



**UNIVERSIDAD CÉSAR VALLEJO**

**FACULTAD DE INGENIERÍA Y ARQUITECTURA  
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS**

**Sistema inteligente para el Control de aforo en Institución de  
Salud Pública, Puno 2023**

**TESIS PARA OBTENER EL TÍTULO PROFESIONAL DE:**

Ingeniero de Sistemas

**AUTOR:**

Ruelas Paredes, Rusbel Oliver (orcid.org/0000-0001-8798-2032)

**ASESOR:**

Dr. Agreda Gamboa, Everson David (orcid.org/0000-0003-1252-9692)

**LÍNEA DE INVESTIGACIÓN:**

Sistemas de Información y Comunicaciones

**LÍNEA DE RESPONSABILIDAD SOCIAL UNIVERSITARIA:**

Desarrollo Económico, Empleo y Emprendimiento

LIMA - PERÚ

2023

## **Dedicatoria**

*Le dedico este trabajo de investigación a mi padre Rolando a mis hermanas Lorenia, Faviola y en memoria de mi madre María Ivia, cuyo apoyo constante es mi mayor motivación e inspiración para seguir adelante.*

*Rusbel*

## Agradecimiento

*Mi gratitud a la facultad de ingeniería de sistemas y a sus docentes quienes con su labor me brindaron la dirección del camino a seguir en cuanto a la ingeniería, así como a la universidad por ofrecer la oportunidad de reunir los recursos y materiales necesarios para realizar el presente trabajo.*

*El autor*

## Índice de contenidos

	Pág.
Carátula .....	i
Dedicatoria .....	ii
Agradecimiento .....	iii
Índice de contenidos .....	iv
Índice de tablas .....	v
Índice de figuras .....	vi
Resumen .....	viii
Abstract .....	ix
I. INTRODUCCIÓN .....	1
II. MARCO TEÓRICO .....	4
III. METODOLOGÍA.....	14
3.1 Tipo y diseño de investigación.....	14
3.2 Variables y operacionalización .....	14
3.3 Población, muestra y muestreo .....	16
3.4 Técnicas e instrumentos de recolección de datos .....	17
3.5 Procedimientos.....	18
3.6 Método de análisis de datos.....	18
3.7 Aspectos éticos .....	19
IV. RESULTADOS .....	20
V. DISCUSIÓN.....	29
VI. CONCLUSIONES.....	31
VII. RECOMENDACIONES .....	32
REFERENCIAS.....	33
ANEXOS .....	34

## Índice de tablas

	Pág.
<i>Tabla 1.</i> Porcentaje de atención.....	17
<i>Tabla 2.</i> Prueba de normalidad indicador 1.....	23
<i>Tabla 3.</i> Prueba de normalidad indicador 2.....	23
<i>Tabla 4.</i> Prueba de normalidad indicador 3.....	24
<i>Tabla 5.</i> Prueba de hipótesis indicador 1. ....	26
<i>Tabla 6.</i> Prueba de hipótesis del indicador 2 .....	27
<i>Tabla 7.</i> Prueba de hipótesis del indicador 3 .....	28

## Índice de figuras

	Pág.
Figura 1. Resultado de tiempo promedio de espera.....	20
Figura 2. Resultado infracciones a las políticas de aforo .....	21
Figura 3. Resultado costos .....	22
Figura 4. Desarrollo del prototipo. ....	45
Figura 5.maqueta del concepto .....	46
Figura 6.Captura de pantalla de IntelliJ-IDEA .....	48
Figura 7.Diagrama de componentes del sistema. ....	49
Figura 8.Diagrama de conexión de los sensores. ....	50
Figura 9.Flujograma de comprobación de sensores, .....	51
Figura 10.flujo de programa al leer los sensores.....	52
Figura 11. Rutina de lectura de sensores .....	54
Figura 12. operación del mecanismo de barrera ultrasónica .....	55
Figura 13. implementación de barrera ultrasónica.....	56
Figura 14. Diagrama de clases despliegue en Android .....	59
Figura 15. Diagrama de clases para la API parte 1 .....	60
Figura 16. Diagrama de clases para la API parte 2 .....	61
Figura 17. Mockup Pantalla principal.....	62
Figura 18. Configuraciones básicas de preferencias. ....	63
Figura 19. Mockup selección de aforo máximo .....	63
Figura 20. Mockup credenciales y rutas a los servicios.....	64
Figura 21. Mockup Confirmación de cierre de app .....	64
Figura 22. Arquitectura de la API REST de servicios .....	65
Figura 23. End-points HTTP del api de cuentas. ....	66
Figura 24. End-points del API de clientes.....	66

