento del problema..... | 1 |
| 1.1.1 Formulación del problema | 3 |
| 1.2 Objetivos | 3 |
| 1.2.1 Objetivo general..... | 3 |
| 1.2.2 Objetivos específicos | 3 |
| 1.3 Justificación | 3 |
| 1.4 Delimitaciones | 6 |
| 1.4.1 Geográfica..... | 6 |
| 1.4.2 Operativa..... | 7 |
| 1.4.3 Conceptual | 7 |
| 1.4.4 Temporal..... | 7 |
| Capítulo 2. Marco Referencial.....8 | |
| 2.1. Marco histórico | 8 |
| 2.1.1 Antecedentes a nivel internacional..... | 9 |
| 2.1.2 Antecedentes a nivel nacional..... | 9 |
| 2.1.3 Antecedentes a nivel local..... | 10 |
| 2.2 Marco contextual | 10 |
| 2.3 Marco conceptual..... | 11 |
| 2.4 Marco teórico | 15 |
| 2.5 Marco legal | 18 |
| Capítulo 3. Diseño Metodológico.....20 | |
| 3.1 Tipo de investigación..... | 20 |
| 3.2 Población y muestra..... | 20 |
| 3.2.1 Población..... | 20 |
| 3.2.2 Muestra | 21 |
| 3.3 Técnicas e instrumentos de la recolección de la información..... | 21 |
| 3.3.1 Cronograma de actividades..... | 22 |

| | |
|---|-----------|
| Capítulo 4. Administración del proyecto..... | 23 |
| 4.1 Administración del proyecto..... | 23 |
| 4.4.1 Recursos humanos..... | 23 |
| 4.4.2 Recursos financieros..... | 23 |
| 4.4.3 Recursos institucionales..... | 24 |
| | |
| Capítulo 5. Resultados..... | 25 |
| 5.1 Requerimientos | 25 |
| 5.1.1 Requisitos funcionales y no funcionales..... | 26 |
| 5.1.1.1 Requisitos funcionales del sistema..... | 27 |
| 5.1.1.2 Requisitos no funcionales del sistema..... | 27 |
| 5.2 Análisis | 28 |
| 5.2.1 Tecnologías Disponibles en Hardware..... | 29 |
| 5.2.1.1 Procesamiento de datos..... | 30 |
| 5.2.1.1.1 Plataformas seleccionadas..... | 33 |
| 5.2.1.2 Sensores | 34 |
| 5.2.1.3 Comunicaciones..... | 37 |
| 5.2.1.3.1 Modulo seleccionado | 43 |
| 5.2.3 Tecnologías Disponibles en software..... | 43 |
| 5.2.3.1 Software para programación en hardware..... | 43 |
| 5.2.3.2 Software para programación de aplicaciones Web..... | 45 |
| 5.3 Diseño | 46 |
| 5.3.1 Diagrama de Arquitectura..... | 47 |
| 5.3.2 Diagrama y descripción de casos de uso..... | 49 |
| 5.3.3 Diagrama de clases..... | 59 |
| 5.3.4 Diagrama de secuencias..... | 60 |
| 5.3.5 Modelo entidad relación..... | 64 |
| 5.3.6 Controles..... | 65 |
| 5.3.7 Diseño de interfaces..... | 66 |
| 5.4 Desarrollo..... | 67 |
| 5.4.1. Desarrollo del aplicativo web | 67 |
| 5.4.1.1 Funcionamiento general..... | 67 |
| 5.4.1.2 Base de datos..... | 68 |
| 5.4.1.3 Arquitectura de la aplicación | 69 |
| 5.4.1.4 Diagrama de paquetes..... | 70 |

| | |
|--|-----------|
| 5.4.1.5 Interfaz gráfica diseño..... | 72 |
| 5.4.2 Desarrollo de los dispositivos físicos..... | 74 |
| 5.4.2.1 Programación y ensamble de los dispositivos..... | 74 |
| 5.4.3 Correo electrónico..... | 78 |
| 5.4.4 Configuración de Pushetta | 78 |
| 5.4.5 Socket..... | 80 |
| 5.5 Pruebas..... | 80 |
| 5.5.1. Configuración y acondicionamiento del vps..... | 83 |
| 5.5.2 Puesta en marcha del socket..... | 84 |
| 5.5.3. Inicialización de los dispositivos | 84 |
| 5.5.4 Frecuencia de toma de datos | 84 |
| 5.6 Despliegue..... | 87 |
| Capítulo 6. Conclusiones..... | 88 |
| 6.1. Conclusiones generales..... | 88 |
| 6.2. Conclusiones por objetivos..... | 88 |
| Capítulo 7. Recomendaciones..... | 89 |
| Referencias..... | 90 |
| Apéndices..... | 94 |
| Apéndice A Configuración de Raspberry | 94 |
| Apéndice B Códigos programados del proyecto..... | 94 |
| Apéndice C Diseños de interfaces | 95 |
| Apéndice D Puesta en marcha del socket | 112 |
| Apéndice E Imágenes de la aplicación en ejecución | 113 |

Lista de tablas

| | |
|---|----|
| Tabla 1. Campos de justificación..... | 6 |
| Tabla 2. Presupuesto global del proyecto | 23 |
| Tabla 3. Características del electrodo de pH..... | 36 |
| Tabla 4. Descripción pines modulo NFRI2401 | 41 |
| Tabla 5. Tabla de controles del proyecto para el administrador y los operarios | 65 |

Lista de figuras

| | |
|---|----|
| Figura 1. Sonda de temperatura DSB18b20 | 35 |
| Figura 2. Sonda de pH profesional | 37 |
| Figura 3. Módulo de radiofrecuencia NFRL2401 | 40 |
| Figura 4. Diagrama de referencia ETSI M2M | 47 |
| Figura 5. Diagrama de bloques del sistema interno | 48 |
| Figura 6. Diagrama de arquitectura | 49 |
| Figura 7. Diagrama de casos de usos | 50 |
| Figura 8. Caso de uso visualizar temperatura y pH | 51 |
| Figura 9. Caso de uso autenticarse ante el sistema | 52 |
| Figura 10. Caso de uso actualizar información | 53 |
| Figura 11. Caso de uso visualizar datos estadísticos | 54 |
| Figura 12. Caso de uso gestionar información | 55 |
| Figura 13. Caso de uso gestionar eventos y soluciones | 56 |
| Figura 14. Caso de uso crear usuario | 57 |
| Figura 15. Caso de uso actualizar usuario | 58 |
| Figura 16. Diagrama de clases | 59 |
| Figura 17. Diagrama de secuencias | 60 |
| Figura 18. Diagrama de secuencia gestionar información | 61 |
| Figura 19. Diagrama de secuencia gestionar soluciones y eventos | 62 |
| Figura 20. Diagrama de secuencia crear usuario y actualizar usuario | 63 |
| Figura 21. Modelo entidad relación | 64 |
| Figura 22. Árbol de navegación de interfaces | 66 |
| Figura 23. Configuración de hibernate | 68 |
| Figura 24. Diagrama de paquetes jerárquico del proyecto | 71 |
| Figura 25. Diagrama de paquetes relación entre paquetes del proyecto | 72 |
| Figura 26. Interfaz desarrollada del login | 73 |
| Figura 27. Interfaz principal desarrollada | 73 |
| Figura 28. Conexiones de un nodo normal de la red de sensores | 74 |
| Figura 29. Conexiones físicas de un nodo de la red sensores | 75 |
| Figura 30. Caja de proyectos Arduino | 76 |
| Figura 31. Conexiones de Raspberry pi y módulo NFRL2401 | 77 |
| Figura 32. Principales características de red de sensores | 77 |
| Figura 33. Nodo padre | 78 |
| Figura 34. Notificaciones generadas por aplicación Pushetta | 79 |
| Figura 35. Precio y características del vps | 81 |
| Figura 36. Información general del vps | 82 |
| Figura 37. Panel de configuraciones del vps | 82 |
| Figura 38. Aplicación manager de Tomcat | 83 |
| Figura 39. Información del servidor | 84 |
| Figura 40. Rendimiento CPU del vps | 85 |
| Figura 41. Interfaz de login que da acceso a todas las funcionalidades del sistema | 95 |
| Figura 42. Modal recuperar contraseña de la interfaz de login | 96 |
| Figura 43. Interfaz datos actuales | 96 |
| Figura 44. Interfaz de usuario | 97 |
| Figura 45. Interfaz de calendario | 98 |

| | |
|---|-----|
| Figura 46. Modal Lista de Eventos/Soluciones de la interfaz calendario | 99 |
| Figura 47. Modal Agregar/actualizar evento de la interfaz calendario..... | 99 |
| Figura 48. Modal Agregar/actualizar solución de la interfaz calendario..... | 100 |
| Figura 49. Interfaz estadística..... | 100 |
| Figura 50. Modal traer datos por día de la interfaz estadística | 101 |
| Figura 51. Modal traer datos por mes de la interfaz estadística..... | 101 |
| Figura 52. Modal traer datos por año de la interfaz estadística | 102 |
| Figura 53. Interfaz tablas | 102 |
| Figura 54. Gestión de usuarios | 103 |
| Figura 55. Modal actualizar/crear usuario de la interfaz gestión de usuarios..... | 104 |
| Figura 56. Interfaz en la cual el usuario podrá crear y editar peces, por defecto el pez estará asociado a todos los sensores, si el pez es creado sus valores de rango estable serán dados por defecto por lo cual estos deben ser modificados, los indicadores son usados como referencia para los casos de incidencia. | 105 |
| Figura 57. Modal agregar pez de interfaz peces | 106 |
| Figura 58. Modal agrega/editar indicador de la interfaz peces | 106 |
| Figura 59. Interfaz donde se podrán crear y editar peceras, además de poder asociarles los peces que esta tendrá y el intervalo de tiempo en el cual estuvo..... | 107 |
| Figura 60. Modal agregar pecera de interfaz peceras | 108 |
| Figura 61. Modal agregar pez a pecera de interfaz peceras | 108 |
| Figura 62. Interfaz sensores | 109 |
| Figura 63. Modal agregar sensor de interfaz sensores | 110 |
| Figura 64. Interfaz donde se podrán crear y editar eventos, los cuales aparecerán en el calendario para ser agregado al respectivo día que se seleccione en el calendario..... | 111 |
| Figura 65. Modal agregar evento de interfaz eventos..... | 112 |
| Figura 66. Archivos del socket en el servidor..... | 113 |
| Figura 67. Interfaz administración de la información del usuario | 114 |
| Figura 68. Interfaz de calendario | 115 |
| Figura 69. Interfaz de calendario lista de incidencias y eventos..... | 116 |
| Figura 70. Interfaz Calendario agregar solución..... | 117 |
| Figura 71. Interfaz de calendario agregar evento..... | 118 |
| Figura 72. Interfaz de estadísticas..... | 119 |
| Figura 73. Interfaz de eventos..... | 120 |
| Figura 74. Interfaz gestión de operarios | 121 |
| Figura 75. Interfaz crear usuario | 122 |
| Figura 76. Interfaz de tablas de datos | 123 |
| Figura 77. Informe de datos | 124 |
| Figura 78. Interfaz de sensores | 125 |
| Figura 79. Interfaz de peces | 126 |
| Figura 80. Interfaz de peceras..... | 127 |

Capítulo 1. Desarrollo de un prototipo para monitorear la temperatura y el pH en la estación piscícola de la Universidad Francisco de Paula Santander Ocaña

1.1 Planteamiento del problema

Actualmente en la universidad Francisco de Paula Santander Ocaña se lleva a cabo la creación de una estación piscícola para la carrera de ingeniería ambiental, en complemento con el desarrollo de la estación se quiere sistematizar el control de la temperatura y pH de los futuros estanques, ya que son variables importantes en el desarrollo del pez.

La relación entre la temperatura del agua y el comportamiento del pez es evidente con tan solo pocos grados de variación. Muchos síntomas de enfermedades se confunden con estos ánimos o comportamientos. Hay maneras de contrarrestar la temperatura en peceras a través de calentadores pero en estanques que son de mayor tamaño solo queda hacer la mejor elección del tipo de pez que se quiere en el estanque con referencia a la temperatura ambiente, ya que existe una gran variedad en las temperaturas para cada pez. (Estanque y peces, 2002).

Los síntomas que puede presentar el pez y los más comunes son: El nado agitado, no querer comer o comer menos, agresividad y estrés. (El tiempo, 2013).

La concentración de iones de hidrógeno (hidrogeniones) es la que determina el pH o grado de acidez o alcalinidad del agua. Los valores posibles de pH en el agua pueden variar entre 0 y 14 (ambos puntos son los extremos de la tabla de valores). La relación entre el pez y su medio ambiente es muy estrecha ya que valores como la temperatura, dureza, conductividad, pH y otros actúan sobre el organismo de manera directa y determinan su biología. El agua de pH

neutro (pH 7) suele presentarse sin color, brillante, inodora y con una ligera presencia de algas sobre el material decorativo y vidrios. En este tipo de agua pueden vivir la mayoría de los peces. (Elacuarista, 2010).

Una elevación del pH produce en ciertos peces alcalosis, una enfermedad abiótica que podría compararse con una intoxicación y cuando el pH desciende más allá de lo tolerado por los peces, se produce en muchas especies otro tipo de enfermedad conocida como acidosis, la cual se manifiesta en forma de derrames sanguinolentos que pueden afectar el cuerpo y aletas de los peces. Algunas de las maneras de controlar el pH son: Añadir agua nueva regularmente, nunca cambiar el agua por completo esto podría afectar el equilibrio entre las bacterias, el pH y los químicos, ventilar el agua para mantener un nivel saludable de oxígeno, añadir soluciones y otros líquidos acondicionadores para mantener un nivel de pH saludable para los peces, retirar los peces o plantas muertas para evitar que se acumulen las bacterias o que los niveles de amoníaco o nitrato se eleven, ya que esto podría dar como resultado un nivel de pH dañino. (Elacuarista, 2010).

Como se puede identificar la temperatura y el pH son las principales variables a monitorizar para asegurarse de un buen ambiente y crecimiento del pez en la estación piscícola. La medición de del pH y la temperatura se toma de forma manual y periódicamente, lo cual puede presentar inconvenientes, debido que, al no tener el constante valor de las variables de los estanques, se pueden generar los inconvenientes mencionados en el párrafo anterior, por lo cual un sistema de monitoreo puede ayudarnos a agilizar el proceso de captura de variables, para que la toma de decisiones sea más oportuna.

1.1.1 Formulación del problema. ¿Es posible el desarrollo de un sistema aplicando conceptos de internet de las cosas para monitorear las variables de temperatura y pH en la estación piscícola de la universidad?

1.2 Objetivos

1.2.1 Objetivo general. Desarrollar un prototipo de monitoreo sistematizado para peceras de la universidad Francisco de Paula Santander Ocaña.

1.2.2 Objetivos específicos. Caracterizar los métodos y técnicas para el desarrollo de un prototipo IOT que monitoree la temperatura y pH, considerando el análisis de la información recolectada.

Construir el sistema que integre los elementos físicos, como lógicos basado en los lineamientos determinados para capturar, transmitir y analizar datos generados.

Realizar prueba para comprobar el correcto funcionamiento del sistema construido, evaluando el cumplimiento de sus requerimientos funcionales.

1.3 Justificación

Actualmente el avance de la tecnología se da a pasos agigantados es por esto que existen múltiples soluciones para los problemas existentes, y su aplicación para mejorar procesos como es el caso de la agricultura adaptando las operaciones a las necesidades de la planta o del animal según un ciclo de monitoreo-diagnóstico-recomendación, utilizando como base a los sensores y

tecnologías digitales asociadas. (Maurel & Huyghe, 2017) En el campo de los sensores existe una gran variedad con respecto a calidad, precio y utilidad, debido a esto en el desarrollo del proyecto se analizarán las mejores posibilidades del mercado, además se utilizará hardware libre para tener una mejor libertad ya que este se deja modificar sin la necesidad de pagar licencias, posee bastante documentación y una gran comunidad.

También se tendrán en cuenta todos los conceptos relacionados al proyecto y el más importante IoT:

Internet de las cosas (IoT) es la combinación de múltiples tecnologías y protocolos que genera un concepto de interconexión entre el mundo real y digital. El beneficio de IoT es la recopilación de información aplicándole inteligencia a objetos comunes y algunos complejos para conectarlos a través de internet e intercambiar información entre ellos, crear servicios que aporten beneficios tangibles a la sociedad, al medio ambiente, a la economía y a las personas.

Gracias a este surgimiento de tecnologías y nuevos conceptos como IoT, se ha comenzado a utilizar la tecnología para hacer seguimiento y mejora de los procesos que nos rodean a diario, como es el caso del desarrollo de framework para monitorear la frescura de la fruta en el comercio electrónico, que asegura la entrega de productos de calidad al usuario final. (Ruan & Shi, 2016) También existe el monitoreo de animales a través de redes de sensores, que se conectan con el animal a través de dispositivos en su cuerpo, normalmente aplicado en bovinos pero utilizables en cualquier tipo de animal. (Benaissa, y otros, 2017) Basados en trabajos como los mencionados se pueden generar soluciones a problemas como el planteado en

el proyecto, ya que lo que se desea hacer es implementar tecnologías para ofrecer un control sobre las variables de temperatura y pH de las futuras peceras que se desarrollan en la universidad Francisco de Paula Santander Ocaña, es importante la monitorización de la temperatura y del pH de la pecera para saber las condiciones exactas en las que habita el pez, y de esta manera mejorar la calidad de su medio, para que tenga un mejor crecimiento.

Este proyecto pretende definir un prototipo que sea escalable para que sirva como base del desarrollo a futuro de un sistema más complejo y que no solo se enfoque en la monitorización, si no que se puedan tomar acciones con el mismo, el sistema se encargara de medir la temperatura y el pH, las cuales serán tomadas a través de sensores específicos para cada variable, que se enlazarán mediante radiofrecuencia a un dispositivo, que tomara los datos de los sensores y los enviara a un servidor, los datos obtenidos se comparan con los valores estándar de la temperatura y del pH que favorezcan al ambiente correcto para el pez, cuando los datos salgan del rango de su valor aceptable se generaran avisos al encargado de la estación piscícola a través de una app y de una página web, advirtiéndole en tiempo real los posibles inconvenientes que se puedan generar, de esta manera optimizar los tiempos de respuesta con referencia a los posibles problemas que se den en la estación y a su vez documentar en la misma página web el progreso y lo realizado para solventar las incidencias presentadas.

En cuanto a la viabilidad del proyecto a continuación se detallan los campos y su cumplimiento:

Tabla 1*Campos de justificación*

| Campos | Descripción |
|---------------|--|
| Teórico | El tema principal que trata el proyecto cuenta con suficiente acceso a la información, gracias las bases de datos de artículos que la universidad posee, como lo es ACM, Sciencedirect y Scopus, además de todos sus recursos digitales. |
| Social | El estudio se realizara sobre las personas relacionadas a la estación piscícola, con la ejecución del proyecto no se alterara ni dañara a ninguna persona, comunidad, ni ambiente, más bien la finalidad del proyecto ayudara a las personas involucradas en su labor. |
| Financiero | El proyecto mejorara los procesos llevados a cabo en la estación piscicultura, mejorando su desempeño, para el desarrollo del proyecto se trabajara haciendo uso de los recursos que ofrece la universidad para la investigación, de esta manera poder financiar el desarrollo del proyecto. |

Fuente: Autor del proyecto

1.4 Delimitaciones

1.4.1 Geográfica. El proyecto se llevará a cabo en la futura estación piscícola de la sede central el algodonal de la universidad Francisco de paula Santander Ocaña.

1.4.2 Operativa. Durante la realización del proyecto se podrán presentar inconvenientes para el cumplimiento de los diferentes objetivos planteados, el proyecto se desarrollara teniendo en cuenta los procesos que se manejan en la granja, más específicamente en relación con la estación de piscicultura.

1.4.3 Conceptual. Para el diseño del proyecto se tendrán en cuenta los siguientes términos: IoT, sensores, pH, temperatura, bases de datos, web, hardware libre, Android, monitorización y piscicultura.

1.4.4 Temporal. El tiempo estimado como mínimo para el desarrollo del sistema de monitorización de temperatura y pH es de un año.

Capítulo 2. Marco Referencial

En esta sección del documento se habla sobre la importancia de IoT, el subcampo de nuestro interés que este ha generado y los desarrollos que se han realizado a nivel internacional y nacional, además se abordan los principales conceptos que se manejan alrededor de IoT y las redes de sensores, de la arquitectura de diseño de sistemas de IoT M2M y las leyes relacionadas al desarrollo de software en Colombia.

2.1. Marco histórico

El avance que representa IoT es tan grande que se extiende a nuevos campos como lo es la agricultura, gracias a la utilización de sensores y sistemas de información se puede controlar, monitorear, prevenir, entre otras tomas de decisiones que pueden mejorar el proceso de la agricultura. Desde hace ya tiempo se han venido incluyendo la tecnología en los procesos de agricultura con sistemas de control como las ayudas de riego controladas, verificaciones del ambiente, utilización de sistemas de información, entre otras. (Kaloxilos, y otros, 2013).

Gracias al concepto de IoT se generan nuevas posibilidades para la agricultura, en la investigación nos enfocaremos en un derivado de esta, la piscicultura a continuación veremos algunos ejemplos de IoT en la agricultura.

2.1.1 Antecedentes a nivel internacional. En Australia existe la granja de ostras Barilla Bay en bahía Pittwater la cual hace uso de IoT para medir calidad del agua y la temperatura con la finalidad de mejorar la salud y calidad de sus ostras, el principal uso que le dan a IoT es la predicción de eventos que puedan dañar la calidad del agua. (Guitierrez, 2016).

En indonesia la empresa eFishery hace utilización de IoT para el cultivo de peces y camarones, más específicamente en el control de la alimentación, ya que este es el principal gasto de la empresa y si se da un alimentación no medida generará sobre costos en la producción, la empresa utiliza alimentadores inteligentes que a través de sensores saben cuándo el pez tiene hambre y de esta manera alimentarlo adecuadamente, además de tener toda la información a la mano en tiempo real. (NLS, 2015).

2.1.2 Antecedentes a nivel nacional. En Colombia se da más una piscicultura tradicional que también hace uso de tecnología pero no al nivel de IoT, sin embargo existe apoyo del gobierno en cuanto información y procesos generales de la piscicultura, podemos encontrar documentos como el entregado por la OCDE un reporte sobre el estado de la piscicultura en el país, el Plan Nacional para el Desarrollo de la Acuicultura Sostenible en Colombia - PlaNDAS, también existen estudios en las universidades del país como el estudio de factibilidad para el establecimiento de una granja productora de alevinos en el departamento del Huila, evaluación del kit fish farming para análisis de calidad de agua en acuicultura, entre otras. (Higuera & Gonzales, 2006).

2.1.3 Antecedentes a nivel local. A nivel local no se encuentran antecedentes de la utilización de IoT en la piscicultura ya que esta se encuentra de la manera tradicional, lo cual es importante partiendo de que se puede a través de la investigación generar un precedente.

2.2 Marco contextual

La universidad Francisco de Paula Santander Ocaña fue fundada en 1974 según el acuerdo No. 03, por parte de la universidad Francisco de Paula Santander Cúcuta como expresión cultural y patrimonio de la región, es por esta razón que la universidad hace que sus procesos vayan acorde al mejoramiento de la región a través de la investigación y el mejoramiento continuo. La universidad en sus procesos de mejoramiento para optar por la alta acreditación ha decidido mejorar e invertir en el campus universitario, y teniendo en cuenta la facultad de ciencias agrarias han decidido construir una estación piscícola que forme parte de la granja experimental que posee la universidad para que ésta tenga más proyectos pecuarios que ayudan a la producción de animales y subproductos.

El Campus universitario posee un amplio terreno donde se encuentran laboratorios dedicados para la granja experimental, en el cual se ofrece un espacio físico idóneo, personal técnico y todas las herramientas necesarias para el desarrollo de la actividad académica de la facultad de ciencias agrarias y del ambiente, además presta diferentes servicios para las demás facultades y todas aquellas instituciones externas que lo requieran se ubica a la margen derecha del río Algodonal, dentro del campus universitario a una altura de 1150 msnm, con una temperatura promedio de 23°C, una humedad relativa del 70% y una extensión 135 ha.

Y es en la futura granja piscícola de la universidad la cual forma parte de la granja experimental y haciendo uso de los laboratorios que la universidad posee tanto de la facultad de ciencias agrarias y de la facultad de ingeniería que se llevará a cabo el desarrollo de la propuesta.

2.3 Marco conceptual

A continuación se presentan todos los conceptos, campos y temas relacionados al proyecto.

IoT

El Internet de las Cosas (IoT) es la integración de sensores dispositivos y sensores en objetos cotidianos que quedan conectados a Internet a través de redes fijas e inalámbricas. El hecho de que Internet esté presente al mismo tiempo en todas partes permite que la adopción masiva de esta tecnología sea más factible. Dado su tamaño y coste, los sensores son fácilmente integrables en hogares, entornos de trabajo y lugares públicos. (Bankinter, 2011).

Partiendo de este pensamiento cualquier objeto se puede conectar y manifestarse en la red, implica fuertes cambios en todo lo que conocemos. “IoT se presenta como un paradigma que parece predestinado a cambiar por completo escenario socioeconómico tal y como hoy lo conocemos”. (López, 2014)

Modelos de conectividad de IoT

La implementación de IoT utilizan diferentes modelos de conectividad cada uno de los cuales tiene sus propias características. Los cuatros modelos de conectividad descritos por la

junta de arquitectura de internet incluyen: dispositivo a dispositivo, dispositivo a la nube, dispositivo a puerta de enlace e intercambios de datos a través del backend. Estos modelos destacan la flexibilidad de los dispositivos que conforman IoT. (Rose, Eldridge, & Chapin, 2015)

Redes de sensores

Componente fuerte de IoT formado por pequeños sensores generalmente conectados por señales de radio que capta al mundo físico. “Los sensores son básicamente pequeños computadores con funciones extremadamente básicas. Consisten de una unidad de procesamiento con capacidad de computación restringida, una memoria limitada, un dispositivo de comunicación de radio, una fuente de alimentación, conformando nodos de sensores”. (Liñan Colonia, Vives, Bagula, Zennaro, & Pietrosemoli, 2015)

Las redes de sensores tienden a ser inalámbricas aunque puede haber excepciones, sin embargo la conectividad inalámbrica favorece las conexiones a áreas difíciles y evita las complicaciones que generalmente traen los cables.

AIoT

El crecimiento de IoT y su gran campo de acción hacen que existan desarrollos de IoT enfocados hacia algún campo en específico, es el caso de AIoT una sub rama de IoT, especializada solo en la agricultura y sus derivados. IoT puede ayudar a la agricultura en el monitoreo y control continuo que este necesita, en el marketing y en la toma de decisiones.

”Los servicios avanzados de IoT pueden acelerar el registro / modificación / cierre de fincas, vigilancia y la expedición de autorizaciones sanitarias.” (Borgia, 2014).

Hardware libre

El Hardware al igual que el software entra en la categoría de libre, pero a diferencia del software en el hardware libre se habla de que los diseños de las arquitectura, los diagramas esquemáticos y lo que uno quiera hacer es libre ya que se debe pagar para tener el hardware.

Piscicultura

La piscicultura es un campo de la acuicultura que se caracteriza por el cultivo de peces en estanques aunque a veces se da en jaulas flotantes en el mar o en lagos, esta ofrece ciertos beneficios en contraste a la pesca, ya que se pueden construir estanques en lugares no apropiados para la ganadería o agricultura, se produce el pez de acuerdo a la necesidad y se evita que el pez muera por sus depredadores naturales. La piscicultura entra en el campo de la acuicultura intensiva la cual se caracteriza por criar al pez controladamente.

Arduino

Arduino es una plataforma electrónica open-source de hardware y software libre, consta de un micro controlador, con él se puede controlar casi todo tipo de sensores y dispositivos electrónicos, además cuenta con su propio lenguaje y su propio IDE. Arduino pasó de ser una iniciativa a ser el cerebro de miles de proyecto, generando una gran comunidad y mucho conocimiento de acceso libre fácil de usar. (Arduino, 2005).

Raspberry pi

Raspberry pi es un computador de placa reducida, soporta los componentes necesarios para ser considerado un computador, y es una de las mejores alternativas para IoT, gracias a su accesible precio principal objetivo de sus creadores, un grupo de la Universidad de Cambridge y su misión es fomentar la enseñanza de los niños en las ciencias de la computación. Existen otros tipos de mini ordenadores pero Raspberry pi es el más conocido y el que posee una mayor comunidad, contando incluso con su propia revista la cual muestra las mejores alternativas de desarrollo con Raspberry, proyectos y mucho contenido de aprendizaje. (Raspberry pi, 2012).

Sensores

Los sensores son un complemento imprescindible a la hora del desarrollo de dispositivos con hardware libre ya que estos son (tomando como referencia al ser humano) los sentidos que le ofrecen al dispositivo un conocimiento sobre el ambiente o alguna parte del mismo. Los sensores relacionados al proyecto son los de temperatura y pH, en el sensor de temperatura mirándolo desde su lado comercial existen principalmente dos, el sensor normal y profesional marcando gran diferencia entre los dos, la vida útil del sensor. Entre los sensores de temperatura, existen muchas soluciones, algunas independientes del manejo de arduino y otras enfocadas en su uso, de estas se pueden resaltar sensor temperatura Max6675 800c, sensor de temperatura Ds18b20, termostato digital W1209 y sensor de temperatura Rtd Pt100, por su precio y su compatibilidad con el uso de arduino.

Módulo de conexión por radio frecuencia

Los módulos de conexión por radio frecuencia son el principal centro de las redes sensores, normalmente las más conocidas y utilizadas son las conexiones wifi y bluetooth respectivamente, pero las conexiones por radio frecuencia se caracterizan por utilizar módulos de bajo costo, cubrir áreas extensas y ser altamente configurables llegando al punto de modificar la antena que poseen los módulos con el fin de ampliar su cobertura. En este campo de sensores de bajo precio y con compatibilidad con arduino resaltan el módulo radiofrecuencia Rf 433 MHz que solo trabaja con un receptor y transmisor, y el módulo de radiofrecuencia Nrf24l01 capaz de generar redes con sus propios protocolos.

2.4 Marco teórico

En esta sección abordamos los conceptos tenidos en cuenta para la elaboración del proyecto:

Desarrollo web

Desarrollo de software enfocado a su uso a través de internet, mediante la utilización de navegadores web en combinación con bases de datos y el uso de lenguajes tradicionales, para el manejo de información y algunas tareas específicas, ya que cuenta con sus propias herramientas que ayudan en su desarrollo como lo son HTML, CSS y JavaScript, que pueden estar presentes en todos los campos del desarrollo de software, además de representar nuevas metodologías de diseño de software como cargas asincrónicas, maquetación rápida y agilidad de desarrollo.

(Hernández, 2014).

Arquitectura cliente-servidor

La arquitectura cliente-servidor hace referencia a la división de responsabilidades, en aplicaciones web donde el cliente es quien hace peticiones y el servidor quien le da respuesta a estas peticiones. Esta arquitectura posee ventajas como la configuración del servidor a medida, mejorando el desempeño de la aplicación en general, procesamiento en paralelo y manipulación de una base de datos para muchos clientes. (Flores Rosas & Martínez Escobedo, 2005).

Arquitectura modelo-vista-controlador

Patrón de arquitectura que se enfoca en la separación de las aplicaciones web, en modelos que se encarga de los datos, vistas representación visual de los datos y controladores que reciben órdenes del usuario, solicitan datos al modelo y se los pasan a la vista. Esta arquitectura es bastante importante, ya que aunque tiene varios años de haber sido presentada, sigue estando presente en la mayoría de frameworks actuales o por lo menos utilizan una variación de la misma. (Alvarez Hernandez , Campos Cantero, & Castelo Delgado, 2011).

Programación embebida.

Programación embebida, sistemas embebidos, empotrados o integrados es el nombre que recibe el software desarrollado para un dispositivo con una función en específico, caracterizado por poseer todos sus componentes incluidos en una placa base, los ejemplos más comunes de sistemas embebidos son los termostatos, sistemas de control de acceso, el software de una impresora entre muchas otras que también pueden encontrarse con el nombre de firmware. (Simon, y otros, 2013).

Fases IoT

IoT se caracteriza por su gran campo de acción pero independientemente de su aplicación siempre entra en la ecuación de IoT las siguientes variables:

- Los sensores, los cuales le servirán a los sistemas IoT como sus sentidos captando el mundo físico y convirtiéndolo a digital para que el sistema lo pueda analizar.
- La conexión, la cual en casi todos los casos estará ligada a internet ya que una de las bases de IoT es que los objetos siempre estén en conexión, aunque se dan excepciones en las cuales no necesariamente el sistema salga internet, sin embargo en estos casos también existe una conexión interna entre distintos dispositivos de un sistema mayor.
- Almacenamiento, a través del cual se asegurará la información recolectada, con la cual se analizará el tema o problema que trata el sistema para ofrecer soluciones debido a que este es otro punto fuerte de IoT, el cuál es el uso que se le da a la información almacenada.
- Las personas, ya que nosotros seremos los que le saquemos provecho a los sistemas IoT.

La arquitectura que se maneja en IoT es muy variada, debido a que IoT está presente en muchos campos y para todos los campos no se puede utilizar la misma, sin embargo la solución de IoT más común y la que se utilizará en el proyecto es la ETSI M2M la cual está compuesta por tres dominios o capas: **(i)** dispositivos y puerta de enlace, consiste en la conexión de dispositivos máquina a máquina(M2M) que generalmente son las conexiones internas de los sensores del sistema y como éstos se conectan a una puerta de enlace a internet o una red interna, **(ii)** dominio de red, conexión que permite la comunicación entre dispositivos y las aplicaciones

de los dispositivos, (iii) aplicaciones de dominio, se encuentran en el servidor y sirve de mediador para que el usuario interactúe con el dispositivo. (Borgia, 2014).

2.5 Marco legal

Decreto número 1780 de 2015 del ministerio de agricultura y desarrollo rural.

Decreto 1071 de 2015, Decreto Único Reglamentario del Sector Administrativo Agropecuario, Pesquero y de Desarrollo Rural, en lo relacionado con la adopción de medidas para administrar, fomentar y controlar la actividad de la acuicultura.

Estatuto General de Pesca Ley 13 de 1990 que declara el dominio público de los recursos pesqueros, define los recursos hidrobiológicos y los pesqueros, clasifica la pesca por el lugar donde se realiza (continental, marítima) y su finalidad (subsistencia, investigación, deportiva, comercial (artesanal, industrial, integrada), determina la conformación del sector (rector, ejecutor, financiero), crea al inpa y le establece las funciones, define los diferentes tipos de actividad pesquera (investigación, extracción, proceso, comercialización, acuicultura. según el medio (continental y marina) y según el manejo (re poblamiento, extensiva, semi-extensiva e intensiva), establece los modos para ejercer la actividad, establece el pago de tasas y derechos, estipula las infracciones, prohibiciones y sanciones, crea el registro general de pesca y acuicultura, crea el sepec, define a los pescadores y establece % de pescadores en las m/n extranjeras, establece mecanismos de coordinación interinstitucional y determina los incentivos a la actividad pesquera.

Decreto 2256 de 1991 que reglamenta todos y cada uno de los capítulos de la ley 13, establece la responsabilidad y mecanismos para determinar las cuotas de pesca, define cada tipo de actividad pesquera: continental (fluvial, lacustre); marina (costera, bajura, altura), define según la finalidad: subsistencia, comercial, artesanal, industrial, investigación, deportiva, determina la jurisdicción: todo el territorio nacional, en las aguas continentales, incluidos los ríos limítrofes, el mar territorial y la zona económica exclusiva, declara la pesca de subsistencia libre de permiso, señala los diferentes tipos existentes, los mecanismos para su obtención: (ministerio de ley permiso, patente, asociación, concesión, autorización), establece el cobro de tasas y derechos: valor de referencia smld, exceptúa a la acuicultura y la investigación, faculta para determinar los tipos de artes y aparejos y las zonas de veda o reserva, determina las prohibiciones, las sanciones, fija multas y obliga a la presentación de informes al sepec.

Administración de la pesca y la acuicultura en Colombia, AUNAP (2012 a la fecha, Decreto 4181 de 2011).

Ley 23 de 1982. Constituida como la ley “sobre Derechos de Autor”, los sujetos protegidos por dicho cuerpo normativo, serán los autores de obras literarias, científicas y artísticas, los cuales gozarán de protección para sus obras en la forma prescrita en esta.

Capítulo 3. Diseño Metodológico

3.1 Tipo de investigación

El proyecto será trabajo sobre la investigación descriptiva, (las investigaciones de tipo descriptiva trabajan sobre la realidad y su característica fundamental es la de presentar una interpretación correcta de los hechos que acontecen a la investigación. Lo primordial en una investigación descriptiva es descubrir algunas características fundamentales de conjuntos homogéneos de fenómenos a través de criterios sistemáticos que ayuden a identificar la estructura de estos o su comportamiento. De esta forma se pueden obtener las notas que caracterizan a la realidad estudiada.

El estudio descriptivo se caracteriza por la alta acumulación de conocimientos sobre un tema, lo cual puede hacer que una investigación pueda ser tomada como referente gracias a toda la información expresada y analizada.

3.2 Población y muestra

3.2.1 Población. La población objeto de estudio del proyecto serán las personas involucradas a la futura estación piscícola de la sede central el algodonal de la universidad Francisco de paula Santander Ocaña, la ingeniera Carmen Liceth Gracia y dos operarios de la estación.

3.2.2 Muestra. Como muestra para la elaboración del proyecto se trabajara con toda la población expuesta anteriormente.

3.3 Técnicas e instrumentos de la recolección de la información

El principal método de recolección de información serán entrevistas con los principales relacionados del problema de investigación, que en términos de desarrollo de software hablaríamos sobre la entrevista principal para obtener los requerimientos del software, posteriormente con búsquedas investigativas se complementara esta información y se validara. Teniendo en cuenta que la entrevista es la herramienta que resalta entre las opciones como cuestionarios, análisis y observación de procesos, para la toma de requerimientos debido a su variación y la interacción directa con las personas que llevan a cabo o lideran los procesos. (Kendall & Kendall, 2005).

Una vez recopilada la información y después de haber sido validada, se procederá al desarrollo del prototipo, haciendo uso de las construcciones más útiles y usuales del lenguaje de modelado unificado UML, enfocadas en las necesidades del profesional, estos modelos están relacionados con la metodología proceso unificado racional RUP, ya que esta hace uso de ella y nos permite crear productos basados en el enfoque de la orientación a objetos en su diseño y permite su personalización para ser adaptados a cualquier escala. (Fontela, 2011).

Capítulo 4. Administración del proyecto

4.1 Administración del proyecto

4.4.1 Recursos humanos. El estudiante Brayan Sánchez Torres, estudiante del programa ingeniería de sistemas. Director Alveiro Alonso Rosado Gómez Magister en Gestión Aplicación y Desarrollo de Software y docente del programa Ingeniería de Sistemas, Especialización en Auditoría de Sistemas e Informática Educativa modalidad virtual.

4.4.2 Recursos financieros.

Tabla 2

Presupuesto global del proyecto (En miles \$)

| RECURSOS | VALOR DE GASTOS |
|--------------------|-----------------|
| Equipo de Cómputo | 1000 |
| Arduino(2) | 56 |
| Raspberry | 190 |
| Sensores y módulos | 800 |
| Transporte | 150 |
| Papelería | 100 |
| Imprevistos | 500 |
| Trabajo de autores | 2000 |
| Total | 3896 |

Fuente: Autor del proyecto

4.4.3 Recursos institucionales

- Asesoría del director del trabajo de grado
- Biblioteca Argemiro Bayona Portillo
- Bibliotecas virtuales disponibles en la universidad
- Laboratorio de robótica de la universidad

Capítulo 5. Resultados

En la presente sección se exponen los resultados obtenidos de la realización del proyecto, esta sección se encuentra dividida en subniveles basados en las etapas que presenta en el ciclo de vida de desarrollo de software desarrollo en cascada.

5.1 Requerimientos

El proyecto cubrirá la problemática existente en la granja piscícola, de la universidad Francisco de Paula Santander Ocaña en la sede algodonal, ya que en esta se desea implementar un sistema que ayude a mantener en buen estado el agua de las peceras, para que en ella se den las bacterias necesarias para generar un ambiente favorable para el crecimiento de la especie de pez tilapia, el sistema prototipo a desarrollar monitoreara las variables de temperatura y pH, ya que estas son las principales variable que pueden afectar a las bacterias. Este proyecto pretende definir un prototipo que sea escalable para que sirva como base del desarrollo a futuro de un sistema más complejo y que no solo se enfoque en la monitorización, si no que se puedan tomar acciones con el mismo, el sistema se encargara de medir la temperatura y pH, las cuales serán tomadas a través de sensores específicos para cada variable, que se enlazarán a través de radiofrecuencia a un dispositivo, que tomara todos los datos de los sensores y los enviara a un servidor, los datos obtenidos se compararan con los valores estándar de la temperatura y del pH que favorezcan el ambiente correcto para el pez, cuando los datos salgan del rango de su valor aceptable se generan avisos al encargado de la estación piscícola a través de una app y de una página web, advirtiéndole en tiempo real los posibles inconvenientes que se puedan generar, de esta manera optimizar los tiempos de respuesta con referencia a los posibles problemas que se

den en la estación y a su vez documentar en la misma página web el progreso y lo realizado para solventar las incidencias presentadas.

Para el desarrollo del proyecto se utilizó el ciclo de vida de desarrollo de software RUP (proceso unificado racional) como principal base, como se mencionó en el capítulo anterior y se reforzó con el modelo en cascada razón por la cual los niveles de este modelo, están presentes en la estructura de los capítulos, exceptuando la fase de mantenimiento, ya que el proyecto es contemplado solo hasta la fase de pruebas. Esta combinación de modelos se da gracias a que RUP es un modelo que se puede acoplar a la exigencia de cualquier otro modelo existente (IBM, 2002) y de que el desarrollo del proyecto consta de poco personal enfocándonos en un desarrollo por niveles como lo hace el modelo en cascada, del modelo unificado racional se hace principal uso de los artefactos que este plantea para la comprensión, el análisis y los diseños del sistema que están presentes en su mayoría en cada capítulo del proyecto.

5.1.1 Requisitos funcionales y no funcionales. A continuación se detalla los requisitos de los proyectos tanto funcionales como no funcionales obtenidos de entrevistar a la ingeniería ambiental Carmen Liceth Gracia quien es encargada del proyecto piscícola de la Universidad Francisco de Paula Santander Ocaña.

La entrevista fue llevada a cabo como una conversación libre, donde predominaron las preguntas abiertas y algún par cerradas donde se necesita la obtención de información concreta sobre el proceso llevado a cabo por ella, debido a que se trabajó una entrevista de tipo conversacional también se hizo uso de la preguntas de sondeo para redirigir la entrevista en este

tipo de preguntas se suele preguntar el porqué de algo o pedir el uso de ejemplos para ilustrar los conceptos o ideas explicadas, la realización de esta entrevista nos deja como resultado los requerimientos funcionales y no funcionales del sistema.

5.1.1.1 Requisitos funcionales del sistema.

- El sistema debe permitir la captura, análisis, almacenamiento y visualización de la temperatura y pH en las peceras que se encuentran en la granja de la Universidad Francisco de Paula Santander Ocaña.
- El sistema debe notificar al usuario cuando los valores de temperatura y pH se encuentran fuera de los rangos establecidos.
- El sistema permitirá visualizar datos estadísticos con la información almacenada previamente en la captura de las variables.
- El sistema también permitirá generar reportes de la información expuesta en el apartado estadístico del mismo.
- En el sistema se podrán incluir incidencias o eventos que estén relacionadas con las peceras de la Universidad y las variables de temperatura y pH.
- El sistema permitirá la creación de usuarios solo cuando se autentique el usuario como administrador.
- El sistema permitirá la modificación de los datos del usuario o administrador.

5.1.1.2 Requisitos no funcionales del sistema.

- La aplicación del sistema debe poder utilizarse en navegadores Firefox y Chrome.

- El sistema debe poseer interfaces graficas bien formadas y con una buena experiencia de usuario.
- El sistema debe tener disponibilidad del 99%.
- El sistema debe asegurar que los datos estarán protegidos de accesos no autorizados.
- El sistema dejara de funcionar en caso de fuego o agua.
- Los dispositivos deben tener una alimentación de energía independiente y estable.
- El sistema debe contar con instrucciones para la configuración de los dispositivos.
- El tiempo de toma de datos debe ser el mismo periodo que se demoran los sensores en tomar mediciones estables.
- Debe existir un plan de acción para cuando se corte el servicio de energía o internet.
- El sistema debe contar con un almacenamiento en la nube.

5.2 Análisis

Análisis de las tecnologías existentes y disponibles en cuanto a hardware y software en la construcción de dispositivos IoT con enfoque en la agricultura o redes de sensores que puedan ser aplicadas para la elaboración del proyecto (Jamieson & Herdtner, 2015), teniendo en cuenta características como:

- Cantidad de usuarios.
- Costos de dispositivos.
- Complejidad de las tecnologías.
- Recursos y tiempo.

Anteriormente se expusieron los principales conceptos y tecnologías que se utilizaran, en esta sección se comentan otras alternativas y el porqué de las seleccionadas, basados en los campos de IoT expuesto en el marco teórico del proyecto, los cuales nos hablan de tres campos o dominios de IoT, de estos dominios para la selección de tecnologías se tienen en cuenta los dominios de aplicación y de dispositivos y puerta de enlace, ya que el dominio de red puede estar fuera del control de la aplicación ya que este dominio se encarga de como la información viaja a través de internet desde el dominio de aplicaciones al dominio de dispositivos y puerta de enlace o al revés.

Basados en lo expuesto anteriormente se presentan dos secciones donde hablamos de las tecnologías en hardware y software presentes en los dos dominios seleccionados, ya que esta es una manera fácil de ver como cada componente del sistema entra en los dominios seleccionados mientras están separados en capas físicas y lógicas. (Martinez Valencia, 2016).

5.2.1 Tecnologías Disponibles en Hardware. Es necesario identificar el hardware apropiado a utilizar para satisfacer las necesidades del proyecto, debido a la gran variedad existente en este campo, por este motivo, en esta sección se hablarán de las opciones más conocidas y que tienen una mayor comunidad, las tecnologías en hardware están más ligadas al dominio de dispositivos y puerta de enlace ya que con estos se construyen los dispositivos que servirán para la toma de datos del mundo real para los sistemas IoT.

5.2.1.1 Procesamiento de datos. Basado en los requerimientos nos dimos cuenta de la necesidad de procesar y controlar el flujo de datos que son arrojados por sensores y otros dispositivos que intervienen en un sistema IoT presentes en la capa de dispositivos, lo cual nos hace pensar, cuál debe ser el hardware que pueda cumplir con estas necesidades llevándonos a observar las plataformas de microcontroladores y computadores de placa reducida más populares del mercado:

Arduino

Arduino es una plataforma de electrónica de código abierto basada en hardware y software fácil de usar. Las placas Arduino pueden leer entradas (luz en un sensor, un dedo en un botón o un mensaje de Twitter) y convertirlo en una salida, activar un motor, encender un LED y publicar algo en línea (Arduino, 2005), el Arduino más común debido a su documentación y su fácil uso, es el Arduino uno, a continuación se presentan sus características:

- Placa de microcontrolador basada en ATmega328P.
- Tiene 14 pines digitales de entrada / salida, 6 entradas analógicas (entrega 5 y 3 voltios).
- Tiene cristal de cuarzo de 16 MHz.
- Tiene conexión USB y un conector de alimentación (9-12voltios).
- Tiene un encabezado ICSP y un botón de reinicio.

Tessel

Tessel es una plataforma de desarrollo de IoT y robótica de fuente completamente abierta e impulsada por la comunidad. Abarca tableros de desarrollo, complementos de módulos de hardware y el software que se ejecuta en ellos (Tessel, 2013). A continuación se presentan sus características:

- Preinstalled with Node.js 4.2.1 LTS or newer
- Easy to use CLI for remote programming and configuration
- 580MHz Mediatek Router-on-a-Chip
- 48MHz Atmel SAMD21 Coprocessor
- 64MB DDR2 RAM
- 32MB Flash Storage
- Wirelessly Programmable 802.11 b/g/n Wifi
- Supports 10/100 Ethernet
- Micro USB Power
- 2x 10-Pin Tessel Module Ports
- 2x USB Puertos

Orange pi

Es una computadora de una sola fuente abierta. Puede ejecutar Android 4.4, Ubuntu, Debían, Raspbian Image. Utiliza AllWinner H3 SoC y tiene 1GB DDR3 SDRAM, sus características son:

- Procesador Cortex-A7 de 4 núcleos @ 1.6GHz.
- 1GB de memoria Ram DDR3.
- Soporte para tarjetas Micro-SD de hasta 64 GB.
- Salida AV y HDMI, tanto para vídeo como para audio.
- Tarjeta de red Fast Ethernet 10/100.
- 3 puertos USB 2.0 y un puerto USB OTG.

BeagleBone.

Definida como la computadora ampliable da hardware abierto, sus principales características son:

- Procesador TI OMAP3530 SoC - Núcleo ARM Cortex-A8 de 720 MHz.
- Núcleo 'HD capable' TMS320C64x+ (Arriba de los 520 MHz puede mostrar 720p a 30 fps).
- PowerVR SGX 2D/3D de Imagination Technologies con soporte para dos pantallas independientes.
- RAM LPDDR de 256 MB.
- Memoria flash NAND de 256 MB.
- DVI-D (Resolución máxima de 1280×1024).
- Entrada para tarjetas SD/MMC y jacks de entrada y salida de audio.
- Puerto RS-232 y conector JTAG.
- Puerto de alimentación (Conector de 5 V de tipo barril).

Raspberry pi

Pc de placa reducida explicada en marco conceptual del documento, Esta placa cuenta con varios modelos, como los son la pi Zero y la gama de Raspberry, para este análisis se tiene en cuenta la Raspberry pi 3 modelo b por contar con más renombre y ser hasta el momento la más completa de la familia Raspberry, a continuación se presentan sus características:

- Quad Core 1.2GHz Broadcom BCM2837 64bit CPU
- 1GB RAM, Micro SD
- BCM43438 wireless LAN and Bluetooth Low Energy (BLE) on board
- 100 Base Ethernet.
- 40 pines GPIO.
- Cuatro puertos USB.

5.2.1.1.1 Plataformas seleccionadas. Las plataformas seleccionadas para el proyecto son Arduino y Raspberry pi debido a su amplia comunidad que cuenta con documentación para la elaboración de proyectos parecidos al presente en este proyecto, además de ser plataformas económicas para el desarrollo de prototipos y gozan de gran reputación en el campo de la educación siendo utilizadas en el caso de Raspberry pi para la enseñanza de programación y nociones básicas sobre computadoras. (Espeso, 2016).

5.2.1.2 Sensores. Existe una gran variedad de sensores que pueden dar solución al problema propuesto en el proyecto, además de algunas plataformas que cubren necesidades parecidas pero enfocadas a acuarios de menor tamaño, sin embargo estas soluciones son muy costosas y utilizarlas hacen que debas elegir todo un grupo de tecnologías como lo es el caso de Open Aquarium, razón por la cual se enfocó la búsqueda en sensores que resolvieran nuestra necesidad en particular sin tener que adquirir más dispositivos que no fueran un Arduino. Los sensores seleccionados fueron las sondas de temperatura ds18b20 y de pH profesional. A continuación enumeramos sus características:

Sonda ds18b20.

- Rango de temperatura: -55 a 125°C
- Resolución: de 9 a 12 bits (configurable).
- Interfaz 1-Wire (Puede funcionar con un solo pin)
- Identificador interno único de 64 bits.
- Múltiples sensores puede compartir el mismo pin.
- Precisión: $\pm 0.5^\circ\text{C}$ (de -10°C a $+85^\circ\text{C}$).
- Tiempo de captura inferior a 750ms.
- Alimentación: 3.0V a 5.5V.

Sonda de temperatura ds18b20



Figura 1. Sonda de temperatura DSB18b20

Fuente: (<https://www.filipeflop.com/blog/sensor-de-temperatura-ds18b20-arduino/>)

Sonda de pH profesional.

- Alimentación: 5.00V.
- Dimensiones: 43x32mm (controlador).
- Rango de medición: 0-14 pH.
- Temperatura de medición: 0-60 °C.
- Precisión: ± 0.1 pH (25 °C).
- Tiempo de respuesta: ≤ 1 min.
- Sonda de pH con conector BNC.

- Controlador pH 2.0 (3 pines).
- Ajuste de ganancia.
- Indicador LED.

Tabla 3*Características del Electrodo de pH*

| Voltage (mV) | pH value | Voltage (mV) | pH value |
|---------------------|-----------------|---------------------|-----------------|
| 414.12 | 0.00 | -414.12 | 14.00 |
| 354.96 | 1.00 | -354.96 | 13.00 |
| 295.80 | 2.00 | -295.80 | 12.00 |
| 236.64 | 3.00 | -236.64 | 11.00 |
| 117.48 | 4.00 | -117.48 | 10.00 |
| 118.32 | 5.00 | -118.32 | 9.00 |
| 59.16 | 6.00 | -59.16 | 8.00 |
| 00 | 7.00 | 00 | 7.00 |

Fuente: Autor del proyecto

Sonda pH profesional



Figura 2. Sonda de pH profesional

Fuente: (<https://tienda.bricogeek.com/otros/748-sensor-analogico-de-ph-profesional.html>)

5.2.1.3 Comunicaciones. En este campo se analizaron varios dispositivos de conexión, enfocados en conexiones entre dispositivos y hacia internet, ya que estos dispositivos son los que representan la puerta de enlace entre el dominio de dispositivos y el dominio de red, en este dominio predominan las conexiones por radiofrecuencia. (Vidyasagaran, Devi, Varma, Charan, & Rajesh, 2017).

Módulo ESP8266

Es un chip integrado con conexión Wifi y compatible con el protocolo TCP/IP, se caracteriza por ser de un tamaño pequeño en relación con otros dispositivos que ofrecen la misma solución, además de ser bastante económico.

Sus características son:

- Protocolos soportados: 802.11 b/g/n.
- WIFI: Wi-Fi Direct (P2p), Soft Access Point.
- Stack: TCP/IP integrado.
- PLL: reguladores y unidades de manejo de energía integrados.
- Potencia de salida: +19.5dBm en modo 802.11b.
- Alimentación: 3.3 - 3.6V mas de este rango lo puede quemar, pero las entradas tx y rx soportan 5v.
- Sensor: de temperatura integrado.
- Consumo en modo de baja energía: menor de 10 uA.
- Procesador: integrado de 32 bits, puede ser utilizado como procesador de aplicaciones.

Shield Ethernet Arduino.

El Arduino Ethernet shield nos da la capacidad de conectar un Arduino a una red Ethernet. Es la parte física que implementa la pila de protocolos TCP/IP. Soporta conexiones de sockets simultáneas teniendo como límite cuatro conexiones. Utiliza la librería Ethernet para leer y escribir los flujos de datos que pasan por el puerto Ethernet. El shield cuenta con un conector

Ethernet RJ45 estándar. La Ethernet shield permite el apilamiento de placas arduino, ya que dispone de unos conectores que le permiten conectar placas arduino sobre ella (Arduino, 2005). Es una solución bastante práctica ya que te deja trabajar cómodamente sin limitarte los pines, ya que no te los ocupa todos, además de poseer un buen soporte y documentación, pero su precio hace que el monto total del producto final pueda aumentar bastante.

Sus características son:

- Opera a 5V suministrados desde la placa de Arduino.
- El controlador Ethernet es el W5100 con 16K de buffer interno. No consume memoria.
- El shield se comunica con el microcontrolador por el bus SPI, por lo se debe usar la librería SPI.
- Soporta hasta 4 conexiones simultáneas
- El shield dispone de un lector de tarjetas micro-SD que puede ser usado para guardar ficheros y servirlos sobre la red.

Módulo NFRL2401

Módulo transceptor de la compañía Nordic Semiconductor, Este transceptor trabaja en la banda de 2.4GHz e incorpora adicionalmente extra pipelines, buffers, y un modo de auto retransmisión, que permite crear redes de sensores, existen variaciones en las cuáles, algunas poseen antenas que mejoran su alcance pero a su vez consumen mayor energía.

Características:

- Voltaje de alimentación de 1.9 a 3.6V
- Banda ISM de 2.4GHz
- Velocidad de datos de hasta 2Mbps.
- Operación a muy baja potencia.
- 11.3mA TX a 0dBm de potencia de salida
- 12.3mA RX a una velocidad de datos de 2Mbps
- 900nA estando apagado
- 22 μ A en espera-I
- ShockBurst™ Mejorado
- Compatible con NRF2401A, 02, E1 y E2

Módulo NFRL2401



Figura 3. Módulo de radiofrecuencia NFRL2401

Fuente: (<http://www.eneka.com.uy/robotica/modulos-comunicacion/m%C3%B3dulo-wireless-nfr2401-2.4ghz-detail.html>)

Tabla 4*Descripción pines módulo NFRL2401*

| Pin | Name | Pin function | Description |
|------------|-------------|---------------------|--|
| 1 | CE | Digital input | Chip enable activates RX or TX mode |
| 2 | CSN | Digital input | SPI chip select |
| 3 | CSK | Digital input | SPI slock |
| 4 | MOSI | Digital input | SPI slave data input |
| 5 | MISO | Digital output | SPI slave data output, with tri-state option |
| 6 | IRQ | Digital output | Maskable interrupt pin. Active low |

Fuente: Autor del proyecto

Módulo RF transmisor y receptor ASK.

Módulos de transmisión por radiofrecuencia que trabaja en una sola dirección, esto quiere decir que un dispositivo transmisor solo se podrá conectar con un dispositivo receptor y viceversa.

Especificaciones.

Módulo TX:

- Voltaje de Operación: 3.5V~12V
- Entrada de datos: TTL
- Corriente de trabajo: Máximo $\leq 40\text{mA}$ (12V), mínimo $\leq 9\text{ mA}$ (3.5V);

- Alcance: 20 metros~200 metros (a mayor voltaje, mayor potencia de transmisión)
- Potencia de transmisión: 10 mW (5V);
- Frecuencia de transmisión: 433MHz
- Tasa de transferencia de datos: 4KB/seg
- Baud Rate recomendado: 2400
- Modulación: ASK/OOK
- Pines de salida: DATA(TX)/VCC/GND

Módulo RX:

- Voltaje de Operación: 5V DC
- Consumo de corriente: 4mA
- Sensibilidad del receptor: -105dB
- Frecuencia de recepción: 433MHz
- Modulación: ASK/OOK
- Pines: GND/DATA(RX)/VCC

5.2.1.3.1 Modulo seleccionado. El módulo seleccionado con respecto a los presentados anteriormente, es el módulo NFR24L0, debido a su capacidad de generar redes de sensores localmente, esto para brindar seguridad al sistema, ya que existiría un nodo padre que será el que salga a internet, este nodo tendrá un módulo de radiofrecuencia NFR24L01 para recibir la información de los demás nodos que también contarán con uno y hará uso de una Raspberry pi para conectarse a internet. Las razones principales para no tener en cuenta los demás módulos presentes en el dominio de dispositivos son:

- Complejidad del uso de un módulo de conexión Ethernet por sensor, además de su administración.
- Costos elevados de usar un shield de Ethernet.
- Conexión unidireccional en el caso de los módulos de transmisión ASK.

5.2.3 Tecnologías Disponibles en software. Se debe tener en cuenta y además diferenciar las tecnologías de software existentes para la programación de hardware y software, principalmente las tecnologías en software van ligadas al dominio de aplicaciones pero también existen herramientas que al estar ligadas al hardware también estarán ligadas al dominio de dispositivos.

5.2.3.1 Software para programación en hardware. Partiendo del hardware seleccionado para el dominio de dispositivos anteriormente podemos determinar las plataformas o lenguajes de programación a utilizar:

Entorno de desarrollo Arduino

El entorno de desarrollo Arduino es un entorno creado por la empresa Arduino para facilitar la programación de todo dispositivo relacionado a la marca, cuenta con su propio lenguaje basado en C y posee un gran comunidad con muchas librerías que permiten la programación de diversos sensores, esta herramienta al igual que los planos de Arduino son totalmente libres y se puede encontrar en la página oficial del mismo.

Python

Python es un lenguaje de programación interpretado, con una gran aplicación hacia el machine learning e inteligencia artificial se caracteriza por su fácil interpretación y escritura, es seleccionado en el proyecto para programar todo lo relacionado a la Raspberry ya que este es lenguaje que más usa su comunidad y el elegido para sus tutoriales por el magazine oficial de Raspberry, por la tanto es el que más tiene documentación de su uso en Raspberry superando a java y C/C++.

Fritzing

Fritzing es una herramienta de software libre que nos permite documentar y desarrollar los planos de nuestros dispositivos electrónicos, posee una buena integración con dispositivos de Arduino y una gran comunidad que le da soporte.

5.2.3.2 Software para programación de aplicaciones Web. En esta sección se tiene en cuenta el software para programación web ya que esta será la principal aplicación del dominio de aplicaciones al que el usuario final tendrá acceso.

Backend

El backend es la capa de accesos a datos de cualquier aplicativo, maneja la lógica de cómo son administrados los archivos, puede disponer de los recursos del servidor y administra las peticiones que se hagan desde el navegador. Para nuestro caso en particular se selecciona por dominio del desarrollador como lenguaje de backend Java, para trabajar con tecnologías como JSP, JPA, SERVELTS, el uso de un ORM(mapeo de objeto relacional) como lo es Hibernate, apache Tomcat como contendor de servelts y para el almacenamiento de la información se utiliza una base de datos tradicional en este caso PostgreSQL, no se ha seleccionado una base de datos no relacional debido a que el sistema contara con pocos usuarios y para estos casos priman las bases de datos tradicionales sobre las no relacionales. (Esteban Sanchis, 2016).

Frontend

Frontend es la parte visual de un software y a la cual el usuario final tiene acceso, de forma general son todas las tecnologías que corren en nuestro navegador, para el desarrollo del proyecto se tienen en cuenta las tecnologías base de la programación web HTML, CSS, JavaScript y se complementa el desarrollo con librerías como jquery, jqueryCalendar, Chartjs, Bootstrap y PDFjs (Esteban Sanchis, 2016). La página se desarrollara de manera responsiva para fácil visualización en dispositivos móviles.