Figura 25. End-points para los datos de los dispositivos.....	67
Figura 26. Vista Web .....	67
Figura 27. Vista Web Login .....	68
Figura 28. Vista Web Landing Page .....	68

## Resumen

A través de este trabajo de investigación se explora el diseño y desarrollo de un dispositivo contador de personas creado para mejorar de manera económica y simple el control del espacio público en una institución de salud en Puno en el año 2022 haciendo uso de tecnologías de la nube para la transferencia y sincronización de los datos a lo largo de todos los dispositivos distribuidos en un espacio geográfico o múltiples espacios. Para evaluar el impacto se ha abordado tres dimensiones y se evidencia que el tiempo promedio de espera de las colas se reducen al igual que la tasa de infracciones y además se reduce la cantidad de personas que requiere la tarea de mantener el respeto a las normas de aforo de la institución de salud. Para el diseño y desarrollo del prototipo se hace uso de microcontroladores y sensores de amplio uso en la industria tecnológica y electrónica como el microcontrolador atmega328p utilizado en la plataforma Arduino y su comunicación a la red por medio de una aplicación móvil diseñada para Android y utilizando http y web sockets para la sincronización de los datos en la nube.

*Palabras clave: Sistema inteligente, Proceso de control, Aforo, Institución de salud pública.*



## **Abstract**

Through this research work we explore the design and development of a people counter device created to improve in an economical and simple way the control of public space in a health institution in Puno in the year 2022 making use of cloud technologies for the transfer and synchronization of data across all devices distributed in a geographical space or multiple spaces. To evaluate the impact, three dimensions have been addressed and it is evident that the average waiting time of queues were reduced as well as the violation rate and also the number of people required to maintain the respect of the capacity rules of the health institution is reduced. For the design and development of the prototype is used microcontrollers and sensors widely used in the technology and electronics industry as the atmega328p microcontroller used in the Arduino platform and its communication to the network through a mobile application designed for Android and using http and web sockets for data synchronization in the cloud.

*Keywords: Intelligent system, Control process, Capacity, Public health institution.*

## I. INTRODUCCIÓN

A raíz del contexto de una pandemia ocasionada por la COVID-19 que amenazó una gran parte de la humanidad, han salido a la luz nuevas necesidades anteriormente no contempladas una de estas necesidades es el control estricto de espacios públicos para poder determinar la cantidad exacta de personas ocupando un determinado espacio público. Este control necesario contribuye a mitigar la propagación del virus al poder evitar la sobre concentración de personas en un mismo local. Producto de esta necesidad se propone un sistema de control para poder contabilizar el ingreso o salida de una persona a un local o espacio a través de múltiples dispositivos colocados en las entradas o salidas, el sistema propuesto utiliza tecnologías de hardware abierto como el microcontrolador atmega328p utilizado en las tarjetas de desarrollo de Arduino y los sensores ultrasónicos HC-SR04 que en conjunto procesan y envían una señal de activación ya sea de entrada o salida al dispositivo móvil el cual por medio de su conexión al dispositivo a través comunicación serial procesa y sincroniza los datos provenientes del sensor y los actualiza con los demás dispositivos en tiempo real haciendo uso de web-sockets y http para la comunicación con un servidor backend desplegado en la nube para poder comunicarse con cada dispositivo del sistema y de esta manera mantener actualizada esta data de forma transparente y brindar información real sobre el aforo actual de un determinado lugar además de concentrar los datos de múltiples lugares y mostrarlos en tiempo real para obtener mediciones acerca de las horas de mayor concurrencia de personas, todo esto con el objetivo de garantizar un numero controlado de personas en un mismo espacio público y ayudar a las organizaciones a poder demostrar el respeto a la políticas de aforo brindadas por el gobierno en salvaguardia de la salud además de dar la oportunidad operativa para las actividades que habían sido anteriormente prohibidas al no poder demostrar fehacientemente el control de su concurrencia y produciendo un grave daño económico a ciertos sectores que dependen de dichas actividades económicas.