Notificaciones.

Inicialmente se analizó el desarrollo de un aplicativo móvil para complementar el aplicativo web, pero la única opción adicional a la página web sería el uso de notificaciones, razón por la cual se decide optar por el uso de herramientas existentes que nos ofrezcan esta solución. Como es el caso de pusheta que es un sistema creado para enviar notificaciones en tiempo real a muchos dispositivos diferentes (teléfonos móviles, navegadores, televisión inteligente) y es configurable desde lenguajes como Python o C.

En esta apartado se selección las tecnologías referentes al hardware y software, ligadas a los dominios de aplicación y de dispositivos que se expusieron anteriormente, estas tecnologías se utilizaron para la elaboración del proyecto, teniendo como criterios de selección para los dispositivos que fueran tecnologías libres, con una comunidad de respaldo y económicos en su adquisición.

5.3 Diseño

En este apartado se exponen los diseños que se realizaron para la construcción del sistema, los diseños utilizados son los que principalmente recomienda la metodología RUP para el desarrollo de sistemas que ayudan a comprender los procesos relacionados, en la documentación de RUP se pueden encontrar los diseños como artefactos de apoyo para la comprensión de sistemas.

5.3.1 Diagrama de Arquitectura. Diagrama de arquitectura del proyecto, para el desarrollo de la arquitectura del proyecto se tiene en cuenta como arquitectura de referencia la ETSI M2M, la cual es la principal arquitectura de referencia en cuanto a construcciones de sistemas IoT, debido a que esta especifica los dominios en los cuales cada componente del sistema debe estar, dividiéndolos en tres partes:

- Dominio de aplicaciones: Interfaz en la que circulan los datos del sistema y donde se haya el motor de los procesos propios del servicio, además es la capa a la cual el usuario se conecta.
- Dominio de Redes: Dominio que hace referencia a los niveles de conexión de la aplicación para transmitir la información del sistema.
- Dominio de dispositivos: Dominio de los dispositivos electrónicos generalmente sensores que se comunican entre ellos para crear redes de sensores con las cuales no interactúa el usuario.

Diagrama de referencia ETSI M2M

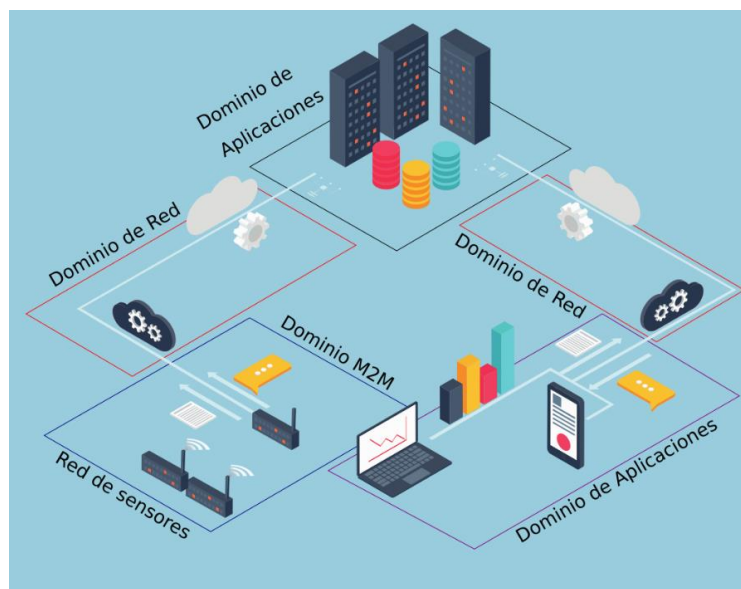


Figura 4. Diagrama de referencia ETSI M2M

Fuente: Autor del proyecto

Diagrama de bloque sobre el funcionamiento del sistema interno el cual ilustra las comunicaciones de la red de sensores, obtención de valores a través de los sensores y su relación hacia el servidor.

Diagrama de bloques para el proyecto

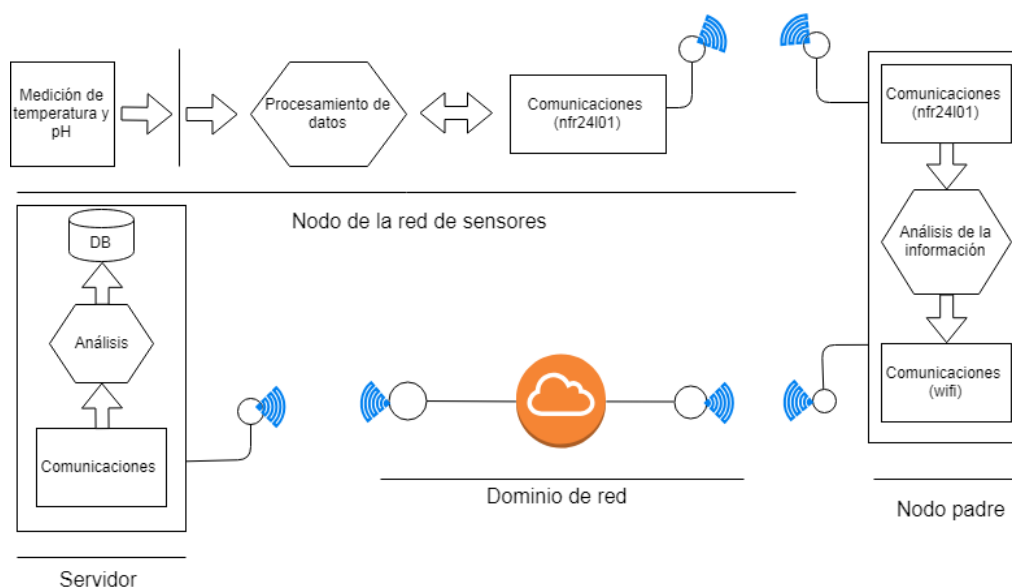


Figura 5. Diagrama de bloques del sistema interno

Fuente: Autor del proyecto

Diagrama de arquitectura realizado para el proyecto, en este diagrama se tiene en cuenta que el sistema interno es la capa de dominio de dispositivos y que hace uso de una red de sensores conectados por radiofrecuencia, el sistema es la capa de aplicación y el internet es la capa de red, al igual que el modelo de referencia ETSI M2M.

Diagrama de arquitectura para el proyecto

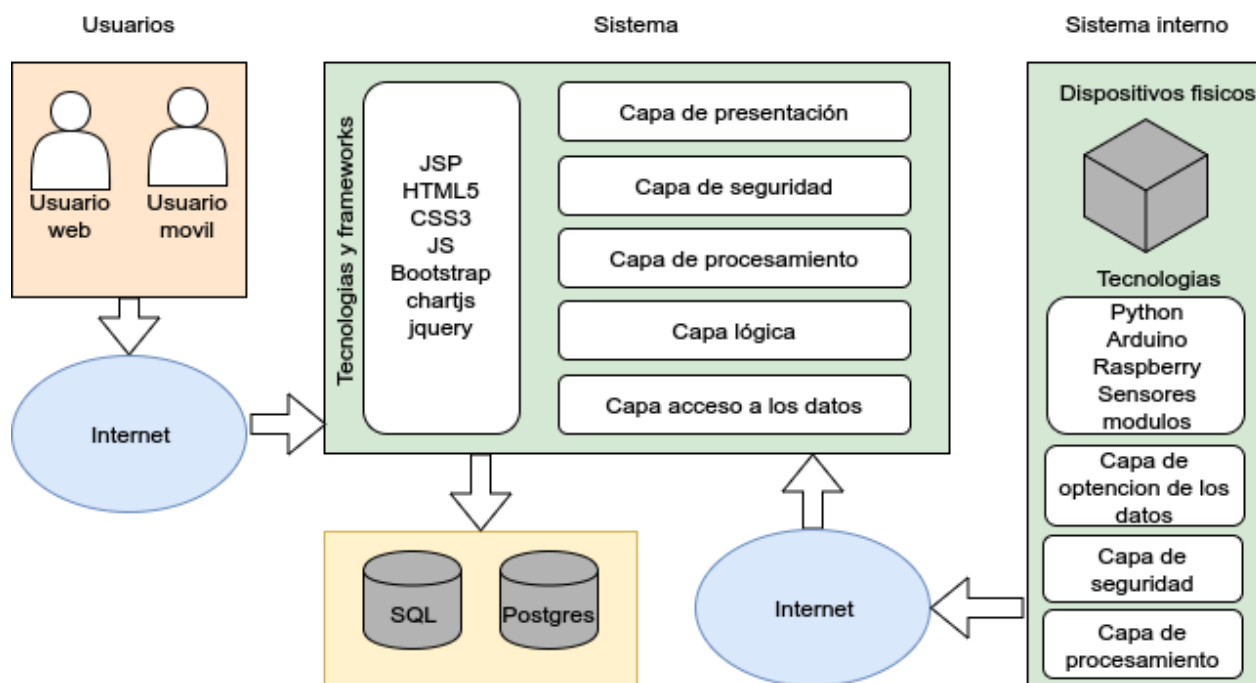


Figura 6. Diagrama de arquitectura

Fuente: Autor del proyecto

5.3.2 Diagrama y descripción de casos de uso. El diagrama de casos de uso es el artefacto que nos permite conocer el comportamiento del sistema cuando este interactúa con un actor del mismo, puede ser representado de forma gráfica y también puede ser encontrado de forma escrita como descripción de casos de uso (Fontela, 2011), en esta sección se presentan los actores del sistema, el diagrama de casos de uso y su descripción.

Los actores presentes en el sistema son un usuario base quien será un operario de la estación piscícola que tendrá acceso a toda la información relacionada a la administración de la estación y un administrador que en este caso será el encargado responsable de la estación, quien contara con los mismo privilegios que un usuario normal pero además será capaz de administrar los usuarios bases del sistema ya que solo a través de él se podrán crear.

Diagrama de casos de uso para el proyecto

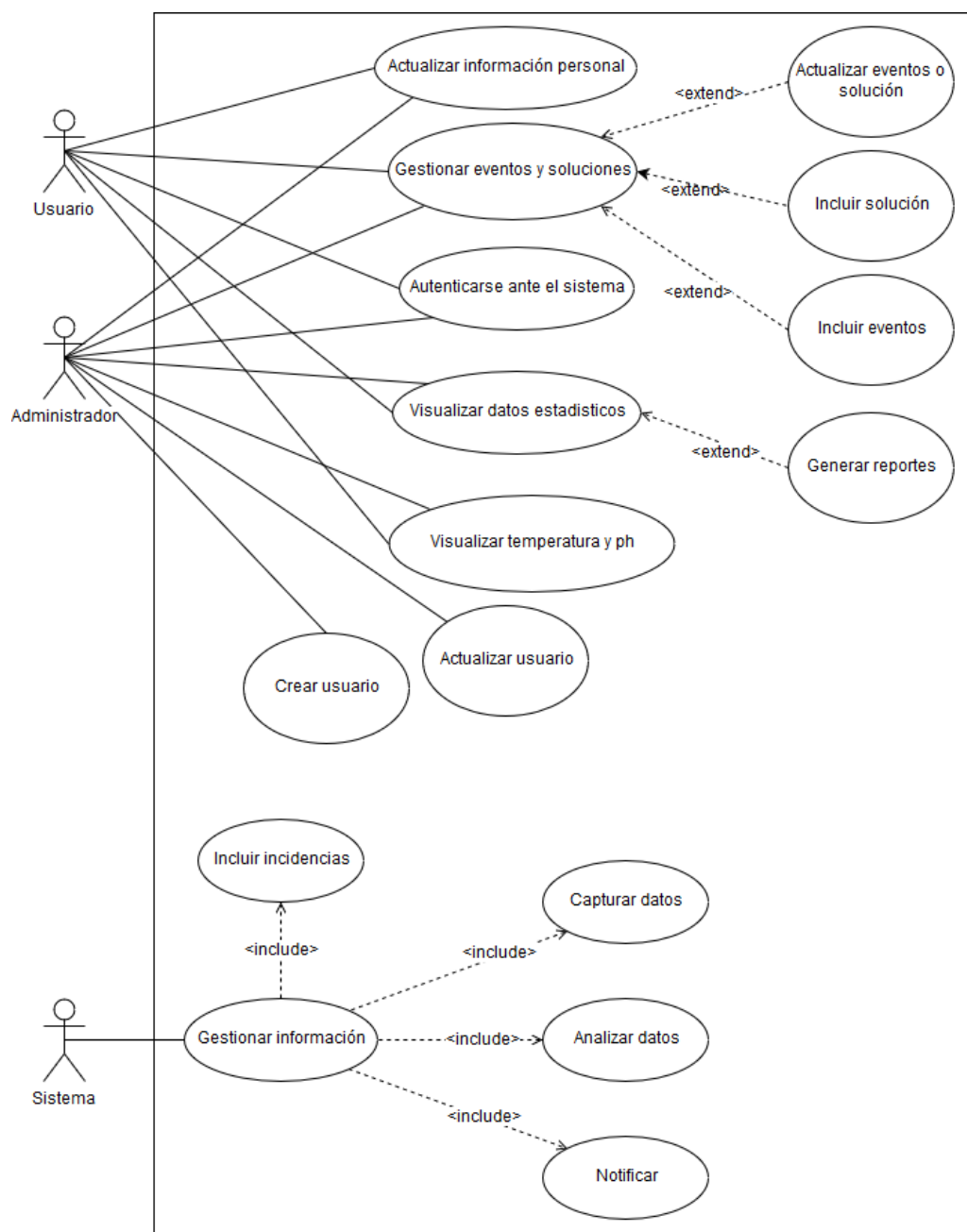


Figura 7. Diagrama de casos de usos

Fuente: Autor del proyecto

La descripción de casos de usos es el artefacto utilizado por RUP para entender el desarrollo de sistemas, empleada para detallar los casos de usos presentes en el proyecto y sus respectivos flujos alternos, a través de una descripción completa, informal o breve (Fontela, 2011), en este caso hacemos uso de la descripción breve:

Caso de uso visualizar temperatura y pH

| ESPECIFICACIÓN DE CASOS DE USO | |
|--|--|
| Nombre del Caso de Uso | Visualizar temperatura y pH. |
| Descripción | Observar los valores de temperatura y pH de cada pecera. |
| Actor (es) | Usuario, Administrador. |
| Pre-condición | Usuario o administrador autenticado ante el sistema. |
| Frecuencia | Alta |
| Importancia(1 a 5) | 5 |
| Flujo Básico | |
| ACTOR | SISTEMA |
| 1. El caso inicia cuando el usuario o administrador va a observar los valores de temperatura y pH. | 2. Consulta valores de sensores. |
| | 3. Muestra las peceras por tarjetas con sus respectivos valores para sus sensores. |
| Post-condición | Valores de temperatura y pH expuestos. |
| Flujos alternos | |
| 2.1 Si la consulta falla el sistema muestra mensaje de error de conexión y queda esperando que se reestablezca la misma. | |
| Observaciones | |
| Ninguna | |
| Requerimiento | |
| El sistema debe permitir la captura, análisis, almacenamiento y visualización de la temperatura y pH en las peceras que se encuentran en la granja de la Universidad Francisco de Paula Santander Ocaña. | |

Figura 8. Caso de uso visualizar temperatura y pH

Fuente: Autor del proyecto

Caso de uso autenticarse ante el sistema

| ESPECIFICACIÓN DE CASOS DE USO | |
|---|--|
| Nombre del Caso de Uso | Autenticarse ante el sistema. |
| Descripción | Permite autenticarse ante el sistema para acceder a las opciones principales del mismo. |
| Actor (es) | Usuario, Administrador. |
| Pre-condición | Acceder al sitio web del proyecto y haber seleccionado la opción de autenticación. |
| Frecuencia | Alta |
| Importancia(1 a 5) | 5 |
| Flujo Básico | |
| ACTOR | SISTEMA |
| 1. El caso inicia cuando el usuario o administrador va a autenticarse ante el sistema para ver las estadísticas y generar reportes de los valores de temperatura y pH. | 2. Muestra formulario de autenticación con los campos de correo y contraseña y un botón de recuperar contraseña. |
| 3. Diligencia los campos del formulario. | 4. Valida la información del formulario. |
| | 5. El sistema queda en el index. |
| Post-condición | Usuario o administrador autenticado. |
| Flujos alternos | |
| 4.1 Si los datos son erróneos o están vacíos, el sistema muestra un mensaje y vuelve al punto 2, mostrando los campos vacíos. | |
| 2.1 Si selecciona recuperar contraseña el sistema le pedirá un correo y lo validara, si este existe le pedirá que revise el correo para ver su contraseña, caso contrario le dirá que no existe y lo dejara en el punto 2 | |
| Observaciones | |
| Ninguna. | |
| Requerimiento | |
| El sistema permitirá la creación de usuarios solo cuando se autentique el usuario como administrador | |

Figura 9. Caso de uso autenticarse ante el sistema

Fuente: Autor del proyecto

Caso de uso actualizar información

| ESPECIFICACIÓN DE CASOS DE USO | |
|--|--|
| Nombre del Caso de Uso | Actualizar información. |
| Descripción | Permite modificar la información personal del usuario. |
| Actor (es) | Usuario, Administrador. |
| Pre-condición | Usuario o administrador autenticado ante el sistema. |
| Frecuencia | Baja. |
| Importancia(1 a 5) | 4 |
| Flujo Básico | |
| ACTOR | SISTEMA |
| 1. El caso inicia cuando el usuario o administrador desea modificar su información personal. | 2. El sistema muestra información personal del usuario en campos inputs que pueden ser modificados, todos estos dentro de una tarjeta. (Datos presentado: nombre, apellido, sexo, fecha de nacimiento, correo, documento y contraseña) |
| 3. Selecciona dato a cambiar. | 6. Valida la información. |
| 4. Modifica dato seleccionado. | 7. Almacena el evento con los nuevos datos. |
| 5. Selecciona opción guardar. | |
| Post-condición | Información del usuario o administrador autenticada. |
| Flujos alternos | |
| 6.1 Si los campos del formulario no están bien diligenciados, el sistema muestra un mensaje y vuelve al punto 3. | |
| Observaciones | |
| Ninguna | |
| Requerimiento | |
| El sistema permitirá la modificación de los datos del usuario o administrador. | |

Figura 10. Caso de uso actualizar información

Fuente: Autor del proyecto

Caso de uso visualizar datos estadísticos

| ESPECIFICACIÓN DE CASOS DE USO | |
|--|---|
| Nombre del Caso de Uso | Visualizar datos estadísticos |
| Descripción | Se encarga de mostrar la información estadística de los valores de temperatura y pH capturada por el sistema. |
| Actor (es) | Usuario, Administrador. |
| Pre-condición | Usuario o administrador autenticado ante el sistema |
| Frecuencia | Media |
| Importancia(1 a 5) | 5 |
| Flujo Básico | |
| ACTOR | SISTEMA |
| 1. El caso inicia cuando el usuario o administrador va a visualizar datos estadísticos de la temperatura y pH. | 2. Muestra por defecto datos de la primera pecera y un input para realizar una búsqueda. |
| 3. Digita la información para el criterio de búsqueda. | 5. Realiza la búsqueda. |
| 4. Selecciona la opción buscar. | 6. Muestra la información de la búsqueda. |
| Flujos Alternos | |
| 5.1 Si el registro no coincide con ninguno de los criterios establecidos en la búsqueda, el sistema muestra un mensaje indicando que la información solicitada no existe en la base de datos y vuelve al punto 3. | |
| <i>Caso de Uso: Generar reportes</i> | |
| 1. El caso de uso inicia cuando el usuario o administrador desea generar reportes. | 2. Muestra un formulario para consulta. |
| 3. Digita los criterios para la búsqueda. | 4. Valida la información de la consulta. |
| 6. Elige la opción imprimir reporte. | 5. muestra la consulta. |
| | 7. Muestra ventana de confirmación. |
| | 8. Imprimir reporte. |
| <i>Flujo alternativo de generar reportes</i> | |
| 4.1 Si el criterio de búsqueda no coincide con la información de la base de datos, el sistema muestra un mensaje de error y vuelve al punto 3. | |
| 7.1 Si se elige la opción de no imprimir el sistema vuelve al punto 3(o puede terminar). | |
| Post-condición | Datos expuestos |
| Observaciones | |
| Ninguna. | |
| Requerimientos | |
| El sistema permitirá visualizar datos estadísticos con la información almacenada previamente en la captura de las variables. El sistema también permitirá generar reportes de la información expuesta en el apartado estadístico del mismo. | |

Figura 11. Caso de uso visualizar datos estadísticos

Fuente: Autor del proyecto

Caso de uso gestionar información

| ESPECIFICACIÓN DE CASOS DE USO | |
|---|--|
| Nombre del Caso de Uso | Gestionar información |
| Descripción | Se encarga de la gestión de las incidencias y eventos de las peceras de la Universidad Francisco de Paula Santander Ocaña. |
| Actor (es) | Sistema |
| Pre-condición | El sistema debe estar en marcha. |
| Frecuencia | Alta |
| Importancia(1 a 5) | 5 |
| Flujo Básico | |
| SISTEMA | |
| <i>Caso de Uso: incluir incidencias</i> | |
| 1. El caso de uso inicia cuando se detecta valores de temperatura y pH fuera de los rangos establecidos. | |
| 2. Los valores son guardados como incidencia. | |
| 3. Notificar al usuario. | |
| Post-condición | Incidencia registrada. |
| <i>Caso de Uso: Capturar datos</i> | |
| 1. El caso de uso inicia cuando se toman los valores directamente de los sensores. | |
| 2. Valor enviado a la Raspberry. | |
| 3. Valor almacenado temporalmente en Raspberry. | |
| 4. Verifica conexión con el servidor. | |
| 5. Se envía datos al servidor. | |
| 6. Se almacena datos (valores) en la base de datos. | |
| Post-condición | Información almacenada en la base de datos. |
| <i>Caso de Uso: Analizar datos</i> | |
| 1. El caso de uso inicia después de que los datos han sido almacenados. | |
| 2. Selecciona últimos valores almacenados. | |
| 3. Se verifican que los datos estén en los rangos correctos. | |
| Post-condición | Datos verificados. |
| <i>Caso de Uso: Notificar</i> | |
| 1. El caso de uso inicia cuando los valores analizados son erróneos. | |
| 2. Se envía una alerta al usuario. | |
| 3. Se verifica que la alerta ha sido visualizada. | |
| Post-condición | Notificación realizada. |
| Flujos Alternos | |
| <i>Caso de Uso: Capturar datos</i> | |
| 4.1 Si la conexión con el servidor es inexistente, el sistema mantiene la información almacenada y sigue intentando la conexión. | |
| <i>Caso de Uso: Analizar datos</i> | |
| 2.1. Si no hay información almacenada el caso de uso finaliza. | |
| <i>Caso de Uso: Notificar</i> | |
| 3.1 Si la alerta no ha sido visualizada, el sistema envía un correo al usuario. | |
| Observaciones | |
| La terminación del caso de uso incluir incidencias es la pos-condición del caso de uso notificar. En estos casos de uso no se involucra el usuario debido a que el sistema trabaja independientemente del usuario para estos casos. | |
| Requerimientos | |
| En el sistema se podrán incluir incidencias o eventos que estén relacionadas con las peceras de la Universidad y las variables de temperatura y pH. El sistema debe notificar al usuario cuando los valores de temperatura y pH se encuentran fuera de los valores establecidos. | |

Figura 12. Caso de uso gestionar información

Fuente: Autor del proyecto

Caso de uso gestionar eventos y soluciones

| ESPECIFICACION DE CASOS DE USO | |
|--|---|
| Nombre del Caso de Uso | Gestionar eventos y soluciones. |
| Descripción | Se encarga de la gestión de las eventos y soluciones de las peceras de la Universidad Francisco de Paula Santander Ocaña. |
| Actor (es) | Usuario, Administrador. |
| Pre-condición | Usuario o administrador autenticado ante el sistema. |
| Frecuencia | Media |
| Importancia(1 a 5) | 5 |
| Flujo Básico | |
| ACTOR | SISTEMA |
| <i>Caso de Uso: Incluir eventos</i> | |
| 1. El caso de uso inicia cuando el usuario o administrador desea registrar una solución o evento. | 2. Muestra calendario del sistema. |
| 3. Selecciona el día que quiere incluir una solución o evento. | 4. Muestra un formulario de registro. |
| 5. Diligencia los campos del formulario. | 7. Valida la información del formulario. |
| 6. Selecciona la opción Guardar registro de solución o evento. | 8. Guardar solución o evento. |
| Post-condición | Solución o evento incluido. |
| Pre-condición | Debe existir una incidencia. |
| <i>Caso de Uso: Incluir solución</i> | |
| 1. El caso de uso inicia cuando el usuario o administrador desea registrar una solución. | 2. Muestra calendario del sistema. |
| 3. Selecciona el día que tiene una incidencia y a la cual se le quiere incluir una solución. | 4. Muestra un formulario de registro. |
| 5. Diligencia los campos del formulario. | 7. Valida la información del formulario. |
| 6. Selecciona la opción Guardar registro de solución. | 8. Guardar solución. |
| Post-condición | Solución incluida. |
| <i>Caso de Uso: Actualizar solución o eventos</i> | |
| 1. El caso inicia cuando el usuario desea modificar un evento o solución. | 2. El sistema muestra el calendario y una ventana para buscar si existe evento registrado. |
| 3. Selecciona día para editar evento o solución | 4. Realiza búsqueda. |
| 3.1. Digita información de criterio para hacer búsqueda de la evento. | |
| 3.2. Selecciona la opción buscar. | |
| 7. Realiza modificaciones necesarias. | 6. Muestra la información del evento o solución. |
| 8. Selecciona opción guardar. | 9. Valida la información. |
| | 10. Almacena el evento con los nuevos datos. |
| Post-condición | Solución o eventos editados. |
| Flujos Alternos | |
| <i>Caso de Uso: Incluir incidencia o eventos</i> | |
| 7.1 Si el formulario no está bien diligenciado, el sistema muestra un error y vuelve al punto 5. | |
| <i>Caso de Uso: Actualizar incidencias o eventos</i> | |
| 4.1 Si el registro no coincide con ninguno de los criterios establecidos en la búsqueda, el sistema muestra un mensaje indicando que la incidencia o evento no existe en la base de datos y vuelve al punto 3. | |
| 9.1 Si los campos obligatorios del formulario no están diligenciados, el sistema muestra un mensaje y vuelve al punto 7, mostrando los campos vacíos. | |
| 6.1 Si la incidencia o evento no existe se crea una nueva y continua el proceso normal de inclusión | |
| Observaciones | |
| Ninguna. | |
| Requerimientos | |
| En el sistema se podrán incluir incidencias o eventos que estén relacionadas con las peceras de la Universidad y las variables de temperatura y pH. | |

Figura 13. Caso de uso gestionar eventos y soluciones

Fuente: Autor del proyecto

Caso de uso crear usuario

| ESPECIFICACIÓN DE CASOS DE USO | |
|---|---|
| Nombre del Caso de Uso | Crear usuario. |
| Descripción | Permite al administrador la creación de usuarios. |
| Actor (es) | Administrador. |
| Pre-condición | Administrador autenticado ante el sistema. |
| Frecuencia | Baja. |
| Importancia(1 a 5) | 4 |
| Flujo Básico | |
| ACTOR | SISTEMA |
| 1. El caso inicia cuando el administrador desea agregar un usuario nuevo al sistema. | 2. El sistema muestra formulario necesario para la inclusión de un nuevo usuario. |
| 3. Completa el formulario | 5. Valida la información. |
| 4. Selecciona opción guardar. | 6. Almacena el nuevo usuario. |
| | |
| Post-condición | Usuario nuevo en el sistema. |
| Flujos alternos | |
| 5.1 Si los campos del formulario no están bien diligenciados, el sistema muestra un mensaje de error y vuelve al punto 3. | |
| Observaciones | |
| Ninguna | |
| Requerimientos | |
| El sistema permitirá la creación de usuarios solo cuando se autentique el usuario como administrador. | |

Figura 14. Caso de uso crear usuario

Fuente: Autor del proyecto

Caso de uso actualizar usuario. Fuente autor del proyecto

| ESPECIFICACIÓN DE CASOS DE USO | |
|---|---|
| Nombre del Caso de Uso | Actualizar usuario. |
| Descripción | Permite al administrador modificar a los usuarios del sistema. |
| Actor (es) | Administrador. |
| Pre-condición | Administrador autenticado ante el sistema. |
| Frecuencia | Baja. |
| Importancia(1 a 5) | 4 |
| Flujo Básico | |
| ACTOR | SISTEMA |
| 1. El caso inicia cuando el administrador desea modificar a un usuario. | 2. El sistema muestra una tabla con todos los usuarios del sistema. |
| 3. Selecciona usuario a editar. | 4. Muestra información en inputs del usuario. |
| 5. Modifica información del usuario. | 7. Valida información. |
| 6. Selecciona opción guardar. | 8. Almacena la nueva información del usuario. |
| Post-condición | |
| | Usuario nuevo en el sistema. |
| Flujos alternos | |
| 7.1 Si los campos del formulario no están bien diligenciados, el sistema muestra un mensaje de error y vuelve al punto 3. | |
| Observaciones | |
| Ninguna | |
| Requerimientos | |
| El sistema permitirá la creación de usuarios solo cuando se autentique el usuario como administrador. | |

Figura 15. Caso de uso actualizar usuario

Fuente: Autor del proyecto

5.3.3 Diagrama de clases. Artefacto del RUP que describe la estructura de un sistema mostrando las clases de los mismos, sus atributos y sus métodos, también puede tener información adicional que enriquezca el entendimiento del sistema (Fontela, 2011).

Diagrama de clases para el proyecto

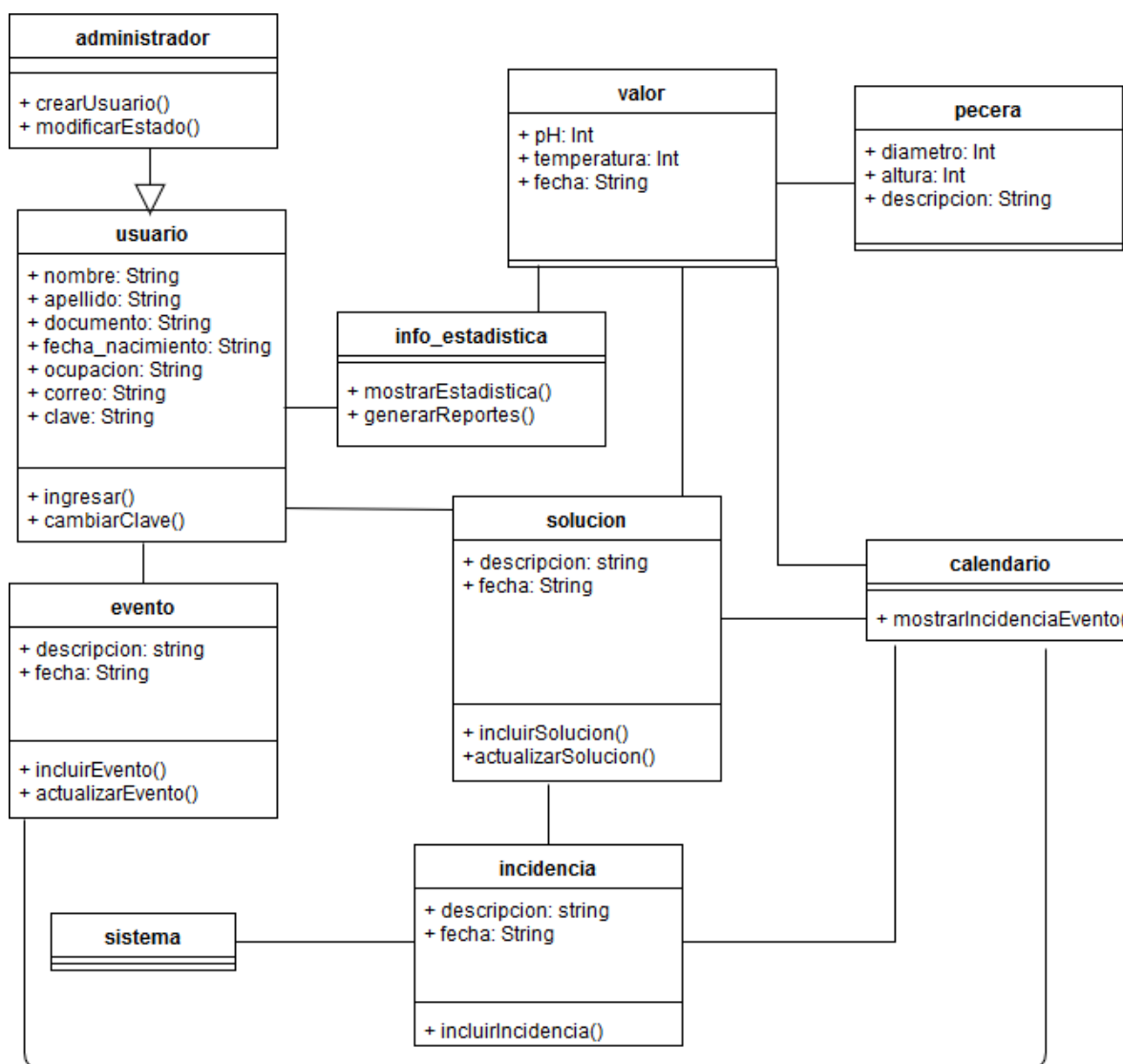


Figura 16. Diagrama de clases

Fuente: Autor del proyecto

5.3.4 Diagrama de secuencias. El diagrama de secuencias es un artefacto del RUP que nos permite ver las relaciones de las clases y que es modelado teniendo en cuenta los casos de uso (Fontela, 2011).

Diagrama de secuencia

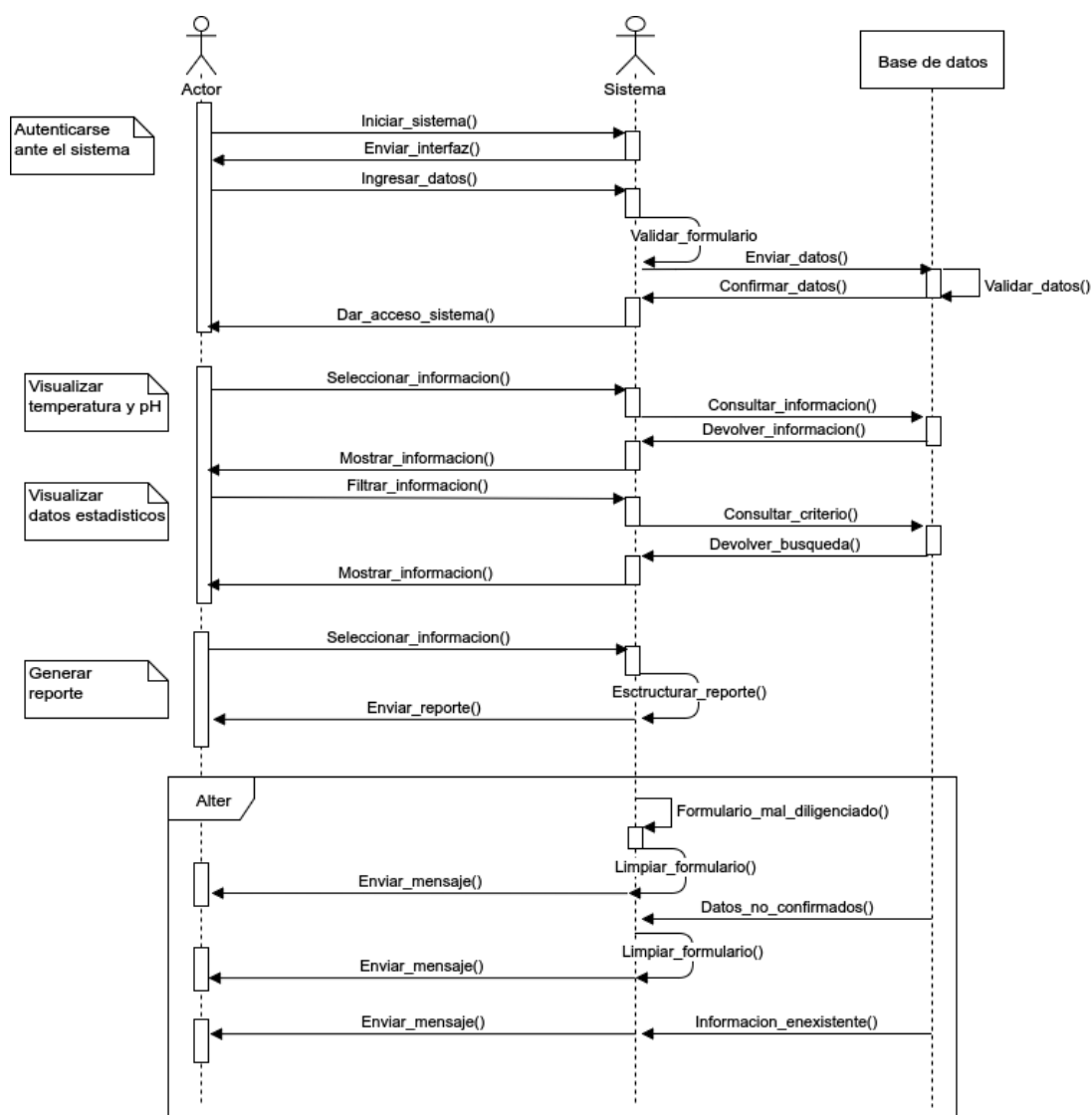


Figura 17. Diagrama de secuencias: Autenticarse ante el sistema, visualizar temperatura y pH, visualizar datos estadísticos y generar reportes. Fuente autor del proyecto

Fuente: Autor del proyecto

Diagrama de secuencia

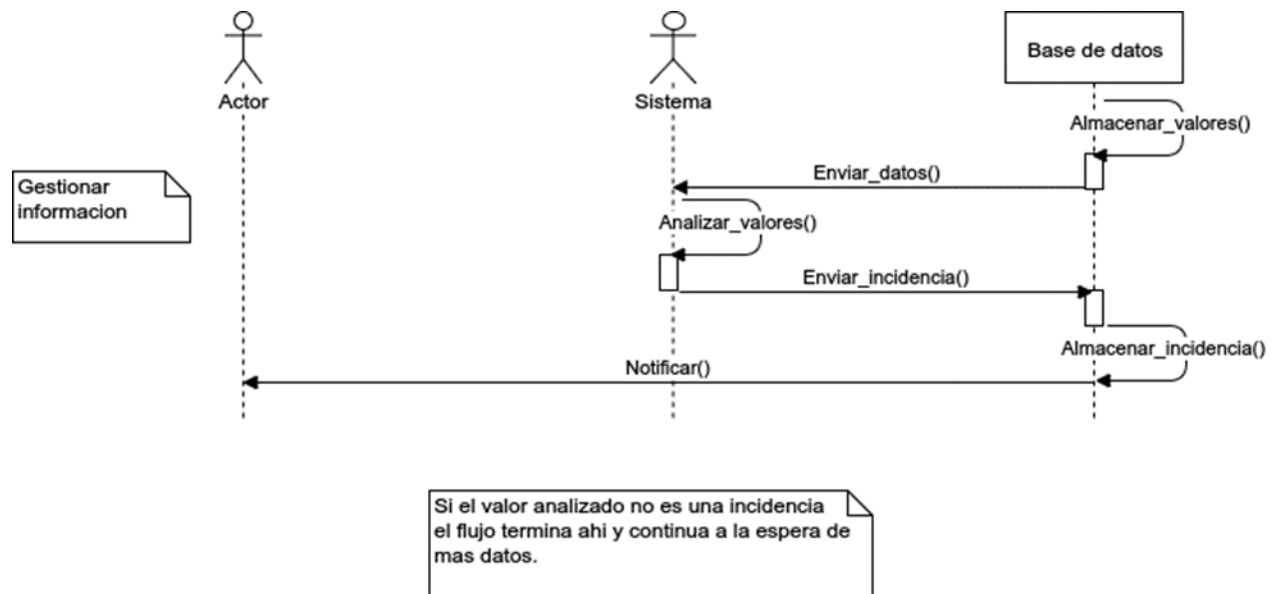


Figura 18. Diagrama de secuencia gestionar información

Fuente: Autor del proyecto

Diagrama de secuencia

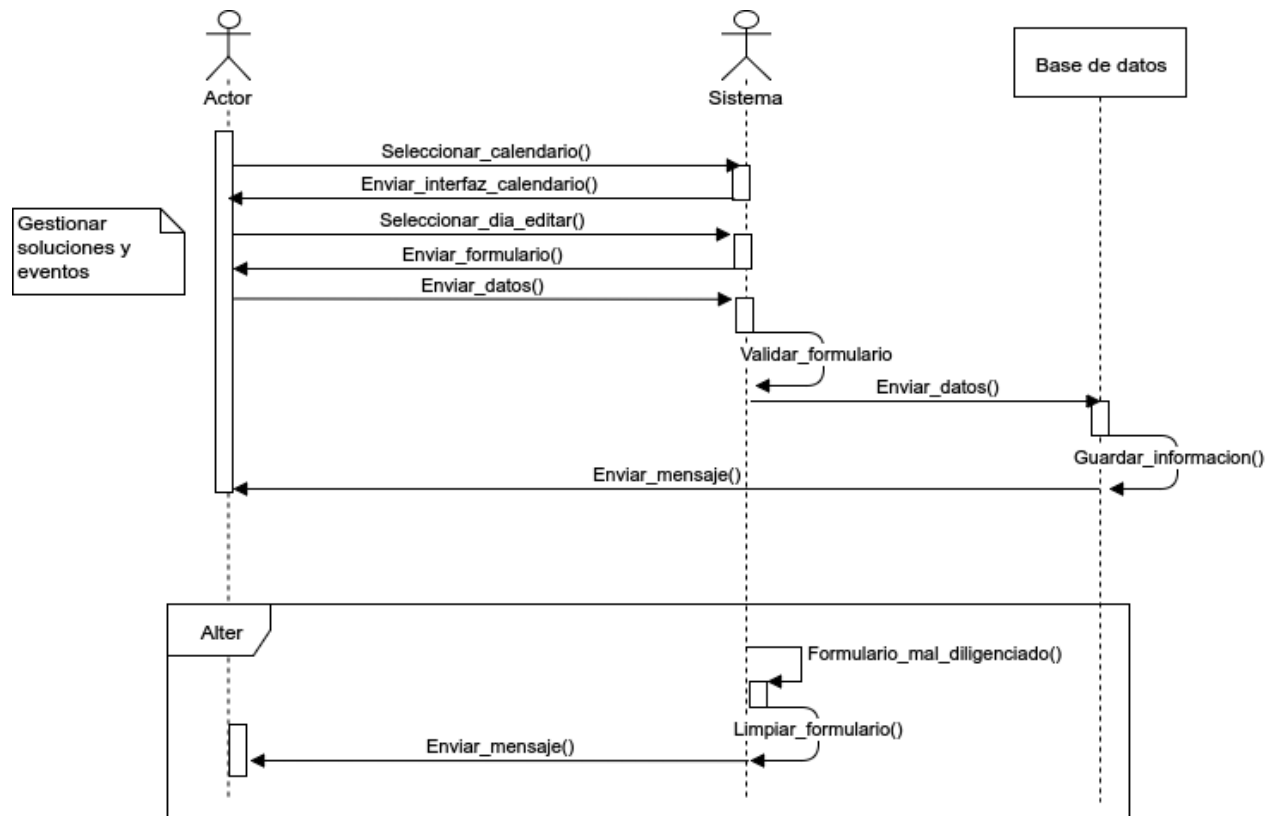


Figura 19. Diagrama de secuencia gestionar soluciones y eventos

Fuente: Autor del proyecto

Diagrama de clases

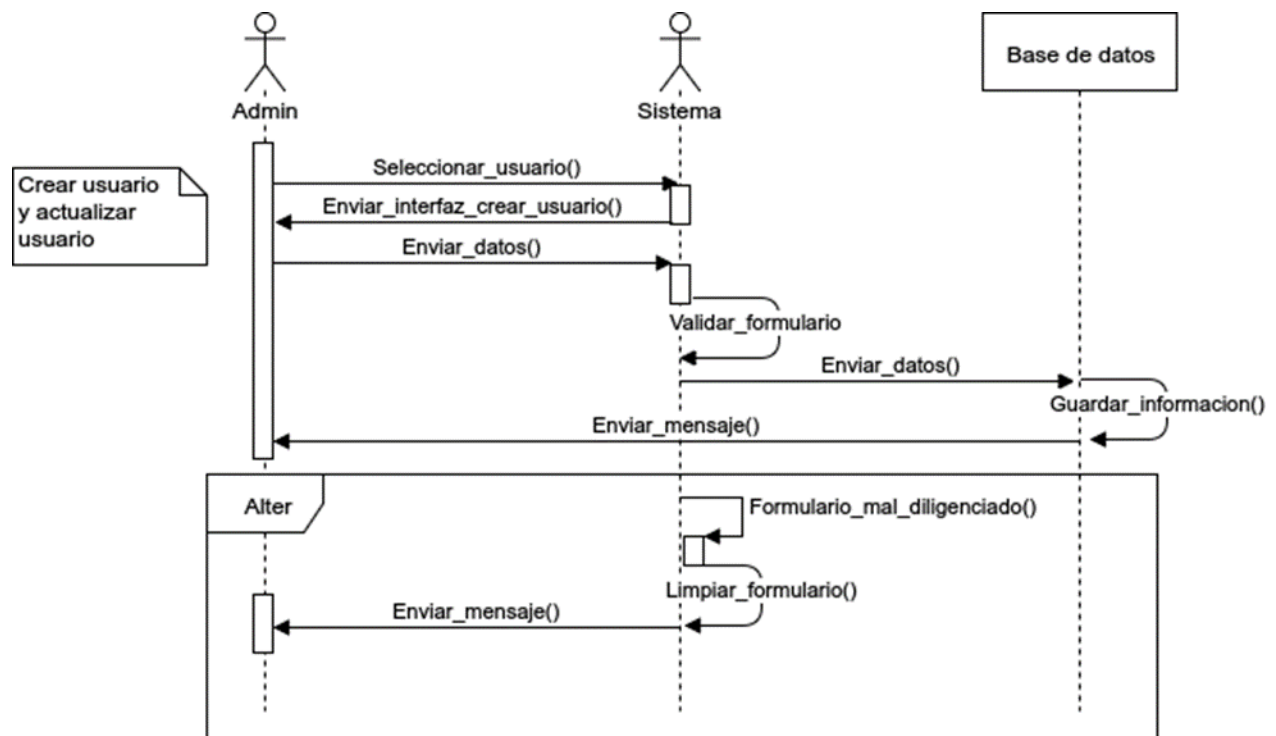


Figura 20. Diagrama de secuencia crear usuario y actualizar usuario

Fuente: Autor del proyecto

5.3.5 Modelo entidad relación. Diagrama de flujo que muestra como los objetos de un sistema se relacionan entre si dentro de sí mismos, usado para diseñar bases de datos ya que ayuda en términos de lógica y regla de negocio. (Fontela, 2011) El modelo entidad relación utilizado en el proyecto, teniendo en cuenta los artefactos expuesto anteriormente.

Modelo entidad relación

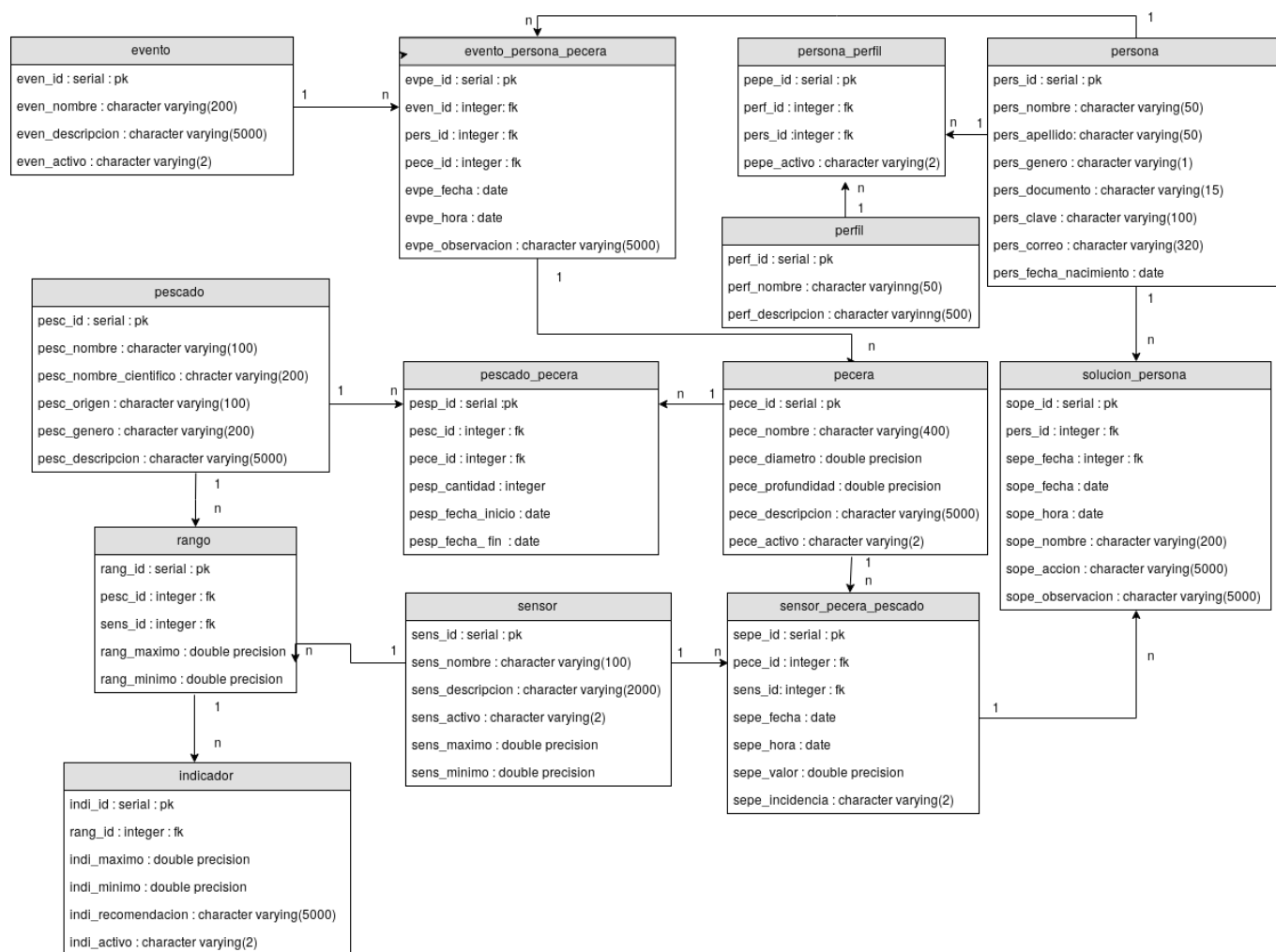


Figura 21. Modelo entidad relación

Fuente: Autor del proyecto

5.3.6 Controles.

Tabla 5

Tabla de los controles del proyecto para el administrador y los operarios (usuarios base):

| Usuarios | Módulos | Permisos |
|-----------------|--|----------------------------------|
| Administrador | Almacenamiento: Valores. | Consultar. |
| | Calendario: Evento, Solución. | Insertar, consultar, actualizar. |
| | Calendario: Incidencia. | Consultar. |
| | Seguridad: Usuarios, Backup, Restauración. | Insertar, consultar, actualizar. |
| | Parámetros: Rango de las variables, Eventos-tipo, Soluciones-tipo. | Insertar, consultar, actualizar. |
| | Informes: Crear reportes, imprimir reportes. | Insertar, consultar, actualizar. |
| Operarios | Almacenamiento: Valores. | Consultar. |
| | Calendario: Evento, Solución. | Insertar, consultar, actualizar. |
| | Calendario: Incidencia. | Consultar. |
| | Informes: Imprimir reportes. | Insertar, consultar, actualizar. |

Fuente: Autor del proyecto

5.3.7 Diseño de interfaces. El diseño de interfaces es la maquetación de cómo va a ser el software en su estado final, este es construido basado en los casos de uso del proyecto ya que su finalidad es dar un bosquejo de cómo sería la solución para que en el momento del desarrollo del software se pueda ser más productivo. (Fontela, 2011).

Árbol de navegación de interfaces

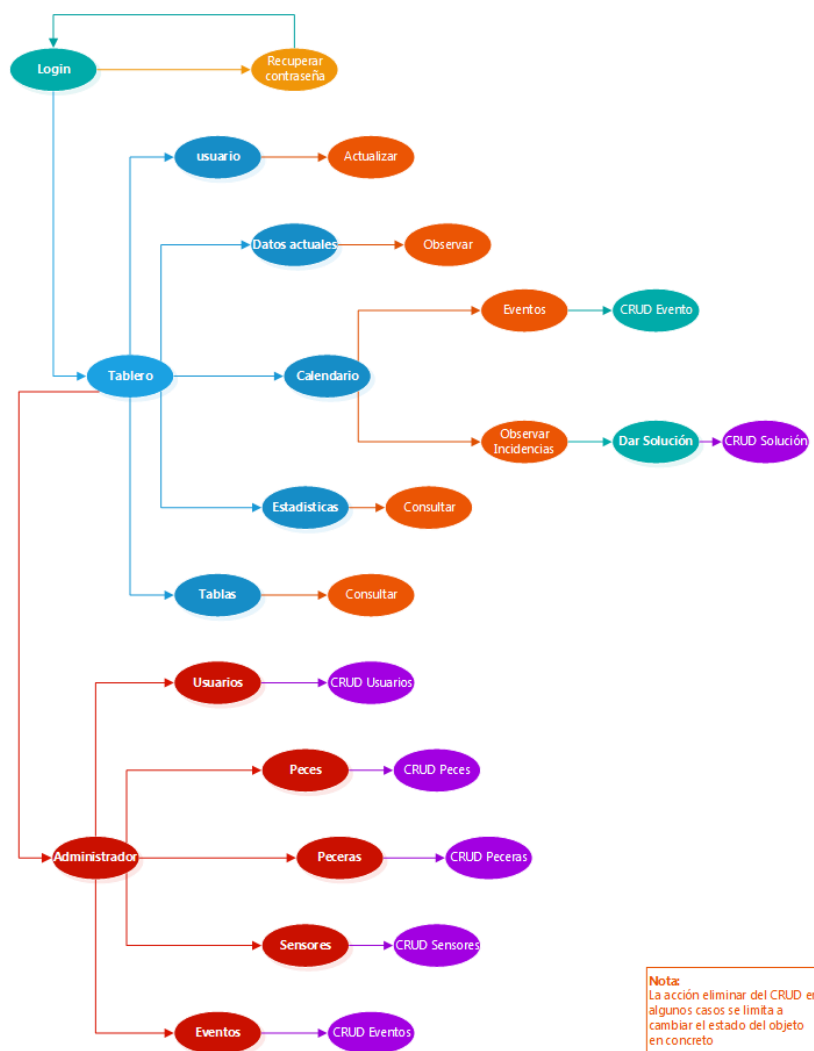


Figura 22. Árbol de navegación de interfaces.

Fuente: Autor del proyecto.

Diseño de interfaces de la página web, desarrollada en el software online ninja mock, están presentes en el anexo 3 del documento.

5.4 Desarrollo

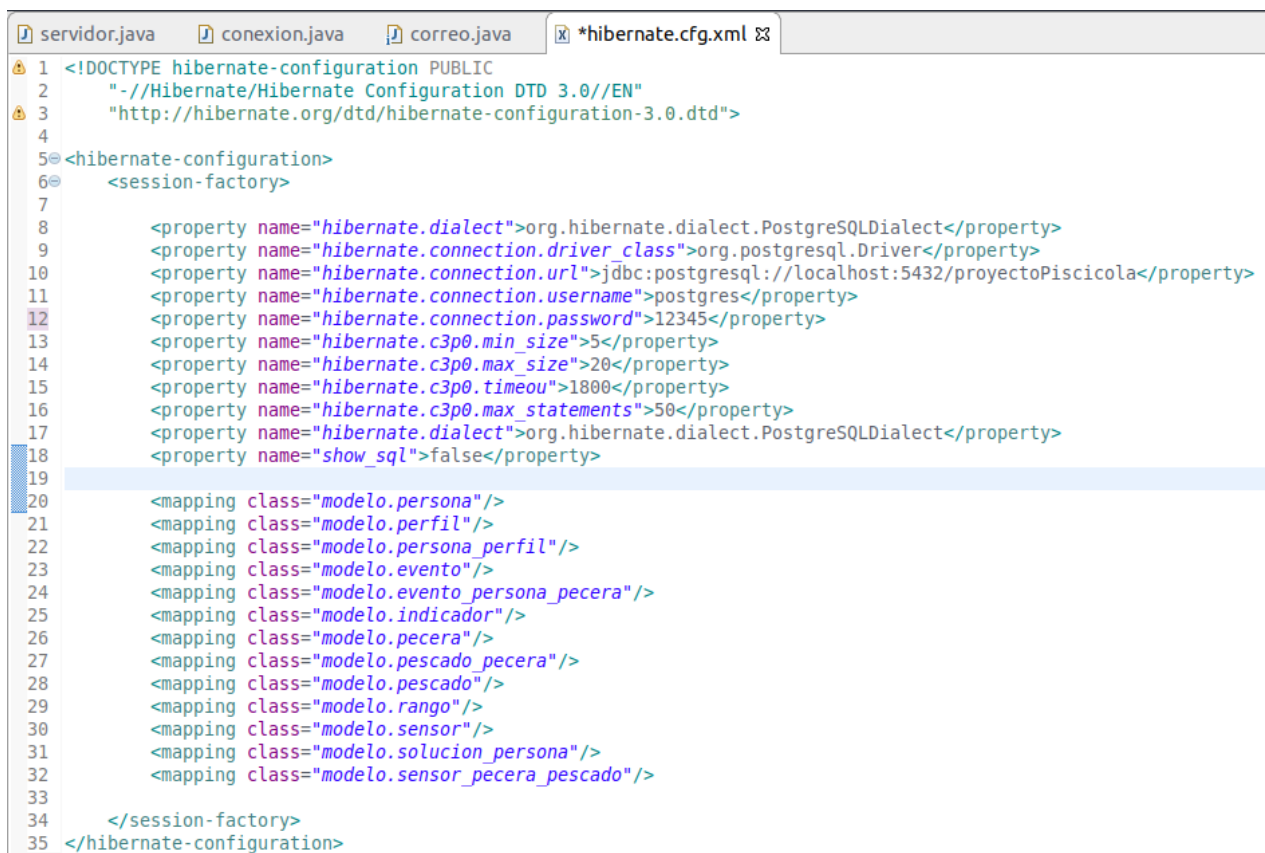
5.4.1. Desarrollo del aplicativo web. En esta apartado se comentara el desarrollo del aplicativo web y como se han implementado las tecnologías mencionadas anteriormente.

5.4.1.1 Funcionamiento general. El aplicativo web funciona teniendo como centro una base de datos que le da soporte a toda la información referente al sistema, tanto a los dispositivos como a la página web.

Al iniciar la aplicación nos encontramos con la interfaz de login, para poder iniciar sesión se nos pide las credenciales de correo y contraseña que deben ser creadas por el administrador previamente, si ya existe el usuario se puede recuperar contraseña si es necesario el caso, luego de que el usuario se autentica ante el sistema, tiene acceso a cinco módulos donde el usuario puede modificar datos personales, visualizar datos actuales de las peceras, visualizar datos en estadísticas o listarlos en tablas, también podrá ver y dar solución a incidencias en el calendario, además de crear eventos ligados a las peceras. Si el usuario logueado es administrador contara con cinco módulos más donde podrá administrar usuarios, peces, peceras, sensores y eventos. Por defecto el sistema tiene creado al usuario administrador y será este usuario quien se encargue de agregar los usuarios normales.

5.4.1.2 Base de datos. Base de datos diseñada basados en el modelo entidad relación expuesto en el capítulo anterior, desarrollada en el motor de base de datos PostgreSQL, como medio de acceso la información en la base de datos se utiliza la api jpa (java persistence api) y el uso del orm (mapeo de objeto relacional) hibernate, lo cual nos da un control más fácil sobre los datos y si es llegado el caso cambiar de base de datos sin la necesidad de cambiar el backend creado.

Configuraciones de hibernate para acceder a la base de datos



```

1 <!DOCTYPE hibernate-configuration PUBLIC
2     "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
3     "http://hibernate.org/dtd/hibernate-configuration-3.0.dtd">
4
5 <hibernate-configuration>
6     <session-factory>
7
8         <property name="hibernate.dialect">org.hibernate.dialect.PostgreSQLDialect</property>
9         <property name="hibernate.connection.driver_class">org.postgresql.Driver</property>
10        <property name="hibernate.connection.url">jdbc:postgresql://localhost:5432/proyectoPiscicola</property>
11        <property name="hibernate.connection.username">postgres</property>
12        <property name="hibernate.connection.password">12345</property>
13        <property name="hibernate.c3p0.min_size">5</property>
14        <property name="hibernate.c3p0.max_size">20</property>
15        <property name="hibernate.c3p0.timeout">1800</property>
16        <property name="hibernate.c3p0.max_statements">50</property>
17        <property name="hibernate.dialect">org.hibernate.dialect.PostgreSQLDialect</property>
18        <property name="show_sql">>false</property>
19
20        <mapping class="modelo.persona"/>
21        <mapping class="modelo.perfil"/>
22        <mapping class="modelo.persona_perfil"/>
23        <mapping class="modelo.evento"/>
24        <mapping class="modelo.evento_persona_pecera"/>
25        <mapping class="modelo.indicador"/>
26        <mapping class="modelo.pecera"/>
27        <mapping class="modelo.pescado_pecera"/>
28        <mapping class="modelo.pescado"/>
29        <mapping class="modelo.rango"/>
30        <mapping class="modelo.sensor"/>
31        <mapping class="modelo.solucion_persona"/>
32        <mapping class="modelo.sensor_pecera_pescado"/>
33
34    </session-factory>
35 </hibernate-configuration>

```

Figura 23. Configuración de hibernate

Fuente: Autor del proyecto

5.4.1.3 Arquitectura de la aplicación. La arquitectura utilizada para el desarrollo de la aplicación es el patrón de modelo-vista-controlador, ya que este facilita el desarrollo al separar las aplicaciones en vistas (interfaz del usuario), el modelo y la lógica de control. Su funcionamiento se caracteriza por una vista la cual es una instancia del modelo en un determinado momento, un control recibe un evento, el cual es disparado por un usuario en un determinado momento a través de la interfaz, accede al modelo dependiendo la acción realizada, y presenta en una nueva vista el resultado de dicha acción. Este modelo se basa en el conjunto de objetos que modelan los procesos de negocio que se realizan a través de un sistema. (Alvarez Hernandez , Campos Cantero, & Castelo Delgado, 2011).