entre muchos factores que han servido para mitigar los contagios a nivel internacional el control de aforo en espacios concurridos es uno de ellos, dada la naturaleza infecciosa del virus a nivel aéreo y principalmente por spray, la concurrencia excesiva de personas en espacios cerrados además puede producir un incremento en la carga viral lo cual lleva a un desarrollo más grave de la enfermedad, sin embargo en el Perú dado el estado actual de desarrollo tecnológico el estado solo se ha limitado a recomendar un control de aforo en los lugares más concurridos, pero sin proponer ninguna manera de ejercer este control en los espacios públicos compartidos llevando estos lineamientos a las interminables colas en las afueras de los establecimientos y el entorpecimiento de las actividades comerciales de estos agentes económicos en el mejor de los casos o el cierre de instituciones como los colegios como medida más extrema, es en esta brecha en la que este trabajo propone el desarrollo de un sistema de control de aforo o concurrencia que integre el procesamiento de una señal de entrada o salida de un establecimiento haciendo uso de un par de sensores en un lugar de acceso, salida o ambas, la integración y sincronización de señales utilizando un servidor que centralice y sirva estas señales además de brindar esta información de manera pública para que las personas que consulten al sistema puedan saber el estado actual de afluencia en cada lugar controlado.

Por lo expuesto en líneas anteriores se desprende el hecho de que las instituciones estatales y en específico las instituciones de salud publicas requieren herramientas que les permitan controlar el aforo para mejorar el proceso de atención brindado.

Por lo tanto, se puede formular el problema general de la presente investigación con la siguiente pregunta: ¿En qué medida el desarrollo de un sistema para la gestión de los espacios públicos concurridos utilizando el microcontrolador ATmega328p y el sensor HC SR04 en comunicación con Android como sistema operativo ayudará a mejorar el control de concurrencia de personas en la institución de salud pública en Puno en el año 2022?

Se justifica el presente trabajo de investigación ante la necesidad de contar con algún mecanismo que permita controlar la concurrencia de personas en espacios públicos para servir como herramienta de mitigación ante la transmisión aérea del virus producto de un abarrotamiento de personas sin control en espacios concentrados, así como la mejora en el proceso de atención pública de instituciones o lugares con demasiada concentración de personas. Por otro lado, el uso de tecnologías de uso abierto como lo son Arduino y Android ayuda a reducir los costos de implementación de un sistema de esta categoría.

En esta investigación se pretende alcanzar el siguiente objetivo general: Mejorar el control de concurrencia de personas en institución pública mediante el desarrollo de un sistema para el control de aforo en espacios públicos concurridos (contador de personas) utilizando el microcontrolador atmega328p y sensor HC-SR04 en comunicación directa con dispositivo Android y sincronización de datos en la nube. Y como objetivos específicos se pretende: Mejorar el cumplimiento de las políticas de control de aforo, mejorar la respuesta en tiempo real de los ambientes controlados, evaluar el costo beneficio de la implementación del sistema; en instituciones de salud pública cuyo proceso de atención se encuentre afectado por las restricciones del estado de emergencia.

En consecuencia, a lo anterior presenta la siguiente hipótesis: El desarrollo de un sistema para el control de concurrencia de personas utilizando el microcontrolador ATmega328p y sensor HC-SR04 en comunicación con dispositivo Android y sincronización de datos en la nube, tendrá un impacto positivo sobre la gestión de espacios públicos en institución de salud pública 2022.

## II. MARCO TEÓRICO

En su investigación (Ballester Mainar 2021) propone un sistema de control de aforo basado en tags de señales basadas en radio frecuencia la cuales son enviadas mediante tecnología bluetooth a un servidor el cual utiliza tecnologías lorawan para la centralización de estos datos generados por los tags además hace uso de un algoritmo de posicionamiento espacial para calcular la posición exacta de cada dispositivo tag con un margen de error de 2.5 mt, Ballester concluye que es posible desarrollar e implementar un sistema de control de aforo basado en Tags RF y propone 2 pruebas para la demostración de 2 etapas del sistema desarrollado, la primera prueba consiste en analizar la data proveniente de los tags para comprobar la fidelidad de los datos obtenidos con respecto a la realidad, prueba que supera exitosamente, la segunda etapa de prueba la realiza a nivel de servidor y centralización de datos basados en los canales usados por las tecnologías de radiofrecuencia prueba que también supera con éxito, en sus recomendaciones propone utilizar un circuito impreso para poder reducir el tamaño de la antena la cual hace que el prototipo tenga un tamaño considerable.