A continuación se detalla la aplicación de cada capa del modelo vista controlador al proyecto:

Modelo

Nuestro modelo es desarrollado en java y son las clases que mapea hibernate para acceder a la base de datos, ya que cada clase representa una de las tablas de la misma.

Vista

Las vistas presentes en la aplicación están formadas por los archivos:

CSS: Hojas de estilo de la aplicación además de las desarrolladas para el sistema se han agregado Bootstrap, alertify, jquery calendar y font awesome.

JSP: Reemplaza al HTML, los principales archivos son index y tablero, como complemento se tienen los archivos vista que se irán cargando dependiendo como navegue el usuario.

JavaScript: Lenguaje de programación del lado del cliente, además de los archivos creados se agregó alertify, autotable, Bootstrap, Chartjs, jquery y Popper.

Controlador

El controlador responde a las acciones del usuario y extrae datos del modelo para que sean entregados a la vista, en la aplicación los encargados de realizar esta función son los servlets de java.

5.4.1.4 Diagrama de paquetes. El diagrama de paquetes muestra como está estructurado el sistema y como se relacionan sus paquetes, archivos, clases o sus dependencias entre sí, dejando notar el funcionamiento y estructura del sistema, los diagramas de paquetes son muy variados y pueden representar desde la jerarquía de paquetes hasta la relación funcional existente entre ellos. (Fontela, 2011) En el desarrollo del proyecto se generaron dos diagramas de paquetes.

Diagrama de paquetes

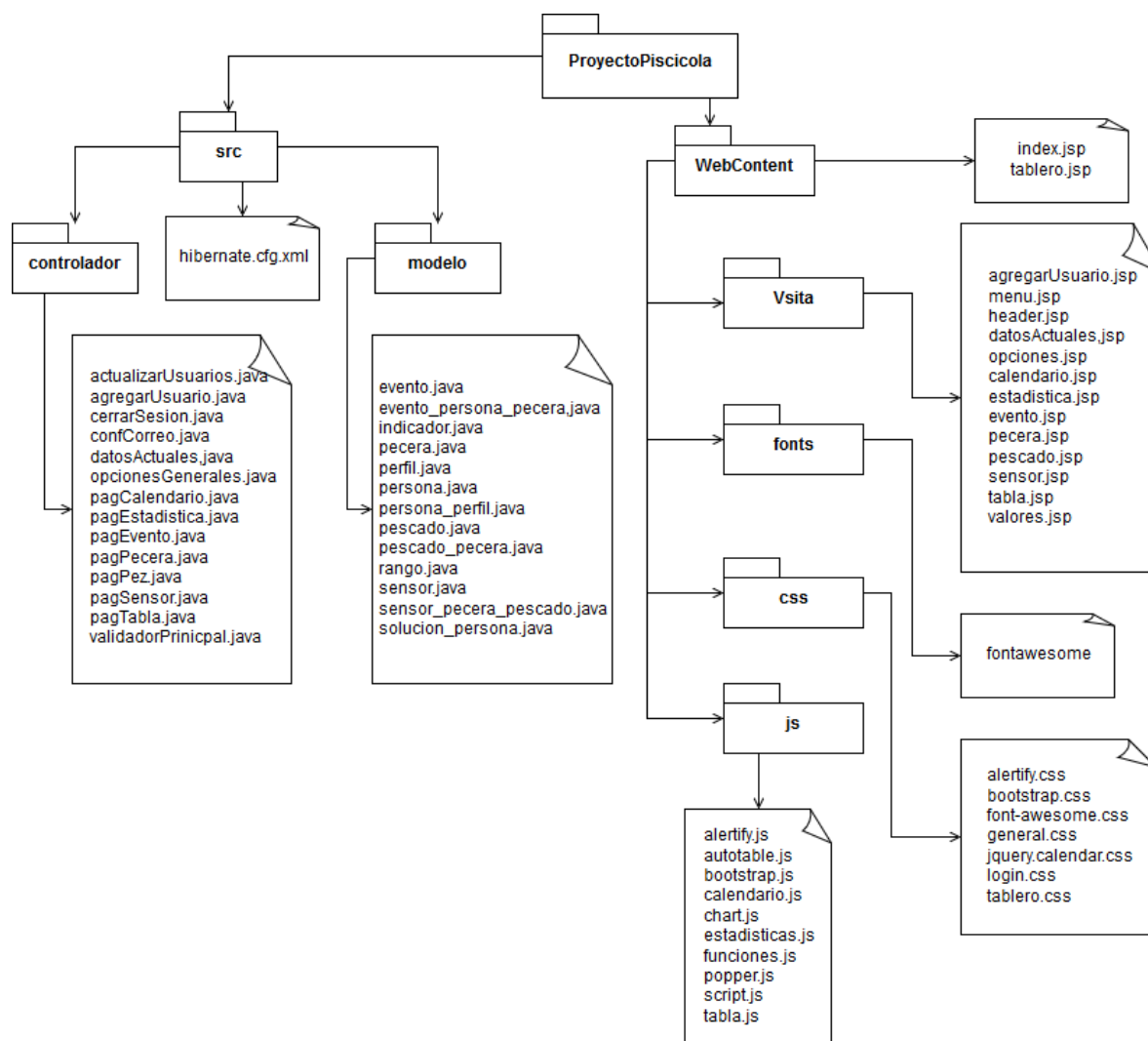


Figura 24. Diagrama de paquetes jerárquico del proyecto

Fuente: Autor del proyecto

Diagrama de paquetes

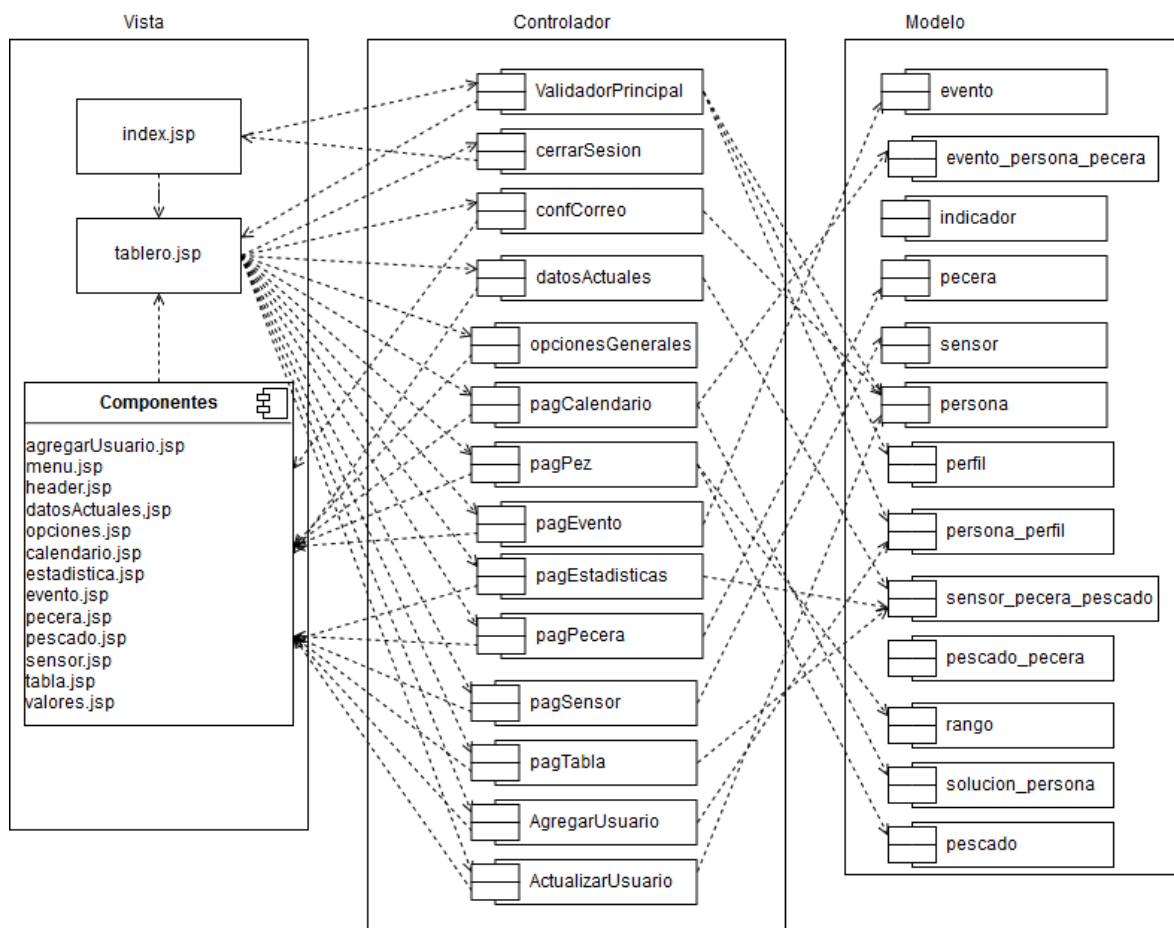


Figura 25. Diagrama de paquetes relación entre paquetes del proyecto

Fuente: Autor del proyecto

5.4.1.5 Interfaz gráfica diseño. Desarrollo de las interfaces basados en los diseños presentados previamente, a continuación se muestra la interfaz de login y la instancia principal del tablero, toda la programación desarrollada para el aplicativo web se encuentra en los anexos del documento.

Interfaz grafica

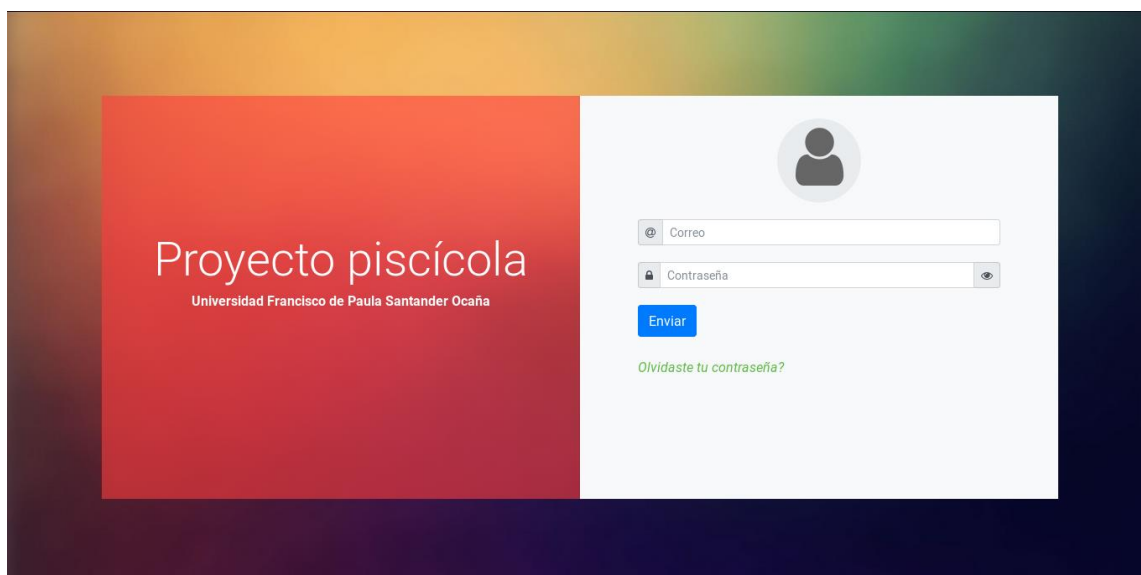


Figura 26. Interfaz desarrollada del login

Fuente: Autor del proyecto

Interfaz grafica



Figura 27. Interfaz principal desarrollada

Fuente: Autor del proyecto

5.4.2 Desarrollo de los dispositivos físicos. Como se había mencionado anteriormente se desea construir una red de sensores para la obtención de los datos, en este apartado se hace uso de las tecnologías seleccionadas en el capítulo 5, para la construcción de la misma.

5.4.2.1 Programación y ensamble de los dispositivos. Se procedió a ensamblar los sensores de medición de cada nodo para generar la red de sensores, inicialmente los nodos encargados de medir estarán compuestos por un Arduino que será quien se encargue de controlar los componentes electrónicos, los cuales serán conformados por los sensores de pH y temperatura mencionados anteriormente y el módulo de radiofrecuencia nfr24l01 para la transferencia de información.

Conexiones de los componentes

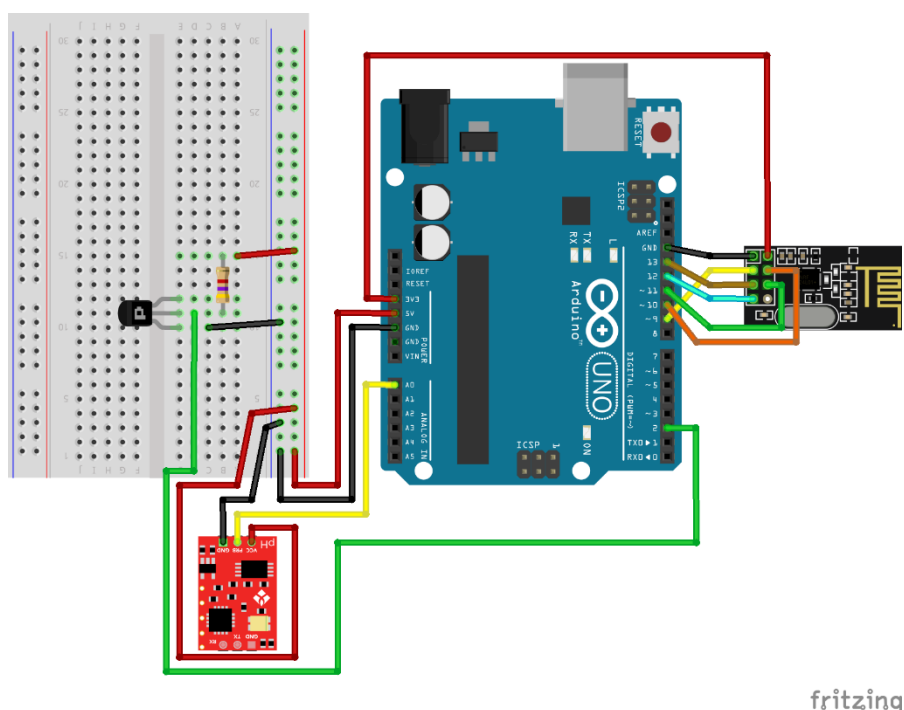


Figura 28. Conexiones de un nodo normal de la red de sensores

Fuente: Autor del proyecto

Conexiones físicas de los componentes

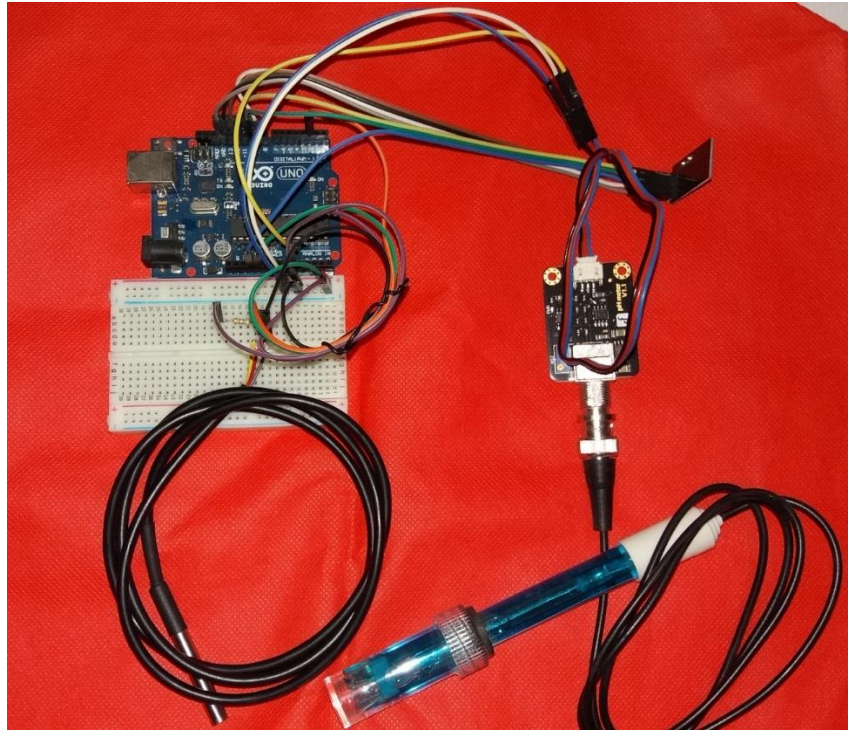


Figura 29. Conexiones físicas de un nodo de la red sensores

Fuente: Autor del proyecto

Y para ensamblarlo se utilizó una caja de proyectos estándar la cual se modificó para que los sensores y las conexiones del Arduino pudiesen salir por un costado.

Caja de proyectos Arduino



Figura 30. Caja de proyectos Arduino

Fuente: Autor del proyecto

La programación de los componentes es desarrollada en el entorno de desarrollo de Arduino y para que la placa Arduino sea capaz de manipular los sensores es necesario importar en su entorno las librerías SPI, RF24 y DallasTemperature. El código programado se trabaja de manera tradicional en el entorno Arduino y puede ser encontrado en los anexos del documento.

Para la programación y configuración del nodo padre el cual sería quien reciba las conexiones de radiofrecuencia se utilizó la Raspberry pi y el lenguaje de programación Python, cabe resaltar que para el uso de módulos en Raspberry pi con Python es necesario la habilitación de los puertos GPIO del mismo.

Conexiones entre el módulo de radiofrecuencia y la Raspberry

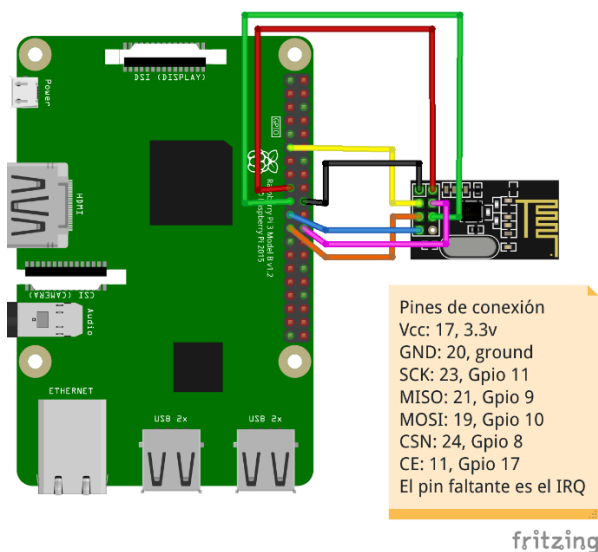


Figura 31. Conexiones de Raspberry pi y módulo NFRL2401

Fuente: Autor del proyecto

Principales características de la red de sensores

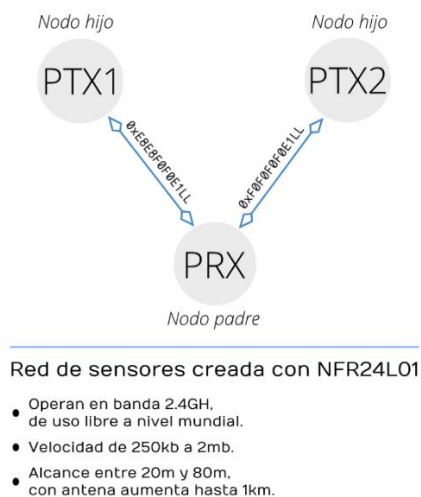


Figura 32. Principales características de red de sensores

Fuente: Autor del proyecto

El dispositivo en físico, haciendo uso de las cajas de protección de la Raspberry



Figura 33. Nodo padre

Fuente: Autor del proyecto

Toda la programación referente a la Raspberry se puede encontrar en los anexo del documento.

5.4.3 Correo electrónico. Entre complementos creados para el proyecto cabe resaltar la creación del correo electrónico proyectopiscicolaufpso@gmail.com el cual es usado para enviar correos y administrar servicios relacionados al proyecto, como lo es el acceso a las plataformas Pushetta y github.

5.4.4 Configuración de Pushetta. Como se había mencionado anteriormente se hace uso de la plataforma Pushetta para notificar al usuario sobre posibles incidencias y problemas presentados, se hace uso de esta plataforma debido a que se puede configurar fácilmente con Python desde la Raspberry pi. Los pasos que se deben seguir para su configuración son:

- Crear cuenta en Pushetta (es un servicio gratuito).

- Crear un canal de transmisión en Pushetta.
- Hacer uso de la llave entregada para la autenticación en el canal y enviar notificaciones.
- Descargar app de Pushetta y suscribirse al canal creado.

El único inconveniente que posee la aplicación es el almacenamiento de las notificaciones, ya que esta solo permite guardarlas en la aplicación por el periodo de un mes, pero que a su vez es suficiente tiempo para que el usuario se informe.

Notificación de la aplicación móvil

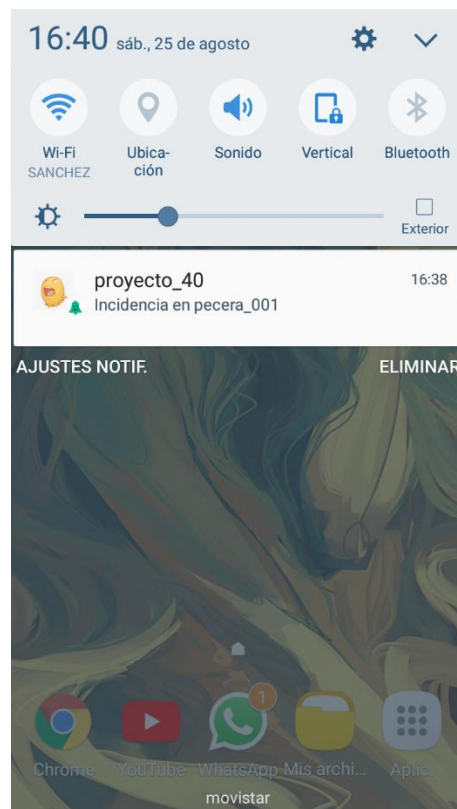


Figura 34. Notificaciones generadas por aplicación Pushetta.

Fuente: Autor del proyecto

La programación empleada para el uso de Pushetta se encuentra en los anexos del documento.

5.4.5 Socket. El desarrollo del socket se da con el propósito de que la Raspberry se pueda conectar al servidor a través de un puerto exclusivo para la transmisión de datos, el cliente es desarrollado en Python y se almacena en la Raspberry, este se ejecuta cada 5 minutos con el fin de transmitir la información de los sensores que lleguen en ese momento analizándola previamente y del lado del servidor el socket es creado en Java, este recibe la información y la almacena en la base de datos, para el desarrollo de socket en el lado del servidor se han creado dos scripts bash (Script que corren en el lenguaje de la consola) con el fin de iniciar el socket y detenerlo. Los códigos tanto de los scripts como del socket se encuentran en el anexo del documento.

5.5 Pruebas

Las pruebas del funcionamiento de los dispositivos se realizaron en un ambiente controlado debido a varios factores.

- Prueba de las diferentes maneras de alimentar los dispositivos.
- El clima templado de la ciudad de Ocaña.
- La estación piscícola no cuenta con conexión a internet.

Para el desarrollo de las pruebas se adquirió un VPS a través de la página Linode, el cual fue utilizado para confirmar el funcionamiento correcto del sistema y la realización de pruebas fuera de una red local que permitiese ver la aplicación corriendo en un entorno real.

Características y configuraciones principales del vps

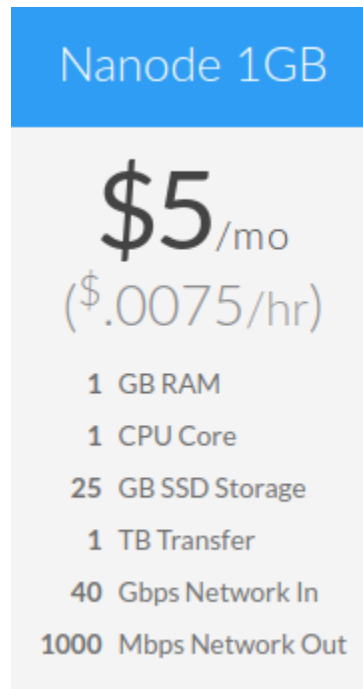


Figura 35. Precio y características del vps

Fuente: (<https://www.linode.com/>)

En las siguientes imágenes se presentan las características del servidor, que nos permite a través de su interfaz configurarlo de manera rápida y nos da la posibilidad de monitorear el estado del mismo a través de su entorno gráfico de esta manera podemos comprobar la escritura en disco de los datos la frecuencia con que trabaja y la posibilidad de detectar errores de una manera más rápida.

Características del vps

Linodes

| Linode | Status | Plan | IP | Location | Backups | Options |
|---------------|---------|-------------|----------------|-----------------|-------------|--|
| ubuntu-server | Running | Linode 1024 | 198.58.113.154 | Dallas, TX, USA | No - Enable | Dashboard Remove |

[Manage Volumes](#) | [Manage Images](#) | [Manage StackScripts](#) | [Add a Linode](#)

This Month's Network Transfer Pool

1% Used | 99% Remaining

1GB Used, 999GB Remaining, 1000GB Quota

Figura 36. Información general del vps

Fuente: (<https://www.linode.com/>)

Características del vps

Dashboard Remote Access Rebuild Rescue Resize Clone Graphs Backups Settings

Linodes » **ubuntu-server**

Dashboard

Select Configuration Profiles Options

My Ubuntu 18.04 LTS Profile (GRUB 2) [Edit](#) | [Remove](#)

[Reboot](#) [Rebuild](#) | [Deploy an Image](#) | [Create a new Configuration Profile](#)

Disks

Ubuntu 18.04 LTS Disk (25344 MB, ext4) [Edit](#) | [Remove](#)

256MB Swap Image (256 MB, swap) [Edit](#) | [Remove](#)

[Create a new Disk](#)

Volumes

[View all Volumes](#) | [Create a new Volume](#)

Host Job Queue (more)

Success System Boot - My Ubuntu 18.04 LTS Profile
Entered: 1 month 4 days ago - Took: 5 seconds

Success System Shutdown
Entered: 1 month 4 days ago - Took: 4 seconds

Success System Boot - My Ubuntu 18.04 LTS Profile
Entered: 1 month 7 days ago - Took: 4 seconds

Success System Shutdown
Entered: 1 month 7 days ago - Took: 4 seconds

Server Status

Your Linode is currently

Running

[Shut down](#)

35 days uptime

Network

- Transfer/mo: 1000 GB
- Incoming: 288 MB
- Outgoing: 218 MB
- Total: 506 MB

You have used 0% of your monthly transfer

Storage

- Total: 25600 MB
- Used: 25600 MB
- Free: 0 MB

You have allocated 100% towards disk images

Backups

No - [Enable](#)

Host

Figura 37. Panel de configuraciones del vps

Fuente: (<https://www.linode.com/>)

5.5.1. Configuración y acondicionamiento del vps. La configuración del vps inicia con la instalación de Tomcat en su versión 8.5, PostgreSQL en su versión 10 y el JVM de java en su versión 8.0, luego de esto se configura Tomcat para que este permita conexiones remotas a su manager y se utiliza el ufw para permitir conexión solo a los puertos utilizados, denegando el acceso a todos los demás puertos del sistema, después de esto solo nos queda el uso del manager de Tomcat para subir nuestra aplicación y subir la base de datos a través del comando:

```
Psql -U postgres -h localhost -d proyectoPiscicola -f backup.sql
```

Apache Tomcat

The screenshot shows the Tomcat Web Application Manager interface. At the top, there is a message box with the text "Message: OK". Below this is a navigation bar with links for "List Applications", "HTML Manager Help", "Manager Help", and "Server Status". The main content area is titled "Applications" and contains a table with the following data:

| Path | Version | Display Name | Running | Sessions | Commands |
|---------------|----------------|---------------------------------|---------|----------|--|
| / | None specified | proyectoPiscicola | true | 1 | Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes |
| /docs | None specified | Tomcat Documentation | true | 0 | Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes |
| /examples | None specified | Servlet and JSP Examples | true | 0 | Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes |
| /host-manager | None specified | Tomcat Host Manager Application | true | 0 | Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes |
| /manager | None specified | Tomcat Manager Application | true | 1 | Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes |

Below the table is a "Deploy" section with the text "Deploy directory or WAR file located on server".

Figura 38. Aplicación manager de Tomcat

Fuente: Apache Tomcat

Apache Tomcat

| Server Information | | | | | | | |
|----------------------|---------------|--------------------|---------|-------------------|-----------------|-----------|------------|
| Tomcat Version | JVM Version | JVM Vendor | OS Name | OS Version | OS Architecture | Hostname | IP Address |
| Apache Tomcat/8.5.32 | 1.8.0_181-b13 | Oracle Corporation | Linux | 4.15.0-20-generic | amd64 | localhost | 127.0.0.1 |

Figura 39. Información del servidor

Fuente: Apache Tomcat

5.5.2 Puesta en marcha del socket. Para la recepción de datos desde la red de sensores, como se mencionó en el capítulo anterior se hizo uso de un socket, su configuración y puesta en marcha se encuentra en los anexos del proyecto.

5.5.3. Inicialización de los dispositivos. Los dispositivos son inicializados luego de que el servidor es puesto en marcha y se ha asegurado de que Tomcat y el socket están en ejecución, ya que si este es inicializado antes saltara un error debido a que la Raspberry pi no encontrara el servidor. De esta misma manera es el proceso de inicialización para la red de sensores, primero de iniciar el nodo padre y luego los demás nodos. También tenemos que asegurarnos de que la Raspberry pi está conectada a internet preferiblemente por su puerto de Ethernet.

5.5.4 Frecuencia de toma de datos. La frecuencia de toma de datos se da cada 5 minutos debido a que a los sensores les toma cierto tiempo, nivelarse antes de comenzar a dar valores normales, esta frecuencia puede aumentar algunos segundos si se presentan incidencias. A través de la gráfica de uso de CPU del vps se evidencia que esta frecuencia de información es soportada por él.

Rendimiento CPU del vps

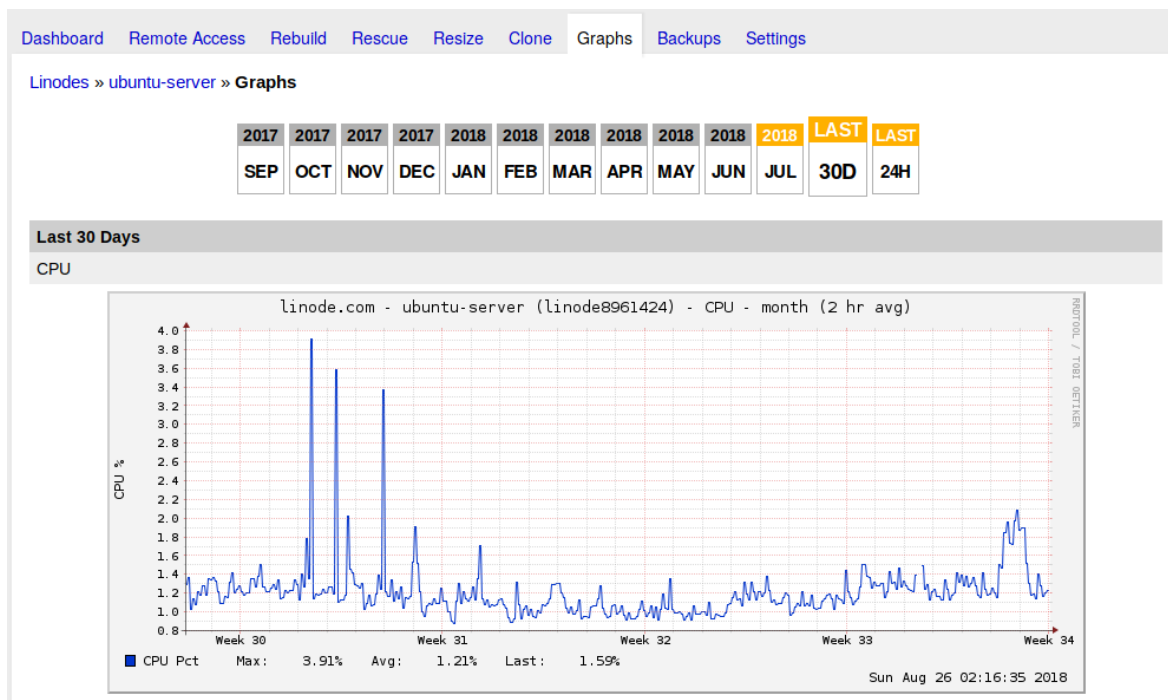


Figura 40. Rendimiento CPU del vps

Fuente: (<https://www.linode.com/>)

Como resultado tenemos que la web trabaja de manera correcta, permitiéndonos manipular la información obtenida por los sensores y cumpliendo con su cometido cumpliendo con el dominio de aplicación de IoT, sin embargo en el dominio de dispositivos no se hicieron pruebas de grandes periodos de tiempo dejando una incertidumbre de como responda el sistema en días enteros o semanas completas de conexión, esto debido a que en el desarrollo del sistema se utilizó un sensor de pH básico, por fácil disposición y este sensor no puede ser expuestos a periodos largos de toma de valores debido a que pierde su calibración, ya que este sensor está pensado para mediciones puntuales, estas pruebas de periodos largos serian una próxima etapa antes de pensar en el despliegue del sistema. Como resultado de las pruebas de alimentación del dispositivo se rectifica de que cualquier panel solar que se capaz de entregar más de 5v es capaz

de energizar el sistema de los dispositivos, sin embargo como la zona en la que se cuenta Ocaña es una zona templada no se recomienda la alimentación por energía solar como principal medio de alimentación del sistema, en esta caso es recomendable hacer uso de la red eléctrica de la universidad, para que el sistema tenga una alimentación estable y evitar incidencias falsas o caídas del sistema por falta de energía. En las pruebas realizadas se utilizó como cubierta del dispositivo las cajas de proyecto base de Arduino, las cuales le brindan protección al sistema pero no están hechas pensadas para exteriores, por lo tanto no es recomendable su uso.

Como conclusión final de las pruebas tenemos que el sistema es funcional pero se deben hacer pruebas fuera de los parámetros planteados para asegurar su completo funcionamiento en todos los escenarios posibles.

En los anexos del sistema se presentan la aplicación en ejecución.

5.6 Despliegue

El despliegue de aplicación se daría basados en la sección descritas anteriormente, siguiendo los pasos presentes en las pruebas, se debe asegurar puntualmente que la aplicación cuente con alojamiento propio para asegurar que esta sea la única que hace uso de los recursos de la maquina en la que se encuentra, además se debe asegurar el acceso de la Raspberry pi a internet, a través de su puerto Ethernet, para asegurar la conexión constante del dispositivo y la configuración remota del mismo, también se recomienda la alimentación de los dispositivos a través de la red eléctrica de la universidad, el despliegue del proyecto tardaría dependiendo de la disponibilidad de los recursos.

Cada pecera contaría con un nodo de la red sensores que se encargaría de tomar los datos de esa peceras, en la torre que posee la estación piscícola se colocaría la Raspberry pi con su módulo de radiofrecuencia apuntando hacia las peceras, de esta manera aseguramos que la información de los sensores está siendo direccionada hacia el nodo padre.

6. Conclusiones.

6.1. Conclusiones generales.

- El resultado de la pruebas fue el esperado en un ambiente de correcto funcionamiento para el sistema, obteniendo una página web funcional y dispositivos que responde a la toma de valores.
- La alimentación de los dispositivos puede darse de diferentes maneras, pero el uso de paneles solares sin el uso de baterías genera inconvenientes y bajas en la alimentación de los dispositivos haciendo que pueda desencadenar en incidencias falsas.
- El uso de una caja estándar de proyectos Arduino no es lo recomendable para este tipo de proyectos, ya que los dispositivos van a estar al aire libre y debido a que el material (plástico básico de impresión ABS) en que están hechos es muy sensible a daños por rayos UV.

6.2. Conclusiones por objetivos.

- La creación de un sistema de monitoreo a través de sensores es posible siempre y cuando se tenga claro el cómo suplir al sistema con internet para los dispositivos que así lo requieran y la debida administración de su alimentación.
- La selección de tecnologías y metodologías para dar solución al problema presente puede variar dependiendo el desarrollador o grupo de desarrollo.

7. Recomendaciones

Como recomendaciones se deberá asegurar que no se instalaran ningún otro dispositivo que trabaje con radiofrecuencia que pueda opacar la frecuencia de los nodos de la red de sensores y que la cubierta de los dispositivos físicos no sea de un estándar de proyectos Arduino, ya que estos por lo general no están pensados para espacios libres.

Referencias

- Alvarez Hernandez , J. M., Campos Cantero, P., & Castelo Delgado, T. (2011). Desarrollo de una aplicación Web para la gestión de entornos virtuales. Madrid, España.
- AUNAP. (s.f.). MARCO LEGAL E INSTITUCIONAL DE LA PESCA. Colombia.
- Arduino. (2005). *Arduino*. Obtenido de <https://www.arduino.cc/>
- Alvarez Hernandez , J. M., Campos Cantero, P., & Castelo Delgado, T. (2011). Desarrollo de una aplicación Web para la. Madrid, España.
- Espeso, P. (04 de abril de 2016). *Raspberry Pi, el ordenador perfecto para educación*. Obtenido de educación 3.0: <https://www.educaciontrespuntocero.com/noticias/raspberry-pi-educacion/34377.html>
- Bakinter. (2011). *El Internet de las Cosas*. Accenture.
- BITAG. (2016). *Internet of Things (IoT) Security and Privacy Recommendations A BROADBAND INTERNET TECHNICAL ADVISORY GROUP TECHNICAL WORKING GROUP REPORT*. BITAG. Obtenido de <https://www.bitag.org/report-internet-of-things-security-privacy-recommendations>.
- Benaissa, S., Joseph, W., Plets, D., Tanghe, E., Trogh, J., Martens, L., . . . Sonck, B. (2017). Internet of animals: Characterisation of LoRa sub-GHz off-body wireless channel in dairy barns. *The institution of engineering and technology*.
- Borgia, E. (2014). The Internet of Things vision: Key features, applications and open issues.

El tiempo. (13 de 9 de 2013). *El tiempo*. Obtenido de Un mal ambiente en el acuario genera estrés en los peces: <http://www.eltiempo.com/archivo/documento/CMS-13086755>

Elacuarista. (2010). *Elacuarista*. Obtenido de Obtenido de Efectos del ph sobre el organismo: <http://www.elacuarista.com/secciones/pH.html>

eldiario. (2013). *eldiario*. Obtenido de Que es el Hadware libre: http://www.eldiario.es/turing/Hardware-Libre_0_139986451.html

Espeso, P. (04 de abril de 2016). *Raspberry Pi, el ordenador perfecto para educación*. Obtenido de educacion 3.0: <https://www.educaciontrespuntocero.com/noticias/raspberry-pi-educacion/34377.html>

Estanques y peces. (2002). *Estanques y peces*. Obtenido de Relación temperatura / ánimo del pez: https://www.estanquesypeces.com/peces/relacion_temperatura_animos.html

Esteban Sanchis, A. (2016). *Implementación y desarrollo de una red de sensores inalámbricas con almacenamiento de datos en la nube*.

Flores Rosas, M. I., & Martínez Escobedo, K. (2005). *Sistema integral de gestión empresarial con aplicación de workflows*. Mexico.

Fontela, C. (2011). *UML modelado de software para profesionales*. Alfaomega.

Guitierrez, P. (2016). *IoTHUB*. Obtenido de <https://www.iothub.com.au/news/how-iot-is-helping-tasmanias-oyster-industry-436397>

Hernández, J. (2014). *Analisis y desarrollo web*. (2. Jesús Hernández, Ed.)

- Higuera, H., & Gonzales, J. (2006). Evaluación del kit fish farming FF1A para analisis de calidad de agua en acuicultura. *Revista de la facultad de medicina veterinaria y zootecnia*, 116-124.
- IBM. (2002). *eweek*. Obtenido de IBM Acquires Rational: <http://www.eweek.com/pc-hardware/ibm-acquires-rational>
- Jamieson, P., & Herdtner, J. (2015). More missing the Boat Arduino, Raspberry Pi, and small prototyping boards and engineering education needs them. *IEEE Frontiers in Education Conference*.
- Kaloxyllos, A., Wolfert, J., Verwaart, T., Maestre Terol, C., Brewster, C., Robbmond, R., & Sundmaker, H. (2013). The Use of Future Internet Technologies in the Agriculture and Food Sectors: Integrating the Supply Chain. *Procedia Technology*, 51-60.
- Kendall, k., & Kendall, J. (2005). *Analisis y diseño de sistemas*. Mexico: Educacion.
- Liñan Colonia, A., Vives, A., Bagula, A., Zennaro, M., & Pietrosemoli, E. (2015). *Internet de las cosas*. Internet Society.
- López, A. T. (2014). *Seguridad en el internet de las cosas*. Innovatech.
- Martinez Valencia, J. L. (2016). *DISEÑO Y DESARROLLO DE UN SISTEMA DE MEDICION Y MONITOREO REMOTO ENFOCADO AL USO DOMESTICO*. pereira.
- Maurel, V. B., & Huyghe, C. (2017). Putting agricultural equipment and digital technologies at the cutting edge of agroecology. *ASP Sciences*.

- NLS, V. (2015). *dealstreetasia*. Obtenido de <https://www.dealstreetasia.com/stories/indonesia-based-iot-startup-efishery-raises-undisclosed-series-a-round-from-aqua%C2%ADspark-ideosource-11818/>
- Robles, I., & Acosta, A. (2016). *IPv6 y el Internet de las Cosas*. Cisco.
- Rose, K., Eldridge, S., & Chapin, L. (2015). *La internet de las cosas - Breve reseña*. Internet Society.
- Ruan, J., & Shi, Y. (2016). Monitoring and assessing fruit freshness in IOT-based e-commerce delivery using scenario analysis and interval number approaches. *Elseiver*.
- Raspberry pi. (2012). *Raspberry pi*. Obtenido de <https://www.raspberrypi.org/>
- Simon, J., Blet, N., Bender, C., Recanzone, R., Sosa, J., & Torres, A. (2013). Sobre la utilización de sistemas embebidos para la enseñanza de la programación en una carrera de ingeniería electrónica. *IV Congreso Microelectrónica Aplicada* .
- Tessel. (2013). *Tessel 2*. Obtenido de <https://tessel.gitbooks.io/t2-docs/content/>
- Vidyasagan, S., Devi, R., Varma, A., Charan, H., & Rajesh, A. (2017). A low cost IoT based crowd management system for public transport. *2017 International Conference on Inventive Computing and Informatics (ICICI)*.
- Su, X., Zhang, H., Rieki, J., Keranen, A., Nurminen, J., & Du, L. (2014). Connecting IoT Sensors to Knowledge-Based Systems by Transforming SenML to RDF. *Procedia Computer Science*, 215-222.
- Wolfert, S., Ge, L., Verdouw, c., & Bogaardt, M.-J. (2017). Big Data in Smart Farming – A review. *Agricultural Systems*, 69-80.

Apéndices

Apéndice A Configuración de Raspberry

Configuraciones a tener en cuenta antes de poner en producción la Raspberry pi.

- Actualizar hora y fecha local con comando **date** o hacer uso de los servidores NTP instalando con los comandos **sudo apt install ntp** y habilitar el servicio con **sudo systemctl enable ntp**.
- Deshabilitar entorno gráfico, desde la consola ejecutamos **sudo raspi-config** y seleccionamos la opciones de BOOT (opción 3 generalmente) y luego seleccionamos la opción Desktop / CLI, esto se hace con el fin de que la Raspberry consuma menos recursos cuando esta esté en ejecución.
- Para usar el módulo NFRL2401 se debe configurar le módulo SPI en la Raspberry pi.

Apéndice B Códigos programados del proyecto

Toda la programación realizada se encuentra en la cuenta de github del correo presentado en la sección 7.3, las credenciales son las mismas para el correo y Pushetta.

- Correo: proyectopiscicolaufpso@gmail.com
- Usuario: proyectopiscicolaufpso

Apéndice C Diseños de interfaces

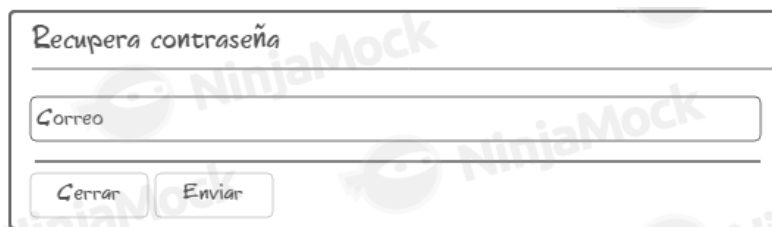
Diseño de interfaz de la página principal login



Figura 41. Interfaz de login que da acceso a todas las funcionalidades del sistema

Fuente: Autor del proyecto

Diseño de interfaz de la página principal login



Recupera contraseña

Correo

Cerrar Enviar

Figura 42. Modal recuperar contraseña de la interfaz de login

Fuente: Autor del proyecto

Diseño de interfaces para usuario autenticado ante el sistema, si el usuario es un operario las opciones de administrador no serán mostradas.

Instancia inicial del usuario luego de su autenticación

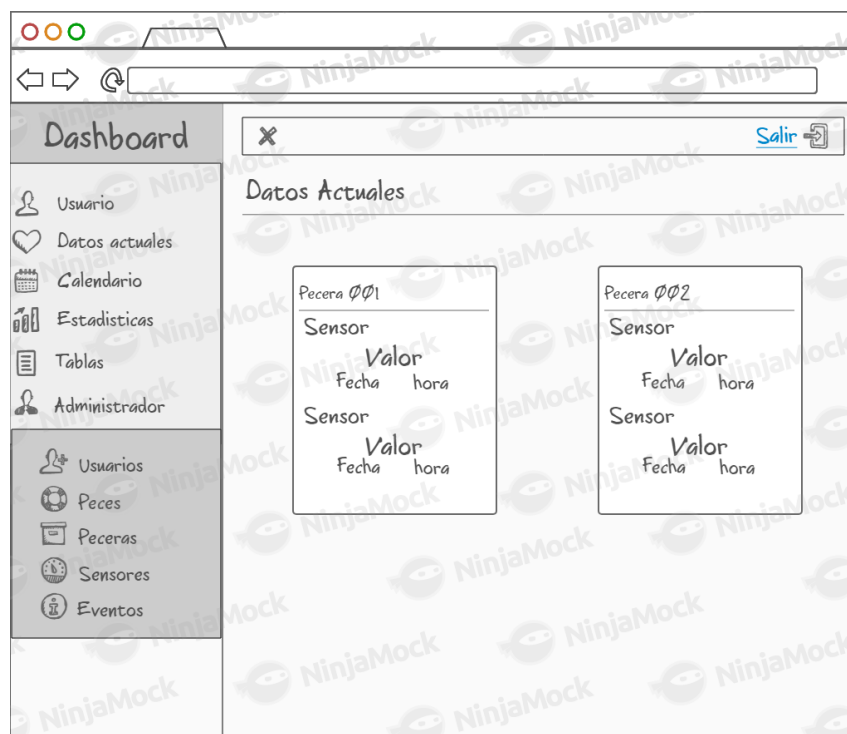


Figura 43. Interfaz datos actuales

Fuente: Autor del proyecto

Interfaz donde el usuario podrá actualizar su información personal

The image shows a web browser window displaying a user profile update page. The browser's address bar is empty. The page has a header with a close button (X) and a 'Salir' (Logout) button with an external link icon. The main content area is titled 'Usuario' and 'Cargo'. It contains several form fields: 'Nombre' and 'Apellido' (Name and Surname), gender selection with radio buttons for 'Masculino' (selected) and 'Femenino', 'Fecha de nacimiento' (Date of Birth), 'Contraseña' (Password), 'Correo' (Email), and 'Documento' (Document). An 'Actualizar' (Update) button is located at the bottom of the form. On the left side, there is a 'Dashboard' sidebar with a menu of options: 'Usuario', 'Datos actuales', 'Calendario', 'Estadísticas', 'Tablas', 'Administrador', 'Usuarios', 'Peces', 'Peceras', 'Sensores', and 'Eventos'. The entire interface is overlaid with a 'NinjaMock' watermark.

Figura 44. Interfaz de usuario

Fuente: Autor del proyecto

Interfaz donde se podrán ver las incidencias, soluciones y eventos creados

The image shows a web browser window displaying a dashboard. The browser's address bar is empty. The dashboard has a sidebar on the left with the title "Dashboard" and several menu items: "Usuario", "Datos actuales", "Calendario", "Estadísticas", "Tablas", and "Administrador". Below these are icons for "Usuarios", "Peces", "Peceras", "Sensores", and "Eventos". The main content area is titled "Calendario" and features a "Salir" button. Below the title is a date input field showing "dd-mm-yy" and a "Mes" label. The calendar grid shows the following data:

| Domingo | Lunes | Martes | Miércoles | Jueves | Viernes | Sábado |
|---------|-------|--------|-----------|--------|---------|--------|
| | | 1 | 2 | 3 | 4 | 5 |
| 6 | 7 | 8 | 10 | 11 | 12 | 13 |
| 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 21 | 22 | 23 | 24 | 25 | 26 | 27 |
| 28 | 29 | 30 | 31 | | | |
| | | | | | | |

Figura 45. Interfaz de calendario

Fuente: Autor del proyecto

Modal de calendario

| Persona que lo creó | opciones |
|----------------------|----------|
| Nombre de la persona | |
| Nombre de la persona | |

Figura 46. Modal Lista de Eventos/Soluciones de la interfaz calendario

Fuente: Autor del proyecto

Modal de calendario

Agregar/actualizar Evento


Evento

Observación

Figura 47. Modal Agregar/actualizar evento de la interfaz calendario

Fuente: Autor del proyecto

Modal de calendario



Modal de calendario para agregar/actualizar solución. El modal tiene un título "Agregar/actualizar Solución". Debajo del título hay un campo de texto etiquetado "Acción". Debajo de ese campo hay otro campo de texto etiquetado "Observación". En la parte inferior del modal hay dos botones: "Cerrar" y "Enviar".

Figura 48. Modal Agregar/actualizar solución de la interfaz calendario.

Fuente: Autor del proyecto

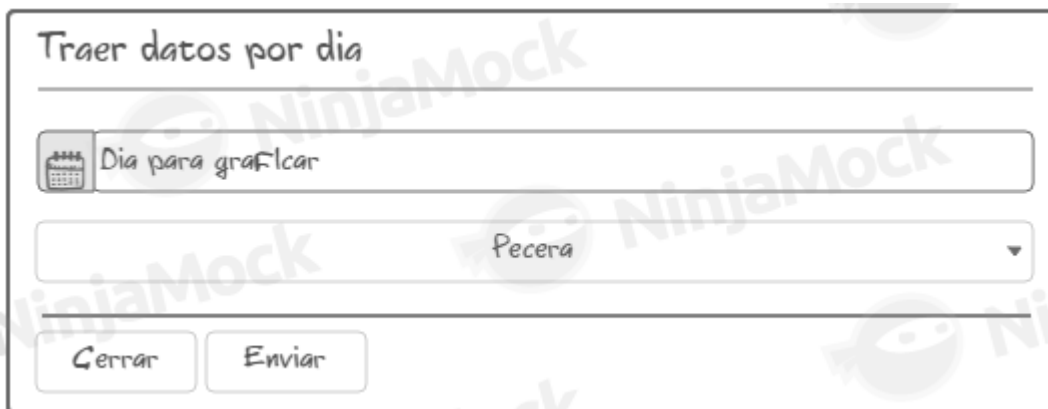
Interfaz donde se podrán ver estadísticas por pecera



Figura 49. Interfaz estadística

Fuente: Autor del proyecto

Modal de interfaz estadística

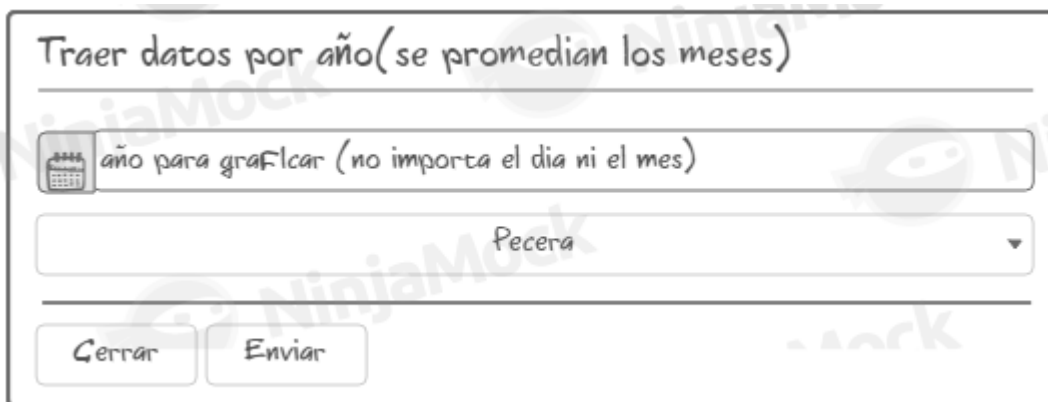


Traer datos por día

Figura 50. Modal traer datos por día de la interfaz estadística

Fuente: Autor del proyecto

Modal de interfaz estadística

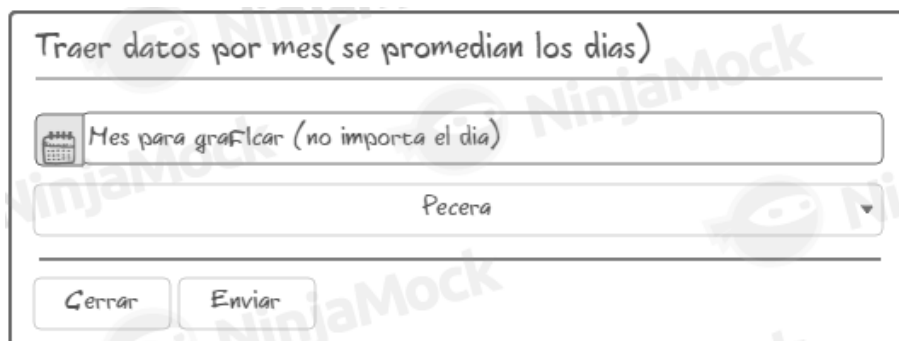


Traer datos por año (se promedian los meses)

Figura 51. Modal traer datos por mes de la interfaz estadística.

Fuente: Autor del proyecto

Modal de interfaz estadística



Modal de interfaz estadística con el título "Traer datos por mes (se promedian los días)".

Contiene un campo de texto con un ícono de calendario y el texto "Mes para graficar (no importa el día)".

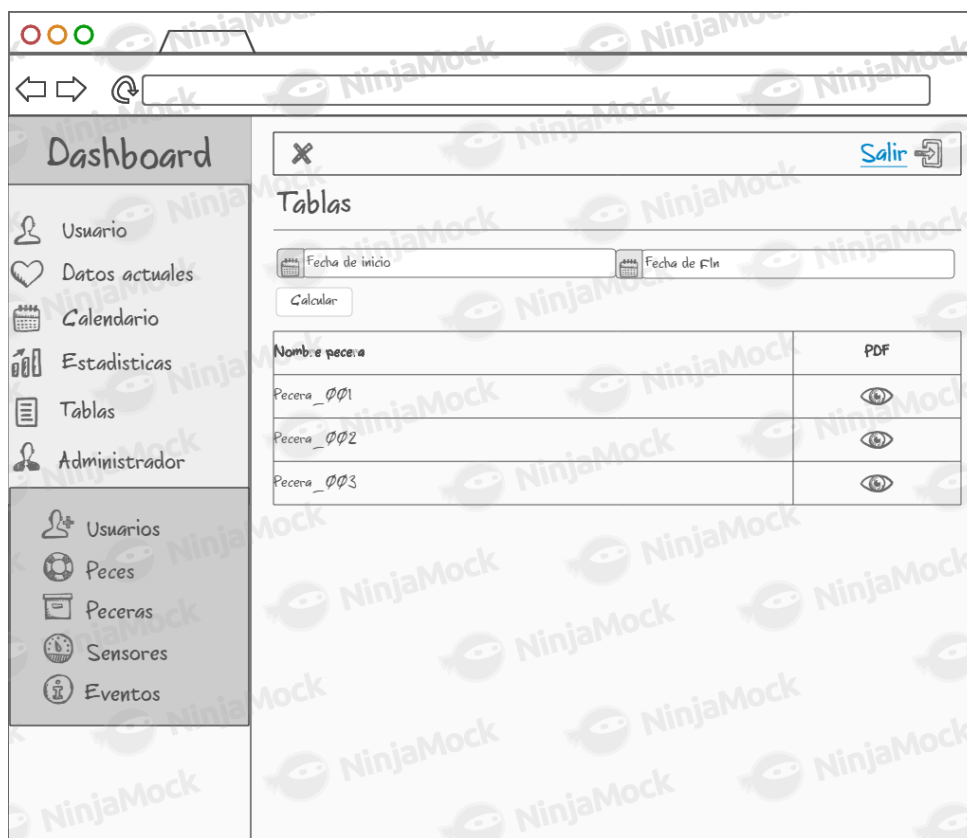
Debajo del campo de texto hay un menú desplegable que muestra "Pecera".

En la parte inferior del modal hay dos botones: "Cerrar" y "Enviar".

Figura 52. Modal traer datos por año de la interfaz estadística

Fuente: Autor del proyecto

Interfaz para ver información por periodos



Interfaz de usuario para ver información por periodos. Incluye un menú de navegación a la izquierda y un panel principal con un formulario de búsqueda y una tabla de resultados.

Menú de navegación:

- Dashboard
- Usuario
- Datos actuales
- Calendario
- Estadísticas
- Tablas
- Administrador
- Usuarios
- Peces
- Peceras
- Sensores
- Eventos

Panel principal:

Título: Tablas

Formulario de búsqueda:

Fecha de inicio: Fecha de fin:

Botón: Calcular

Tabla de resultados:

| Nombre pecera | PDF |
|---------------|-----|
| Pecera_001 | |
| Pecera_002 | |
| Pecera_003 | |

Figura 53. Interfaz tablas

Fuente: Autor del proyecto

Interfaces para el administrador, en esta el administrador podrá crear y editar usuarios, peces, peceras, sensores y eventos.