En el trabajo de fin de grado propuesto por (Gallego Ferrer 2016) desarrolla un sistema de control de aforo de personas basado en brazaletes de radiofrecuencia pasiva los cuales son escaneados al ingresar o salir de un establecimiento, permitiendo de esta manera contar con un aforo controlado de asistentes a eventos masivos con el fin de poder evitar la sobreventa de entradas a ciertos espectáculos, el sistema propuesto consta de 4 componentes, por un lado la pulsera con un tag de radiofrecuencia pasivo , un lector de tags , una app para monitorear los datos y un servidor backend el cual centraliza los datos y los sirve para que la app pueda mostrarlos.

(Calvo Rebollo 2017) para su trabajo de fin de grado propone un sistema de control de acceso haciendo uso de múltiples tecnologías como son el NFC de los teléfonos inteligentes, una aplicación a modo de servidor en python, una app desarrollada para Android las cuales interactúan para poder controlar el acceso o los permisos a cualquier instalación, en su trabajo indica los

beneficios de este sistema frente a los existentes en el mercado, son lo de no necesitar ningún tipo de infraestructura previa y su adopción se facilita por el uso de sistema android y las tarjetas de NFC pasivas las cuales tienen un costo relativamente menor, el proyecto concluye con un plan de pruebas ejecutado exitosamente y la recomendación del uso de las tecnologías NFC por su adaptabilidad frente a diferentes tipos de aplicaciones.

(Sánchez et al. 2017) realizan un proyecto en el cual desarrollan un dispositivo electrónico basado en una red de sensores ultrasónicos que se comunican con un servidor central para controlar la ocupación de lotes de parqueo para de esta manera controlar el aforo vehicular de una playa de estacionamiento, estos datos son transmitidos mediante señales de radio a un servidor y estos datos son retribuidos a una aplicación diseñada para teléfonos móviles que funcionen bajo el sistema operativo Android, la cual sirve los datos para informar a los usuarios del sistema de la capacidad de aforo actual de automóviles en el estacionamiento.

Con respecto a las **bases teóricas**, se tuvo:

**Arduino**, definido en su propio sitio web como una plataforma para la electrónica de código abierto, es una empresa la cual opera en base a modelos de tarjetas electrónicas de entrenamiento y una plataforma de IDE para el desarrollo de software exclusivo para estos dispositivos conocidos como Tarjetas Arduino estas tarjetas son comercializadas por la empresa sin embargo los planos y diagramas electrónicos al ser de código abierto están disponibles para ser descargados y producidos, los tres componentes principales de su plataforma son el Hardware o la placa, la plataforma y el IDE.

**Tarjetas Arduino**, cuenta con varios modelos de tarjetas algunos de estos son: Arduino UNO, El modelo se diferencia de los demás por el tipo de puerto USB tipo B además tiene la característica de poseer un socket que permite remover el microcontrolador. Arduino Nano, lleva el microcontrolador ATmega328p de tipo SMD y el puerto USB micro -b es característico en este modelo el tamaño de la placa y por llevar un microcontrolador con

encapsulado soldado a la placa. Arduino Leonardo, el cual tiene más memoria y un microcontrolador ATmega32u.

**Plataforma arduino IDE**, Arduino IDE en la versión 1.8.15 a la fecha de este trabajo, Es un programa multiplataforma que permite escribir código para las tarjetas, compilarlas y subir los archivos generados al microcontrolador, este utiliza un lenguaje de programación variante de c++ llamado sketch además cuenta con una amplia gama de librerías listas para ser descargadas a los proyectos.