Interfaz gestión de usuarios

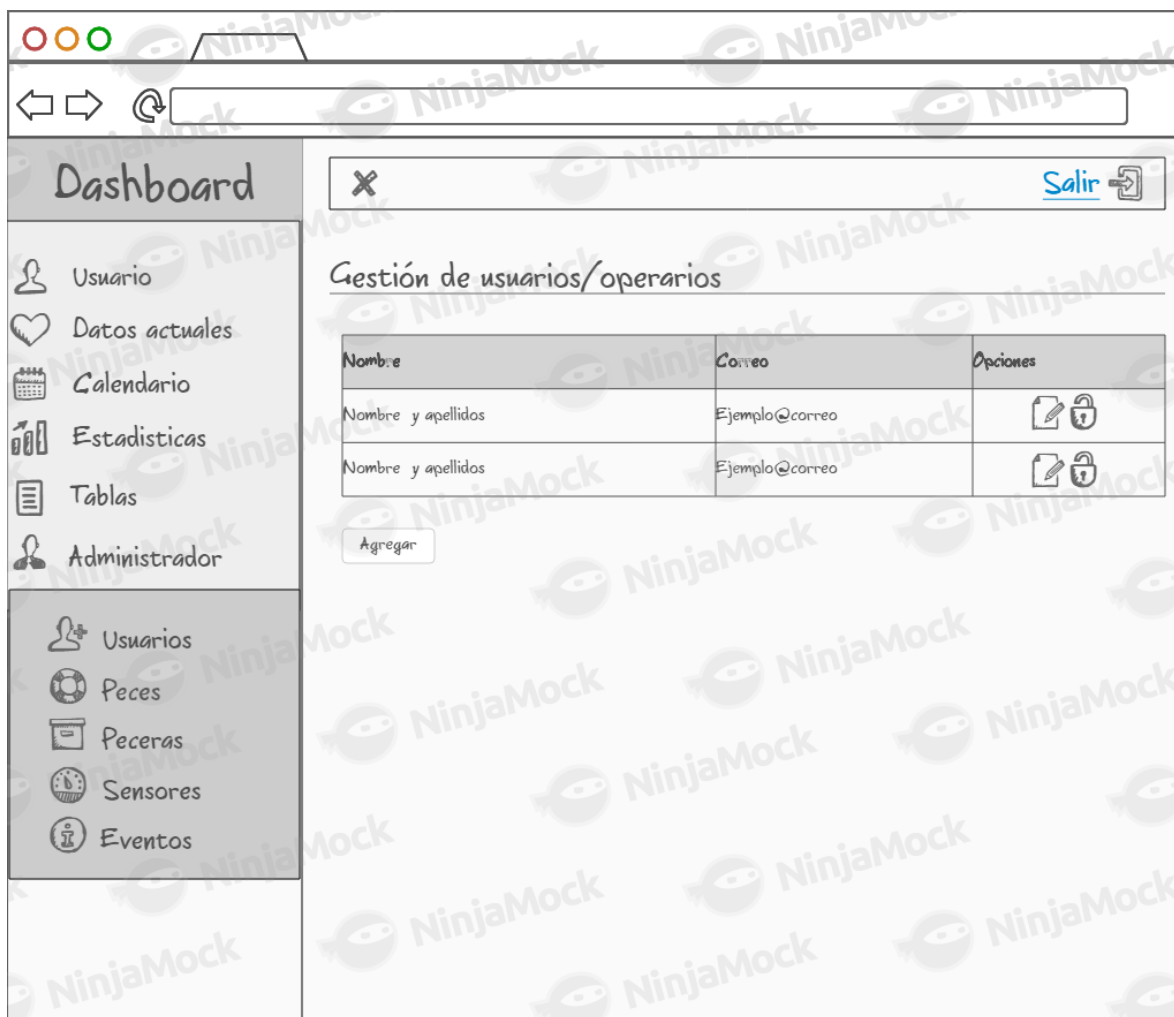
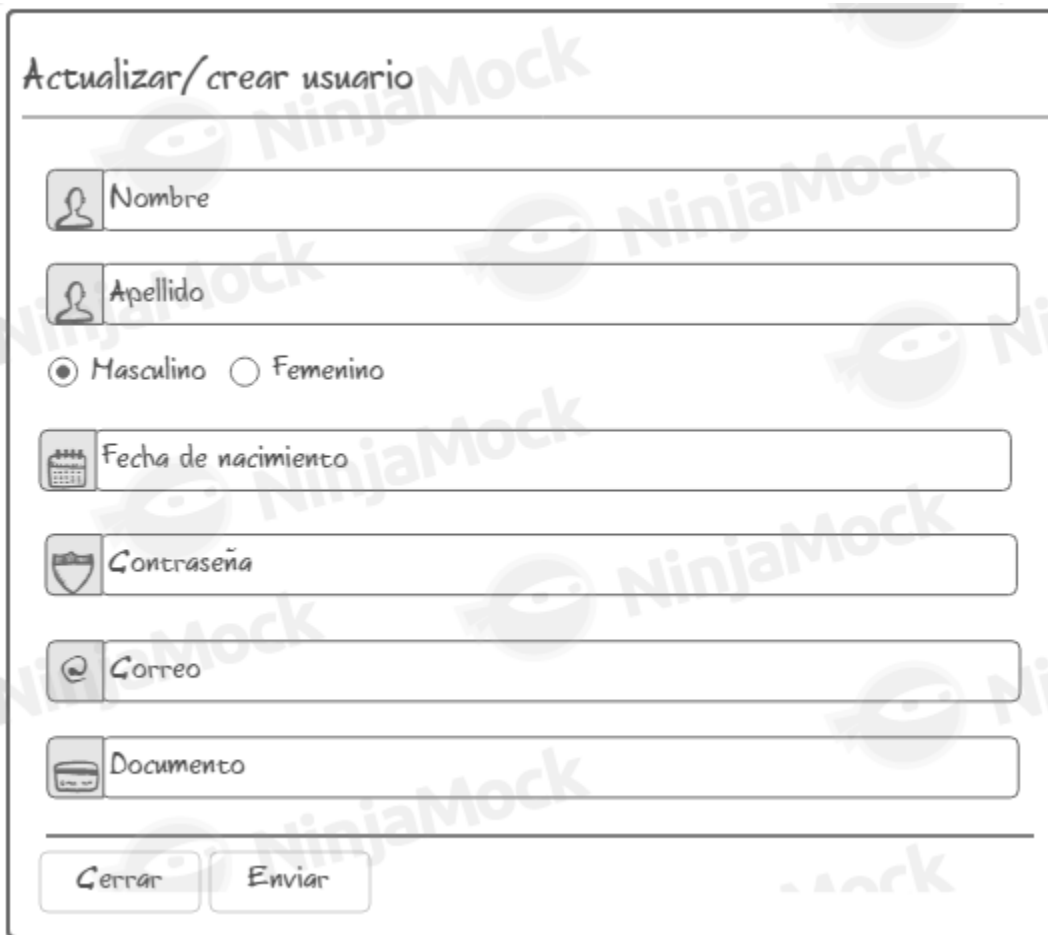


Figura 54. Gestión de usuarios, en esta interfaz el usuario podrá crear y editar usuarios, no podrá eliminar usuarios solo se podrá cambiar su estado de activo a inactivo

Fuente: Autor del proyecto

Modal gestión usuarios



Actualizar/crear usuario

Nombre

Apellido

Masculino Femenino

Fecha de nacimiento

Contraseña

Correo

Documento

Cerrar Enviar

The image shows a modal window for user management. It has a title bar 'Actualizar/crear usuario'. Below the title bar are several input fields: 'Nombre' (with a person icon), 'Apellido' (with a person icon), a gender selection with radio buttons for 'Masculino' (selected) and 'Femenino', 'Fecha de nacimiento' (with a calendar icon), 'Contraseña' (with a shield icon), 'Correo' (with an email icon), and 'Documento' (with a document icon). At the bottom, there are two buttons: 'Cerrar' and 'Enviar'.

Figura 55. Modal actualizar/crear usuario de la interfaz gestión de usuarios

Fuente: Autor del proyecto

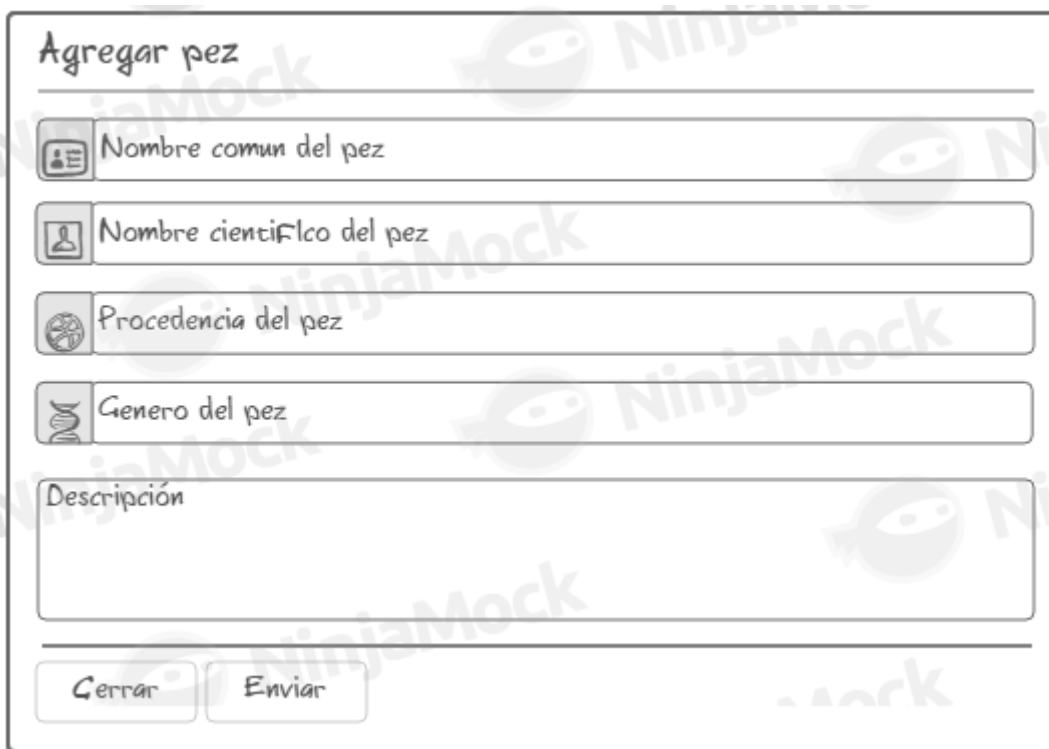
Interfaz peces

The image shows a web interface for managing fish. On the left is a sidebar menu with the following items: Usuario, Datos actuales, Calendario, Estadísticas, Tablas, and Administrador. Below these are icons for Usuarios, Peces, Peceras, Sensores, and Eventos. The main content area is titled 'Dashboard' and contains a form for adding or updating a fish. The form includes a dropdown for 'Pez', followed by input fields for 'Nombre común del pez', 'Nombre científico del pez', 'Procedencia del pez', and 'Genero del pez'. There is a large text area for 'Descripción'. Below the description are 'Actualizar' and 'Agregar Nuevo' buttons. The form also includes a 'Rango Estable' section with 'Valor minimo' and 'Valor maximo' fields and an 'Actualizar' button. At the bottom, there is a 'Rango indicadores' table with columns for 'Minimo' and 'Maximo', and a 'Header: 3' section with a table structure and an 'Agregar indicador' button.

Figura 56. Interfaz en la cual el usuario podrá crear y editar peces, por defecto el pez estará asociado a todos los sensores, si el pez es creado sus valores de rango estable serán dados por defecto por lo cual estos deben ser modificados, los indicadores son usados como referencia para los casos de incidencia.

Fuente: Autor del proyecto

Modal interfaz peces



Agregar pez

Nombre comun del pez

Nombre cientifico del pez

Procedencia del pez

Genero del pez


Descripción

Cerrar Enviar

Figura 57. Modal agregar pez de interfaz peces

Fuente: Autor del proyecto

Modal interfaz peces



Agregar/editar indicador

▼ Valor minimo ▲ Valor maximo

Descripción

Cerrar Enviar

Figura 58. Modal agrega/editar indicador de la interfaz peces

Fuente: Autor del proyecto

Interfaz peceras

The image shows a web application interface for managing fish tanks. On the left is a sidebar menu with the following items: Usuario, Datos actuales, Calendario, Estadísticas, Tablas, and Administrador. Below these are icons for Usuarios, Peces, Peceras, Sensores, and Eventos. The main content area is titled 'Peceras' and contains a form for creating or editing a tank. The form includes a dropdown menu for 'Peceras', a radio button for 'Estado' (Active/Inactive), input fields for 'Diametro de la pecera' and 'Profundidad de la pecera', a text area for 'Descripción', and buttons for 'Actualizar' and 'Agregar nuevo'. Below the form is a table with the following structure:

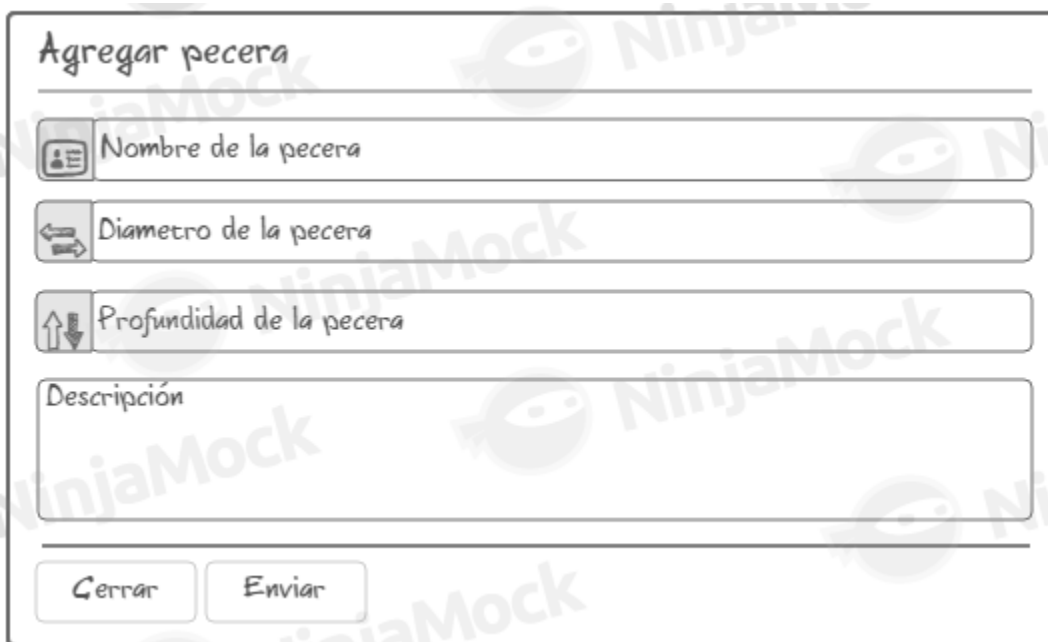
| Pescado | Cantidad | Fecha inicio | Fecha Fin |
|--------------------|----------|--------------|-------------------|
| Nombre del pescado | numero | dd-mm-yy | dd-mm-yy |
| Nombre del pescado | numero | dd-mm-yy | Agregar fecha Fin |

At the bottom of the table is an 'Agregar' button.

Figura 59. Interfaz donde se podrán crear y editar peceras, además de poder asociarles los peces que esta tendrá y el intervalo de tiempo en el cual estuvo

Fuente: Autor del proyecto

Modal peceras



Este modal, titulado "Agregar pecera", permite configurar una nueva pecera. Incluye los siguientes campos de entrada:

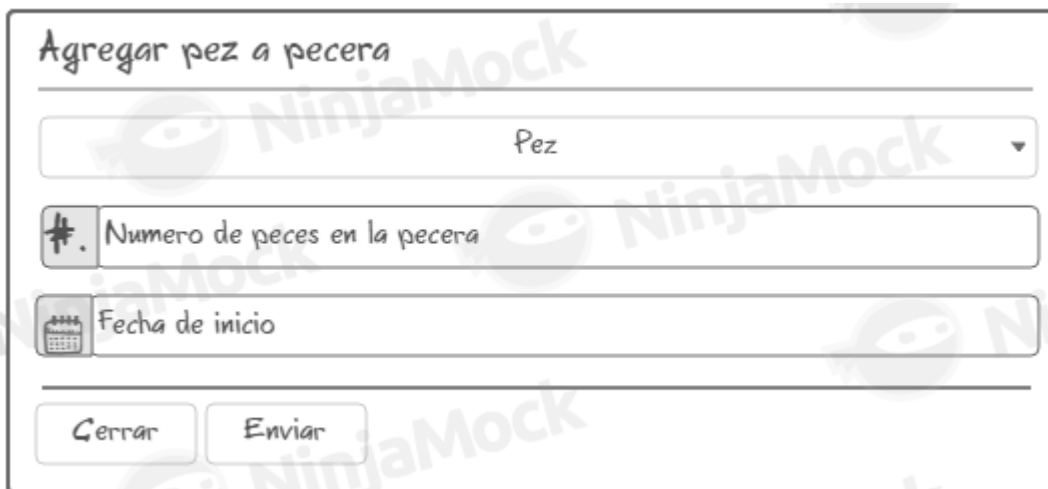
- Nombre de la pecera (con ícono de lista)
- Diametro de la pecera (con ícono de flechas horizontales)
- Profundidad de la pecera (con ícono de flechas verticales)
- Descripción (campo de texto libre)

En la parte inferior del modal se encuentran dos botones: "Cerrar" y "Enviar".

Figura 60. Modal agregar pecera de interfaz peceras

Fuente: Autor del proyecto

Modal peceras



Este modal, titulado "Agregar pez a pecera", permite agregar peces a una pecera existente. Incluye los siguientes campos de entrada:

- Un menú desplegable para seleccionar el tipo de pez (actualmente muestra "Pez").
- Numero de peces en la pecera (con ícono de hashtag)
- Fecha de inicio (con ícono de calendario)

En la parte inferior del modal se encuentran dos botones: "Cerrar" y "Enviar".

Figura 61. Modal agregar pez a pecera de interfaz peceras

Fuente: Autor del proyecto

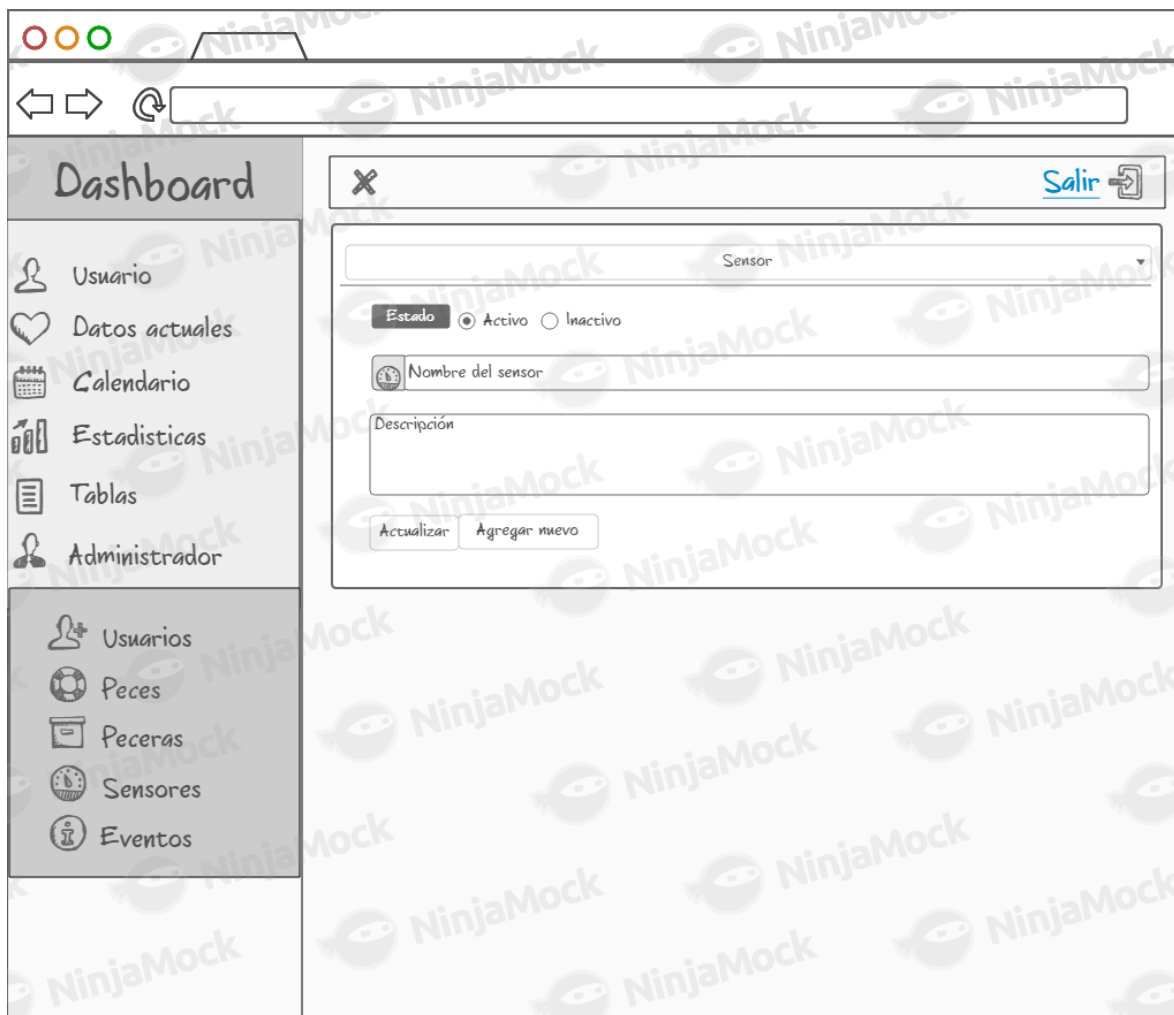
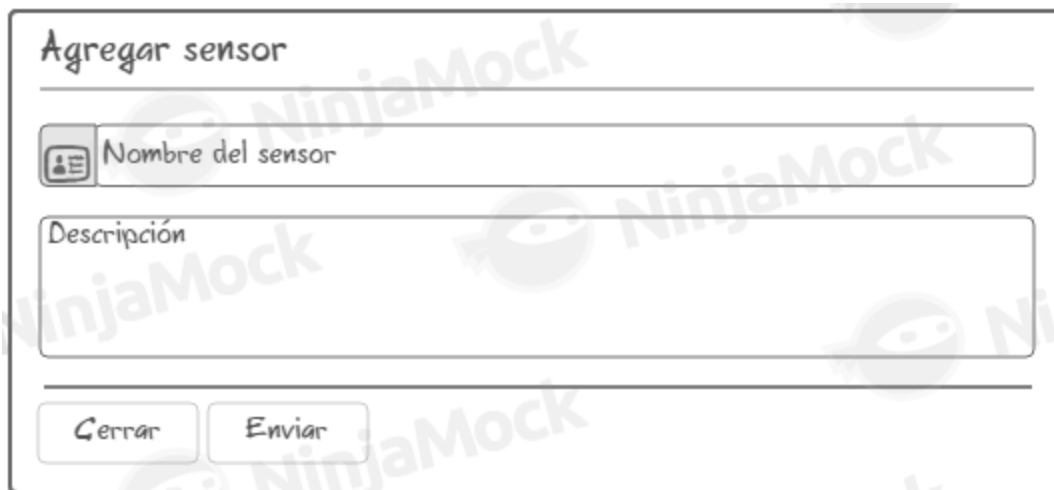
Interfaz sensores.

Figura 62. Interfaz sensores

Fuente: Autor del proyecto

Modal sensores



Agregar sensor

Nombre del sensor

Descripción

Cerrar Enviar

The image shows a modal window titled "Agregar sensor". It contains two input fields: "Nombre del sensor" with a list icon on the left, and "Descripción" with a larger text area. At the bottom, there are two buttons: "Cerrar" and "Enviar". The entire modal is enclosed in a rectangular border.

Figura 63. Modal agregar sensor de interfaz sensores

Fuente: Autor del proyecto


Interfaz eventos

The image shows a web application interface for event management. On the left is a sidebar titled "Dashboard" with the following menu items: Usuario, Datos actuales, Calendario, Estadísticas, Tablas, and Administrador. Below these are icons for Usuarios, Peces, Peceras, Sensores, and Eventos. The main content area contains a form for creating or editing an event. The form has a dropdown menu labeled "Evento", a radio button for "Estado" (Active/Inactive), a text input for "Nombre del Evento", a text area for "Descripción", and buttons for "Actualizar" and "Agregar nuevo". A "Salir" button is in the top right corner of the form area.

Figura 64. Interfaz donde se podrán crear y editar eventos, los cuales aparecerán en el calendario para ser agregado al respectivo día que se seleccione en el calendario

Fuente: Autor del proyecto

Modal interfaz eventos



The image shows a modal window titled "Agregar evento". It contains two input fields: "Nombre del evento" with a person icon on the left, and "Descripción" with a larger text area. At the bottom, there are two buttons: "Cerrar" and "Enviar". The entire modal is enclosed in a rectangular border.

Figura 65.Modal agregar evento de interfaz eventos

Fuente: Autor del proyecto

Apéndice D Puesta en marcha del socket

Para la puesta en marcha del socket se utilizó la carpeta /opt para almacenar los archivos en una carpeta servidor, tanto el jar como los script de iniciar y parar el socket, a los cuales tenemos que darles permisos de ejecución para poder inicializarlos, pero antes de ejecutar los scripts debemos habilitar el puerto establecido para las comunicaciones para que este pueda permitir la conexión de los clientes.

Servidor

```

Last login: Sat Aug 25 18:59:50 2018 from 190.90.21.2
root@localhost:~# cd /opt/
root@localhost:/opt# ls
apache-tomcat-8.5.32      jdk1.8.0_181          servidor
apache-tomcat-8.5.32.tar.gz  server-jre-8u181-linux-x64.tar.gz
root@localhost:/opt# cd servidor/
root@localhost:/opt/servidor# ls -l
total 1352
-rw-r--r-- 1 root root 1375463 Aug 12 10:28 Servidor.jar
-rwxr-xr-x 1 root root    38 Aug 12 10:16 Servidor-start.sh
-rwxr-xr-x 1 root root    74 Aug 12 09:28 Servidor-stop.sh
root@localhost:/opt/servidor# |

```

Figura 66. Archivos del socket en el servidor

Fuente: Autor del proyecto

Apéndice E Imágenes de la aplicación en ejecución

En esta sección de anexos se presentan imágenes de la aplicación en ejecución y se respectiva descripción, la cual cumple con las características presentes en el anexo 3 diseño de interfaces.