**Bootloader**, (Purdum 2015) define el bootloader en su libro como “Una pequeña pieza de código que permite al microcontrolador comunicarse con el mundo exterior”, (J. M. Hughes 2016) trata el bootloader como una zona de memoria reservada cargada con un firmware en una dirección específica, esta dirección específica en memoria es donde el micro controlador busca instrucciones por primera vez desde el arranque, debe ser configurada y tratada de forma especial, por lo tanto esta zona no es sobrescrita durante la carga de otros programas al dispositivo; además se mantiene su información durante los ciclos de interrupción de trabajo, reservar este espacio de memoria en un microcontrolador típicamente involucra el uso de un dispositivo especial que sea capaz de comunicarse con el microcontrolador a través de ICSP(In Circuit Serial Programming), también es necesario que los bit fuse o bits de configuración del dispositivo hayan sido correctamente configurados para que el micro controlador obtenga información sobre la organización de la memoria y el inicio de ejecución de las instrucciones que no son parte del bootloader. Esta disposición de bootloader permite a los dispositivos de avr como el ATmega328 cargar partes del código de firmware a la memoria no volátil y transferir datos mediante comunicación serial y es el corazón mismo de la plataforma Arduino.

**Organización de la memoria en Arduino**, se tocará a profundidad la arquitectura del microcontrolador paginas adelante, sin embargo, en este punto es preciso referir a la forma en que se organiza la memoria en Arduino, ya que de esta organización depende la codificación de los programas de

manera efectiva, en la ilustración 6 se muestra un esquema general de la organización básica de la memoria en el microcontrolador y como es administrada en la plataforma de Arduino. (J. M. Hughes 2016) detalla que Arduino utiliza todos los tipos de memoria disponibles en el chip de avr ATmega328 siendo las mismas Flash, SRAM(Static random access) y EEPROM, siendo la memoria flash la utilizada para guardar el código de ejecución de los programas cargados al Arduino, mientras que la SRAM contiene datos transitorios como pueden ser las variables del programa y el stack y finalmente la memoria EEPROM que puede guardar datos incluso interrumpiendo la energía y puede ser cargada externamente al igual que la memoria flash.

**Plataforma Android**, (Manuel Báez, Álvaro Borrego et al. 2016) Definen Android como una plataforma de software, así como un sistema operativo basado en el kernel de Linux, propuesto en sus inicios como un sistema operativo solo para teléfonos móviles en la actualidad se encuentra en una amplia gama de dispositivos inteligentes como tablets notebooks entre otros dispositivos como relojes Pc y automóviles. La arquitectura actual propuesta por Android en el proyecto Android open source nos muestra los siguientes componentes:

**Application framework**, el marco de trabajo de aplicación es la parte designada del sistema para que los desarrolladores puedan trabajar con el software y el hardware del dispositivo por medio de las API's disponibles.

**Binder IPC**, es un mecanismo de enlazamiento interproceso que permite al marco de trabajo de la aplicación comunicarse fuera de los límites del proceso de la aplicación y le permite al marco de trabajo de aplicación acceder a los servicios del sistema Android e interactuar con los mismos.

**System Services**, los servicios del sistema son componentes modulares enfocados a una tarea como el administrador de ventanas o el administrador de notificaciones. Estos servicios son expuestos al marco de trabajo de la aplicación por medio de API's y permiten acceder a las llamadas expuestas por el hardware.



**Hardware abstraction layer (HAL)**, la capa de abstracción de hardware permite al sistema operativo de Android ser agnóstico sobre la implementación de los controladores a bajo nivel.

**Linux Kernel**, todo el sistema operativo está basado en una parte del kernel de Linux la cual es responsable de los principales procesos y tareas del dispositivo Android es además en una buena parte una compilación de muchos drivers los cuales deben ser cargados o desarrollados sin alterar las capas de abstracción superiores.

**HTTP**, del inglés *Hyper text transfer protocol* o protocolo de transferencia de hiper texto en (Mozilla [sin fecha]) se hace un certero resumen sobre http definido como un protocolo que permite intercambiar recursos entre un cliente y un servidor, estos recursos pueden ser documentos html pero sin limitarse a estos, ya que por su sencillez y extensibilidad el protocolo http ha sido mejorado ampliando sus capacidades de transferencia pero sin perder la esencia de ser un protocolo de intercambio que no guarda estados entre sus consultas; sin embargo se pueden guardar estos datos entre sesiones a través de cookies. Para este protocolo los mensajes enviados desde el cliente se llaman peticiones y los mensajes que devuelve el servidor se llaman respuestas.

**Android app**, Android tiene un ecosistema de desarrollo de software para su plataforma por lo tanto el termino Android app se ha acuñado para denotar el software diseñado para ejecutarse en tal plataforma móvil, comúnmente este software es desarrollado haciendo uso del lenguaje java Android y el relativamente reciente kotlin, los programas son almacenados haciendo uso de un compresor de archivos y suelen tener la extensión APK.

**Comunicación serial**, (Russell 2010) explica que la comunicación serial a diferencia de la comunicación paralela hace uso de un cable para el envío de datos además de hacer uso de un protocolo personalizado para el intercambio de información entre el dispositivo central conocido como Maestro (Master) el cual dirige las comunicaciones y envía señales a los dispositivos periféricos para iniciar una comunicación con alguno de ellos conocidos como Esclavos

(Slave), El dispositivo maestro dirige la comunicación haciendo uso de un pulso de reloj (CLK) para la sincronización y una línea dedicada conocida como chip select (CS), cualquier dispositivo esclavo conectado al bus de comunicación esperará una señal de activación del maestro para en seguida responder a la comunicación,

**Sistemas embebidos**, (Heath 2003) en su libro diseño de sistemas embebidos hace una diferenciación para reconocer fácilmente que es un sistema embebido comparándolo con lo que no es, se compara un sistema embebido con un computador personal haciendo notar la relevancia de que cambiar la funcionalidad del PC es tan simple como cambiar el software que se está ejecutando, a diferencia de un sistema embebido el cual se diseña y produce para ejecutar una tarea específica y esta no puede ser cambiada por el usuario con facilidad, entonces un sistema embebido es un microprocesador con hardware y software diseñados específicamente para realizar una función o un rango de funciones sin permitir al usuario final del sistema alterar esta funcionalidad.

**Firmware**, (Tan et al. 2017) hace una revisión de los orígenes y surgimiento del término firmware como una necesidad a la confusión generada en 1967 por el código residente en ROM (Read Only Memory) el cual no estaba pensado para ser modificado ni directamente accedido por los usuarios de los dispositivos, por lo tanto se acuña el término firmware y este ha venido cambiando su significado a través del tiempo tanto que incluso hoy en día es debatible la definición; sin embargo la acepción actual más aceptada define firmware como un programa compilado generado para ejecutarse en un sistema embebido o dispositivo cerrado que se encarga de una secuencia de arranque y gestión de los dispositivos periféricos en dispositivos micro controlados.

**Hardware**, el hardware como tal no tiene una traducción al idioma español; sin embargo, se comprende su significado como la parte física de un sistema informático todo aquello que se puede tocar, aunque esta definición puede ser más certera si se refiere hardware como el conjunto de componentes

electromecánicos y circuitos electrónicos que hacen posible el funcionamiento del sistema informático en su conjunto.

**Driver**, entendido en el idioma español como controlador de dispositivo, es comúnmente definido como un programa informático con la capacidad de interactuar a través de una interfaz con un dispositivo periférico desde el sistema operativo. Para (Kadav y Swift 2012) un driver es un componente de software el cual provee una interfaz entre el software del sistema operativo y el hardware permitiendo una abstracción del mismo además de administrar y configurar el dispositivo. Para el kernel de Linux los controladores se apoyan en tres interfaces:

- Interfaz entre el controlador y el kernel.
- Interfaz entre el controlador y el dispositivo.
- Interfaz entre el controlador y el bus de datos.

**Microcontrolador Atmega328p**, (J. M. Hughes 2016) hace una revisión a profundidad de la arquitectura de los microcontroladores de Atmel utilizados en la plataforma Arduino en general la familia de microcontroladores ATmega tienen en común una arquitectura Harvard modificada, esto en contraposición de una arquitectura von neuman que es la comúnmente encontrada en las computadoras modernas difiere en el acceso a la memoria puesto que la arquitectura Harvard propone espacios separados para la memoria ROM de los programas y los datos mientras que la arquitectura von neuman propone un espacio de memoria ROM compartido. El microcontrolador encontrado comúnmente en los modelos más comercializados de Arduino es el ATmega328p, el cual es un microcontrolador de 8 bits con una arquitectura Harvard RISC por lo tanto el código y las variables de operación se encuentran en espacios separados.