Interfaz de administración de información del usuario

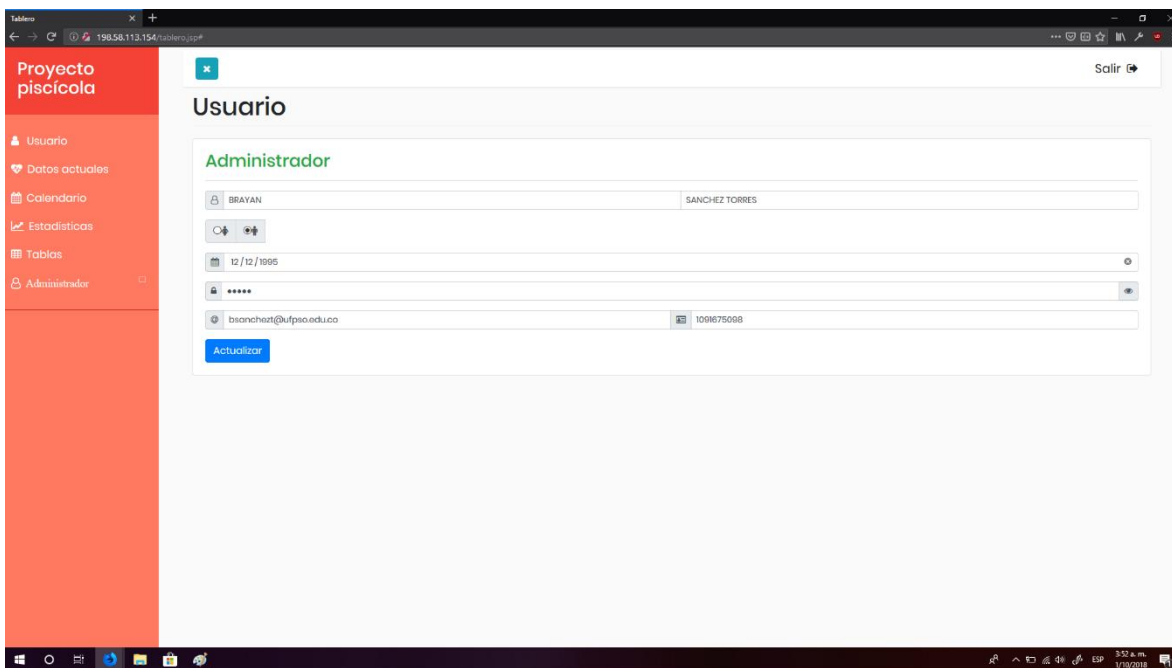


Figura 67. Interfaz administración de la información del usuario

Fuente: Autor del proyecto

Interfaz de calendario

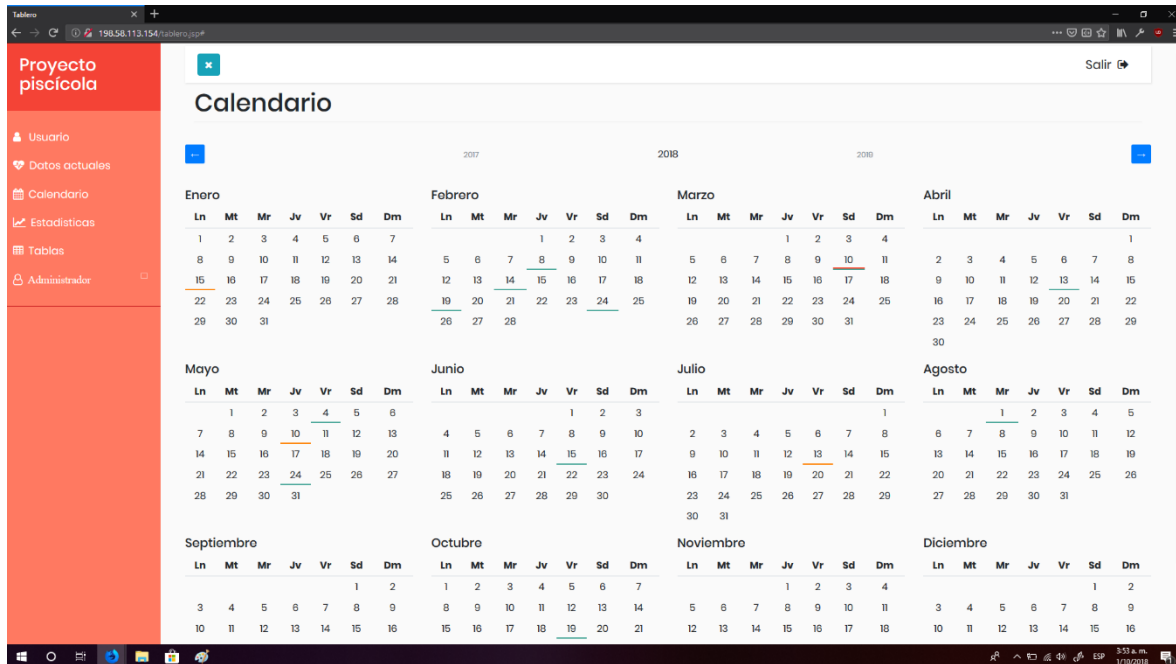


Figura 68. Interfaz de calendario

Fuente: Autor del proyecto

Modal interfaz calendario

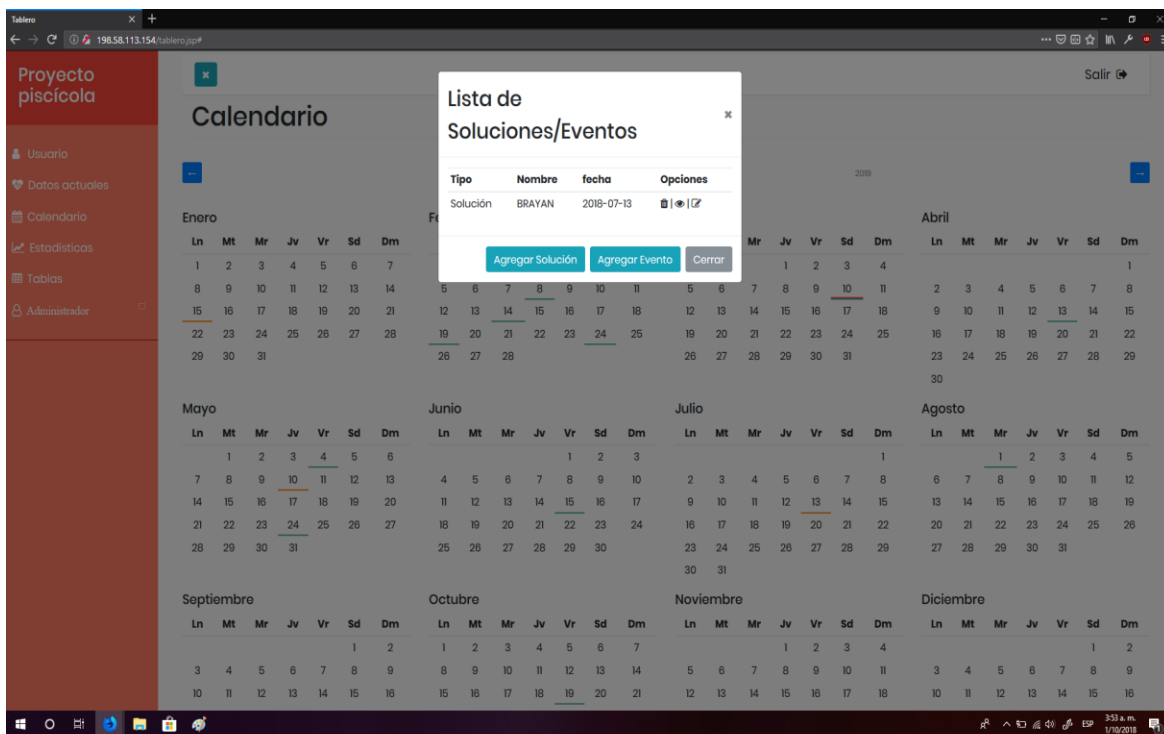


Figura 69. Interfaz de calendario lista de incidencias y eventos

Fuente: Autor del proyecto

Modal interfaz calendario

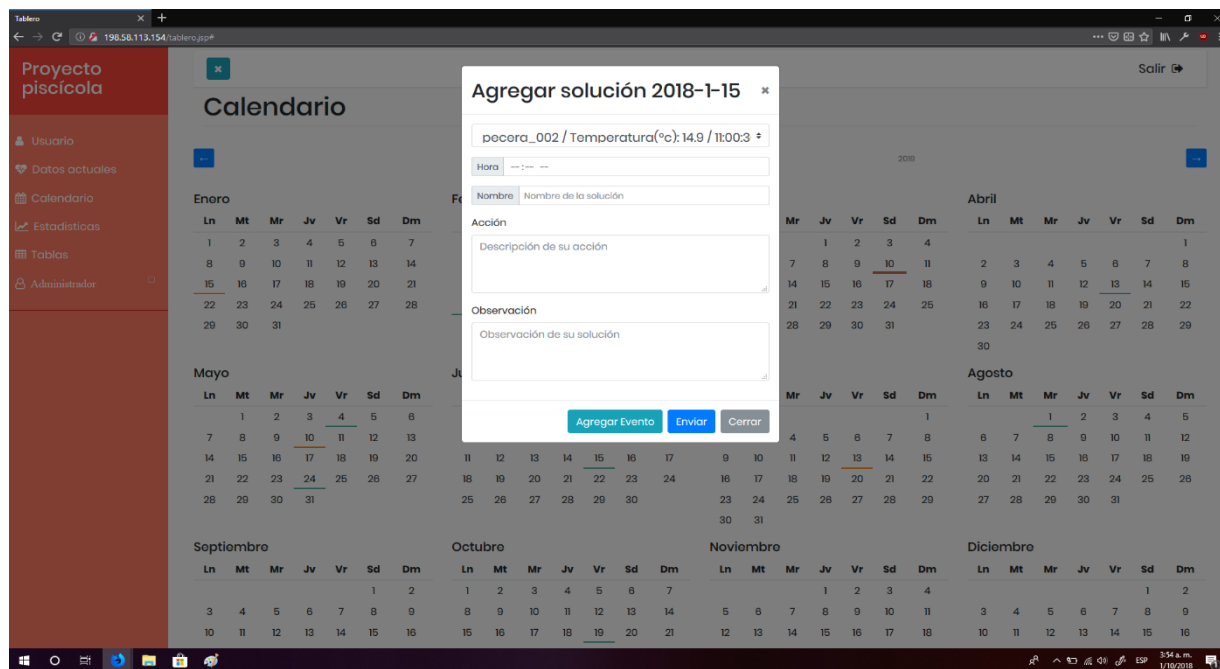


Figura 70. Interfaz Calendario agregar solución

Fuente: Autor del proyecto

Modal interfaz calendario

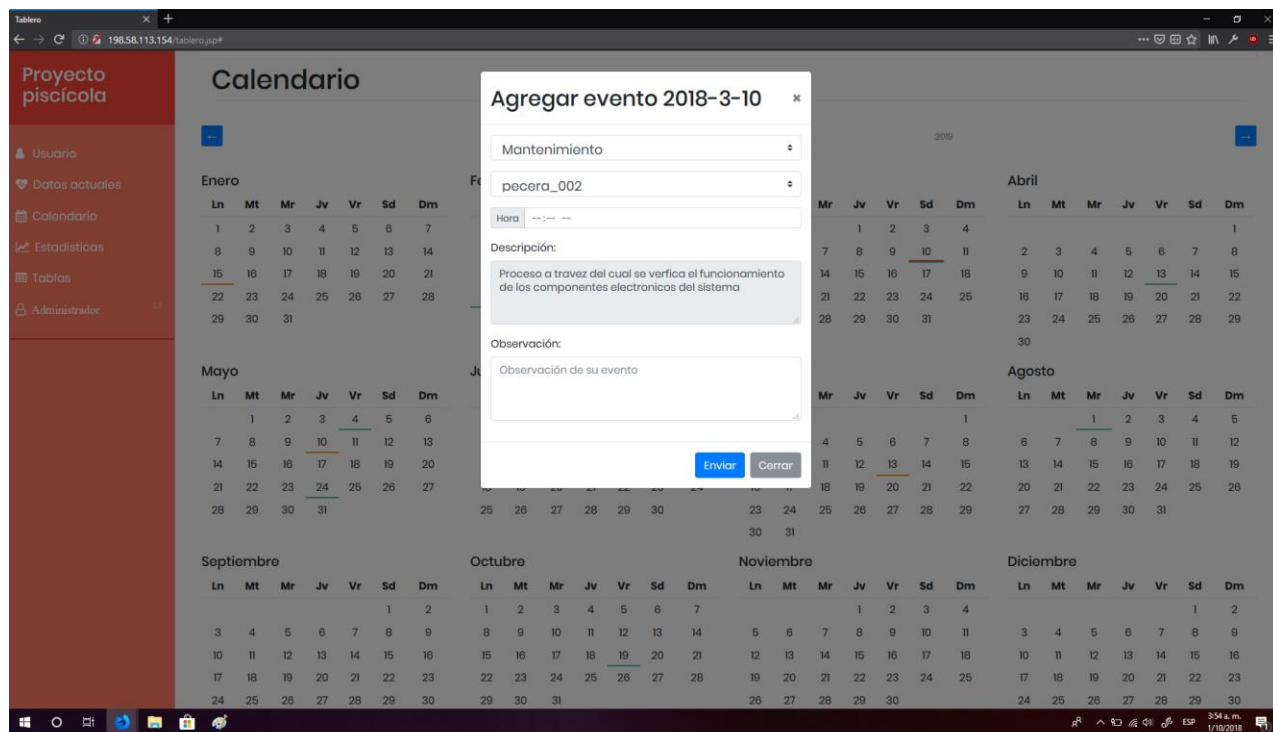


Figura 71. Interfaz de calendario agregar evento

Fuente: Autor del proyecto

Interfaz relacionada con la parte estadística del software.

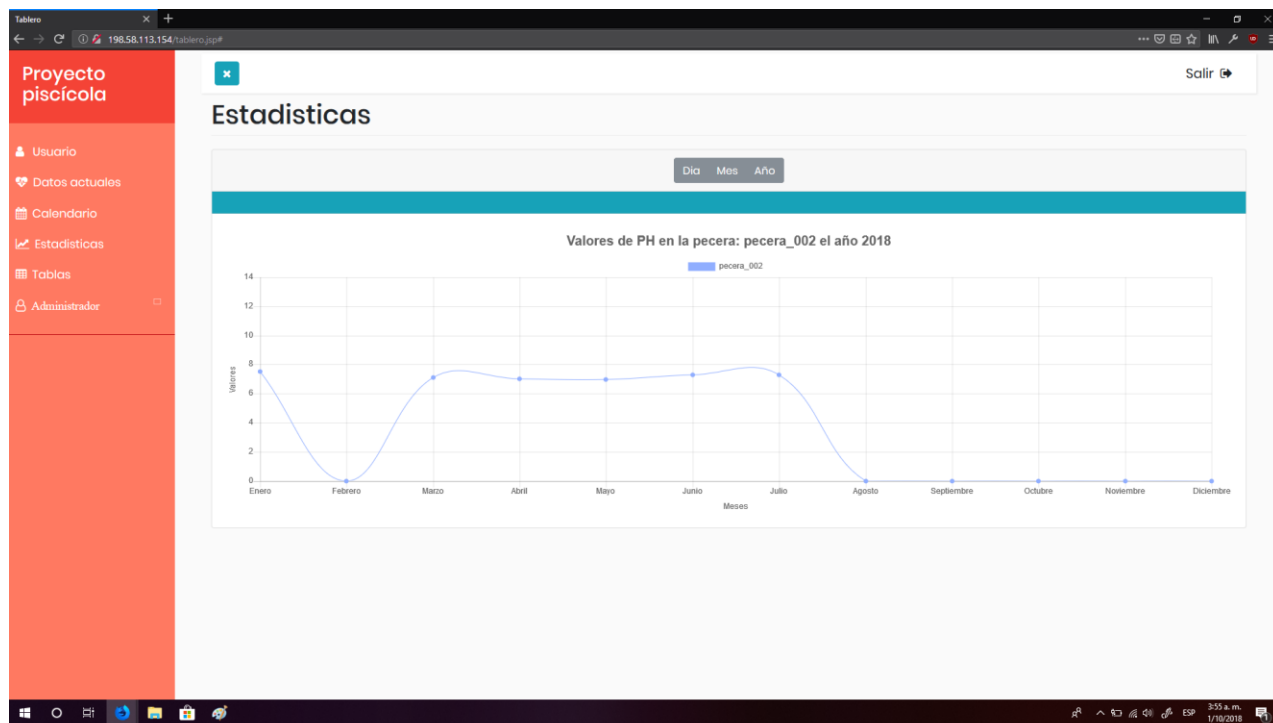


Figura 72. Interfaz de estadísticas

Fuente: Autor del proyecto

Interfaz relacionada con los eventos del sistema

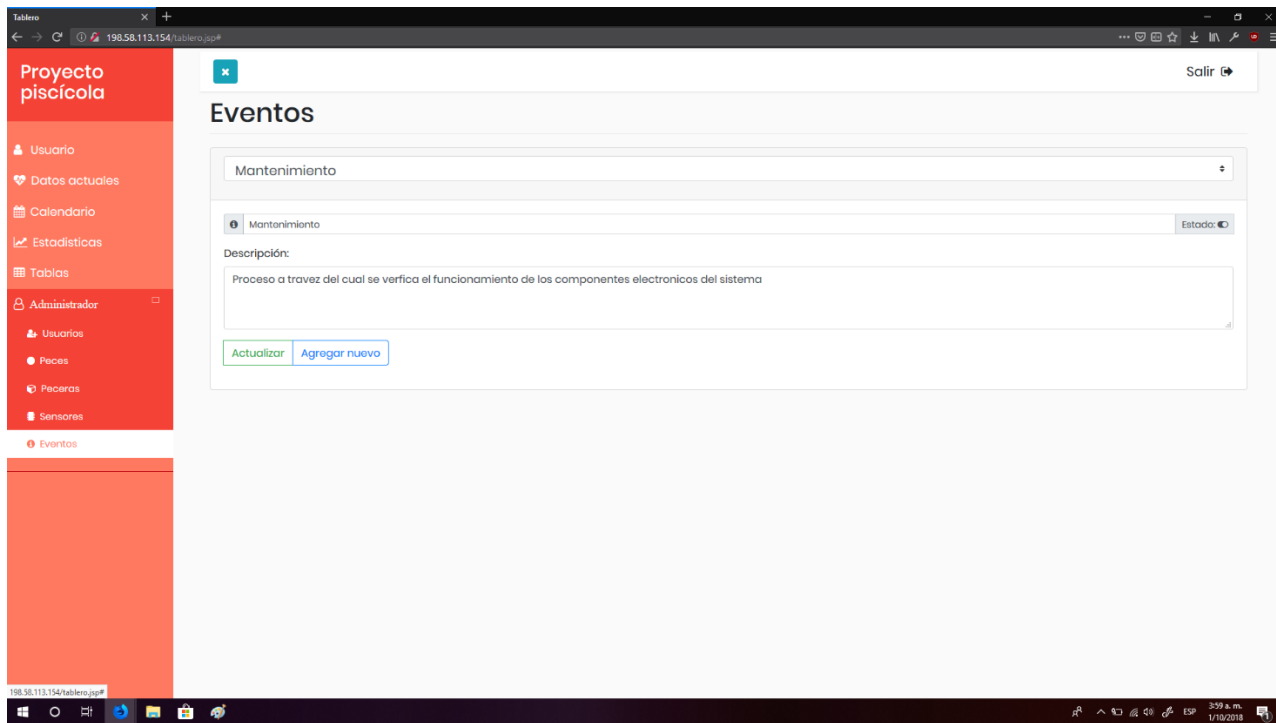


Figura 73. Interfaz de eventos

Fuente: Autor del proyecto

Interfaz relacionada a la gestión de operarios llevada a cabo por el administrador

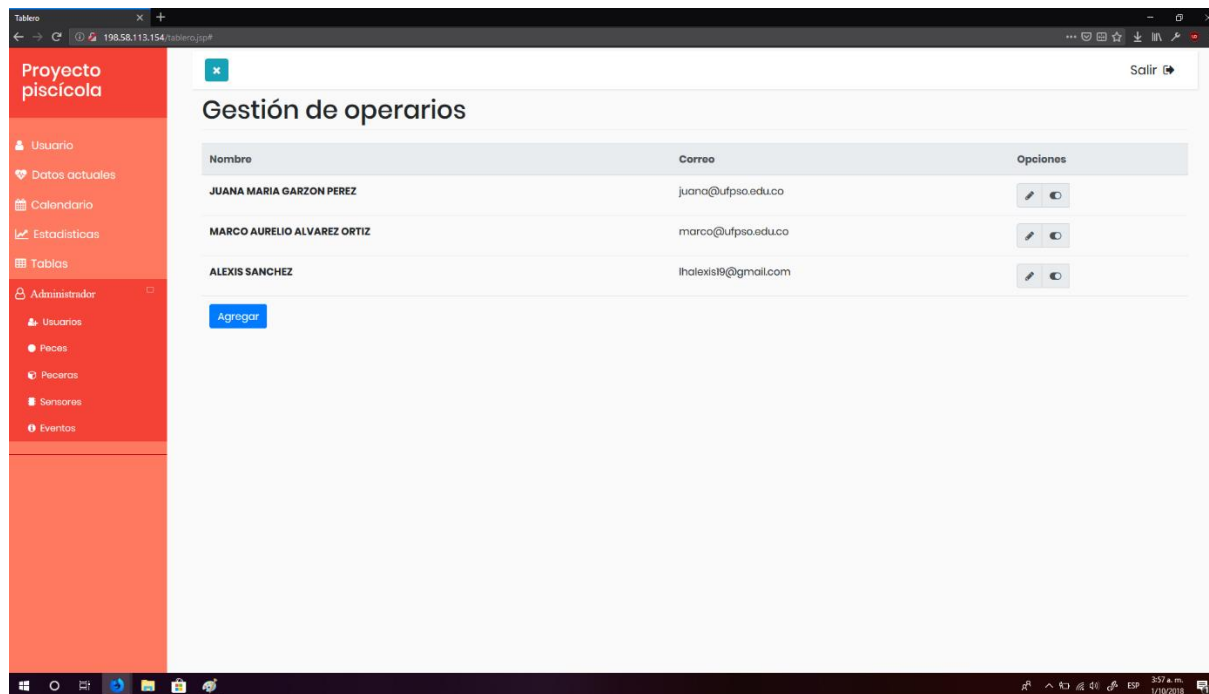


Figura 74. Interfaz gestión de operarios

Fuente: Autor del proyecto

Modal interfaz gestión de operarios

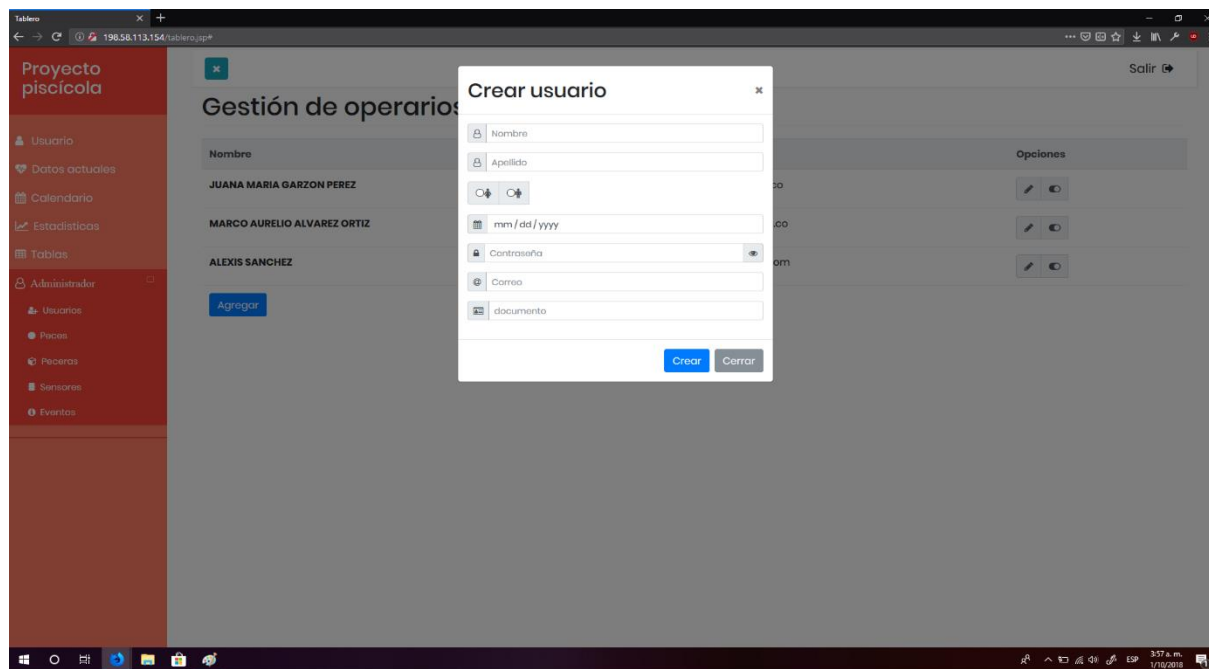


Figura 75. Interfaz crear usuario

Fuente: Autor del proyecto

Interfaz relacionada a la elaboración de informes del sistema.

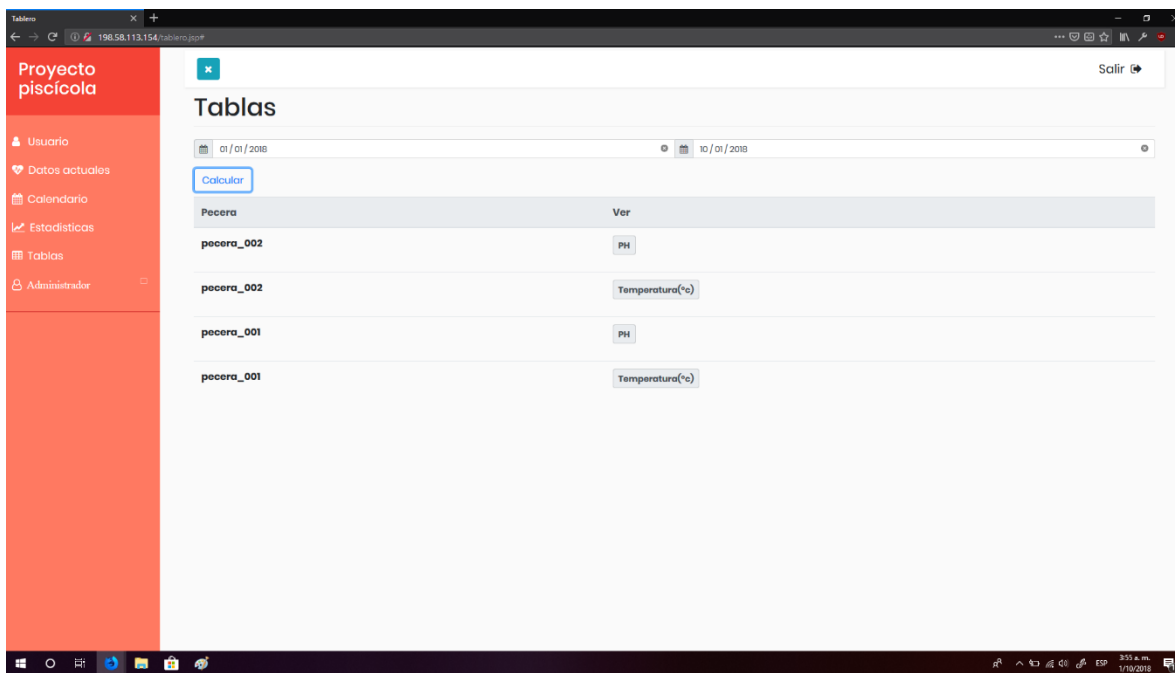
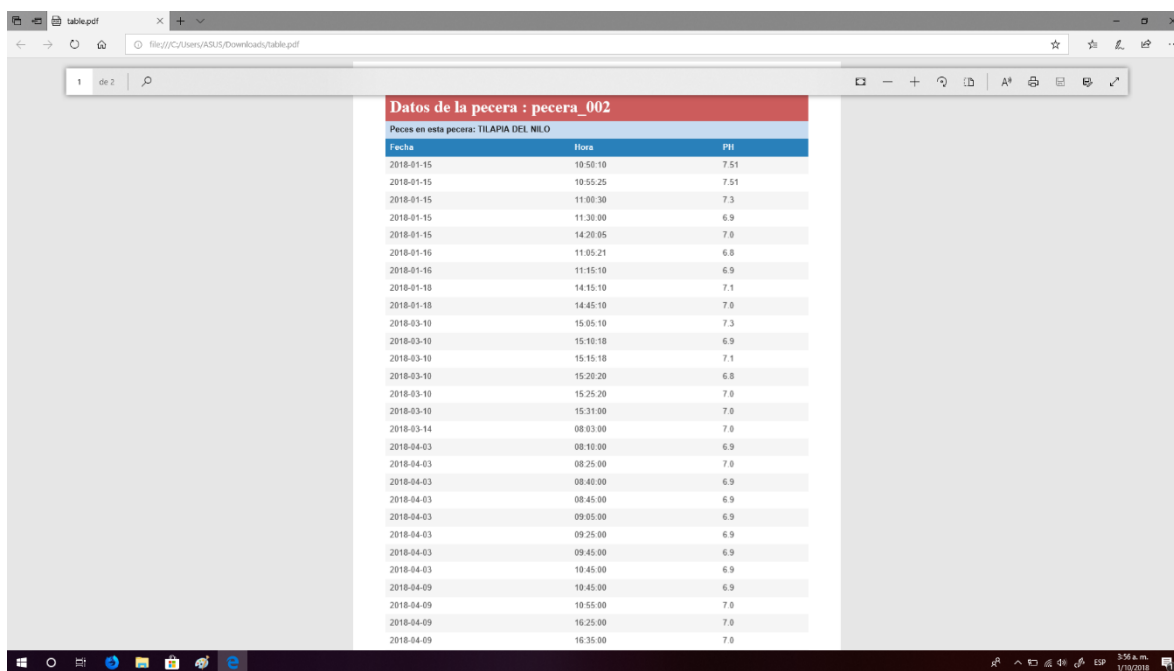


Figura 76. Interfaz de tablas de datos

Fuente: Autor del proyecto

Interfaz de datos



1 de 2

Datos de la pecera : pecera_002

Peces en esta pecera: TILAPIA DEL NILO

| Fecha | Hora | PH |
|------------|----------|------|
| 2018-01-15 | 10:50:10 | 7.51 |
| 2018-01-15 | 10:55:25 | 7.51 |
| 2018-01-15 | 11:00:30 | 7.3 |
| 2018-01-15 | 11:30:00 | 6.9 |
| 2018-01-15 | 14:20:05 | 7.0 |
| 2018-01-16 | 11:05:21 | 6.8 |
| 2018-01-16 | 11:15:10 | 6.9 |
| 2018-01-18 | 14:15:10 | 7.1 |
| 2018-01-18 | 14:45:10 | 7.0 |
| 2018-03-10 | 15:05:10 | 7.3 |
| 2018-03-10 | 15:10:18 | 6.9 |
| 2018-03-10 | 15:15:18 | 7.1 |
| 2018-03-10 | 15:20:20 | 6.8 |
| 2018-03-10 | 15:25:20 | 7.0 |
| 2018-03-10 | 15:31:00 | 7.0 |
| 2018-03-14 | 08:03:00 | 7.0 |
| 2018-04-03 | 08:10:00 | 6.9 |
| 2018-04-03 | 08:25:00 | 7.0 |
| 2018-04-03 | 08:40:00 | 6.9 |
| 2018-04-03 | 08:45:00 | 6.9 |
| 2018-04-03 | 09:05:00 | 6.9 |
| 2018-04-03 | 09:25:00 | 6.9 |
| 2018-04-03 | 09:45:00 | 6.9 |
| 2018-04-03 | 10:45:00 | 6.9 |
| 2018-04-09 | 10:45:00 | 6.9 |
| 2018-04-09 | 10:55:00 | 7.0 |
| 2018-04-09 | 16:25:00 | 7.0 |
| 2018-04-09 | 16:35:00 | 7.0 |

Figura 77. Informe de datos

Fuente: Autor del proyecto

Interfaz para la administración de los sensores del sistema

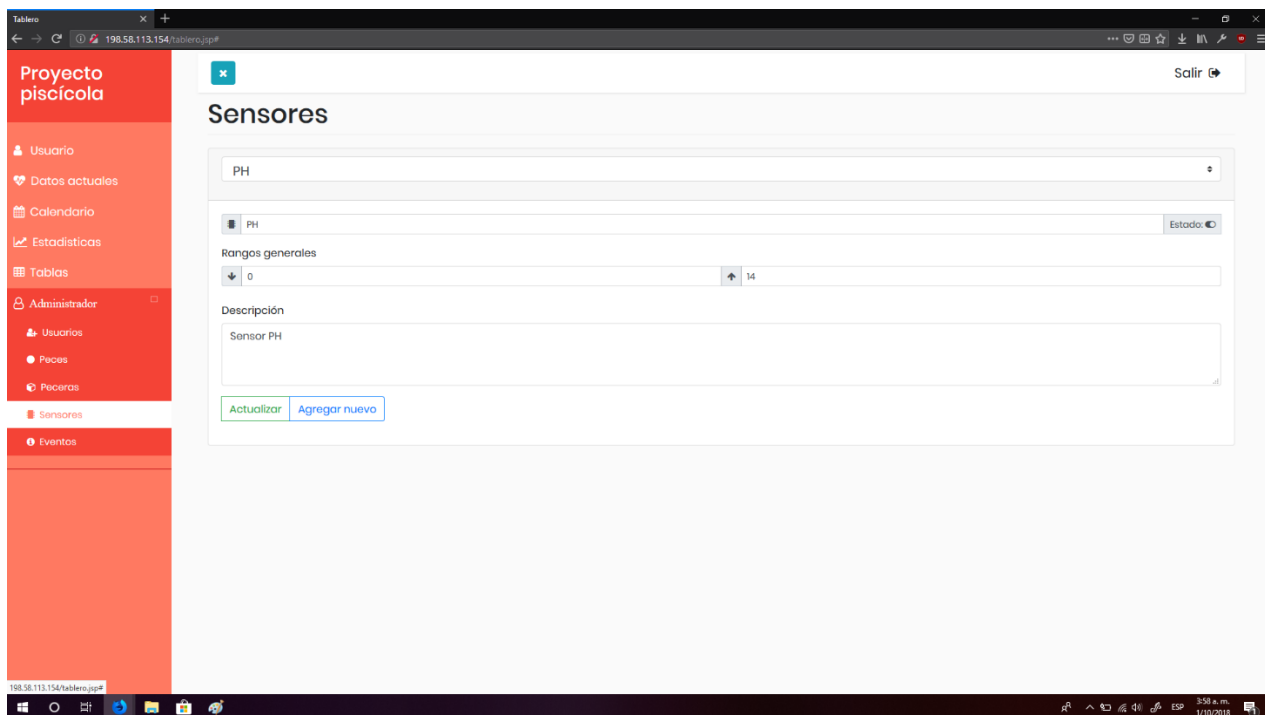


Figura 78. Interfaz de sensores

Fuente: Autor del proyecto

Interfaz para la administración de los peces que estén registrados en el sistema

Proyecto piscícola

Usuario

Datos actuales

Calendario

Estadísticas

Tablas

Administrador

Usuarios

Peces

Paceras

Sensores

Eventos

Salir

Peces

TILAPIA DEL NILO

TILAPIA DEL NILO OREOCHROMIS NILOTICUS

AFRICANO

OREOCHROMIS

Descripción

Aquí va una descripción del pez que se yo

Actualizar Agregar nuevo

PH

Rango estable

6 8

Actualizar

Rangos indicadores

| Min | Max | Opciones |
|-----|-----|----------|
| 9 | 12 | |

Figura 79. Interfaz de peces

Fuente: Autor del proyecto

Interfaz de la administración de las peceras existentes en el sistema

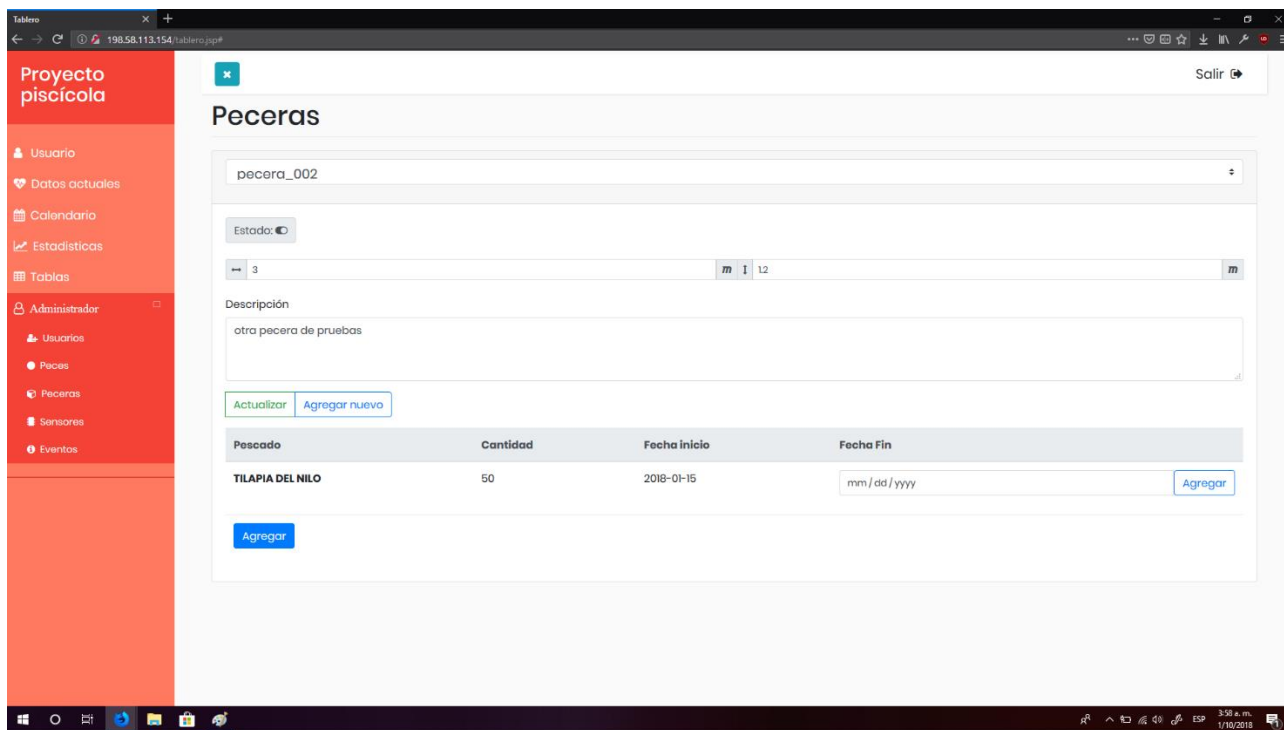


Figura 80. Interfaz de peceras

Fuente: Autor del proyecto