**Arquitectura interna**, Arquitectura RISC: 131 instrucciones, 32 registros de 8 bits de propósito general, Reloj interno de 16 mips (hasta 16Mhz), EEPROM de 1kb, Voltaje de operación en rango desde 2.7v hasta 5.7v, Rango de temperatura desde -40°C hasta 125°C.

**Memoria**, Especificación: 32 Kbytes de memoria autoprogramable flash en sistema, 1 KByte de memoria EEPROM, 2 Kbytes de memoria SRAM interna, 10000 ciclos de escritura/borrado para la memoria flash, 100000 ciclos de escritura/borrado para la memoria EEPROM.

**Funciones periféricas**, las Funciones periféricas del dispositivo son controladas mediante un bus de datos de alta velocidad mediante el CPU a través de los registros de control, de los cuales se hace uso para la configuración y operación de los periféricos del sistema.

**Registros de Estado**, los registros de estado son definidos por el fabricante como pequeños espacios de memoria que guardan información acerca de la última instrucción aritmética ejecutada y estos datos se pueden utilizar para redirigir el flujo de ejecución para poder realizar operaciones condicionales.

**Puertos digitales de E/S**, los puertos del ATmega328 están como se muestra en la ilustración designados como B, C y D todos son puertos de 8 bits, cada puerto provee E/S bidireccional discreta y cuenta con resistencias internas de pull-up programables a través de los bits de un registro interno para pines específicos del puerto. Para la tarjeta Arduino los puertos son mapeados como se ilustra en la figura 12 para que su uso se oriente al control separado de cada pin haciendo distinción de los pines digitales y los pines analógicos, cabe destacar que la funcionalidad de un pin como Entrada o salida es configurada por medio de una función parte de la librería Arduino y esta configuración se hace antes de ejecutar el bucle principal del programa.

**Sensor ultrasonico hc-sr04**, el sensor HC-SR04 elegido para este proyecto mostrado en la figura 13 por su precio y sencillez de operación, es un sensor ultrasónico basado en un emisor y un receptor, para fines didácticos podemos definirlo como un micrófono y un parlante a nivel ultrasónico. Su funcionamiento está basado como se muestra en la imagen 13 en la emisión de una onda ultrasónica por medio de un pulso y la medición en el tiempo de la misma, de esta manera el receptor escucha continuamente tras enviar la señal, entonces al recibir la señal de regreso calcula la distancia a la que ha viajado la onda de sonido por medio de la constante de velocidad del sonido.

### **Ecuación de cálculo de distancia:**

$$d = (v * t)$$

### **Ecuación del cálculo de distancia para el sensor:**

$$\frac{340m}{s} \times \frac{1s}{1000000} \times \frac{100cm}{1m} = \frac{2d}{t}$$
$$d(cm) = \frac{t(us)}{59}$$

El sensor necesita recibir una señal de activación de 5v en la entrada de activación por un tiempo de al menos 10 microsegundos, con esta señal el sensor comenzará la transmisión de 8 ciclos de pulsos a una frecuencia de 40 khz en seguida el receptor del sensor queda a la espera del arribo de las señales de retorno, cuando alguna señal de retorno llegue al receptor la salida en el pin Echo se pondrá a 5v y esperará por un periodo proporcional a la distancia.

**Comunicación serial con ch340**, arduino utiliza en su esquema de comunicación un circuito integrado con capacidad para convertir el protocolo USB a un protocolo serial y comunicarse directamente con el microcontrolador del Arduino es decir el ATmega328p, al ser la tarjeta Arduino un dispositivo de hardware libre diferentes fabricantes han utilizado diferentes versiones del producto convertidor en sus modelos originales la empresa comercializa su producto utilizando los chips FTDI sin embargo los fabricantes chinos utilizan un CH340g para la misma funcionalidad y es el que se describe a continuación debido a su presencia en el Arduino nano utilizado para desarrollar el prototipo. El circuito integrado ch340g es un dispositivo desarrollado para convertir señales usb a interfaces seriales, protocolo rs232 o IrDA infrared, en el presente proyecto se utiliza en su configuración como interfaz serial en la ilustración 16 se muestra el diagrama de bloques de configuración del chip CH340 en general.

**Lineamientos de android para el desarrollo de accesorios**, (ANDROID [sin fecha]) en su documentación oficial brinda información sobre los aspectos técnicos que debe tomar en cuenta el desarrollador de dispositivos para la

plataforma con el objetivo de garantizar su funcionalidad y estabilidad en el sistema así como su compatibilidad en dispositivos con diferentes piezas de hardware, en especial se centra en la compatibilidad del desarrollo de dispositivos que no pueden funcionar como host para usb el cual tiene la función de alimentar la energía y administrar el bus para datos con el dispositivo conectado además de enumerar los dispositivos. Es necesario poner en el archivo manifest de la aplicación para brindar la información al sistema sobre el uso de la API de accesorios, el uso del intent filter y el uso necesario del archivo de filtro de dispositivos para que al conectar el dispositivo obtengamos una ventana de confirmación de un determinado tipo de dispositivo conectado.

A continuación, se describe el archivo .xml para el filtrado de dispositivos propuesto por la documentación de Android, este archivo se debe poner en la dirección *res/xml/accessory\_filter.xml* y sirve para indicar al sistema los tipos de dispositivos que pueden iniciar una comunicación con la aplicación que se necesite utilizar.

### III. METODOLOGÍA

#### 3.1 Tipo y diseño de investigación

- **Tipo de Investigación**
  - Aplicada
- **Diseño de Investigación**
  - Pre Experimental

#### 3.2 Variables y operacionalización

- **Variable dependiente:** Control de flujo de personas

- Definición conceptual:

En su libro "Crowd Science: Designing, Analyzing and Managing Crowd-Based Events" (Keith Still, Marina Papalexi, Yiyi Fan 2013), define el control de flujo como "el uso de barreras físicas, señalización, personal y otros medios para controlar y dirigir el flujo de personas a través de un espacio o un sistema de transporte".

- Definición operacional

Cumplimiento de la capacidad máxima: Se puede medir el cumplimiento de la capacidad máxima establecida. Para ello, se puede instalar el sistema en diferentes ambientes y medir la cantidad de personas que ingresan a cada ambiente antes de que el sistema restrinja el ingreso.

Velocidad de respuesta del sistema: El tiempo que tarda el sistema en responder ante un aumento de la capacidad de un ambiente.

Costo-beneficio: El impacto del sistema también puede ser evaluado desde una perspectiva de costo-beneficio. Es decir,

se debe comparar el costo de implementar el sistema con los beneficios en términos de seguridad para los usuarios.

- **Variable Independiente:** Sistema de control de aforo inteligente

- Definición conceptual:

Un sistema de control de aforo inteligente es un sistema que utiliza tecnologías avanzadas, como sensores y análisis de datos, para monitorizar y gestionar el número de personas que acceden a un espacio determinado en tiempo real. Este tipo de sistema se utiliza para controlar el aforo máximo permitido en un espacio, prevenir aglomeraciones, mejorar la seguridad y garantizar una mejor experiencia de usuario, del artículo "Intelligent People Counting System for Crowded Places using Image Processing and Deep Learning Techniques" (Ilyas, N.; Shahzad, A.; Kim 2020).

- Definición operacional:

El sistema de control de aforo inteligente está compuesto por 4 arquitecturas distintas el primero es un sistema embebido que lee la información de los sensores (HC SR 04) y se comunica con el dispositivo Android mediante el puerto usb utilizando el puerto serial, el segundo sistema es el aplicativo desarrollado para Android con la capacidad de obtener las respuestas del sensor y emitir los respectivos mensajes de alerta según su configuración además de realizar las verificaciones de autenticación y autorización, el tercero es el API de backend en la nube el cual procesa y envía respuestas a diferentes grupos de dispositivos para realizar la sincronización el cuarto es una web que permite gestionar los dispositivos y enviar la configuración además de obtener en tiempo real la información de los demás elementos del sistema.



Se puede medir usando el atributo de calidad en cuanto a la funcionalidad por la complejidad de sus partes.

### **3.3 Población, muestra y muestreo**

La población para el presente proyecto de investigación consta de los ambientes que registrarán el control del aforo y las personas que ingresan en una cola de espera.

- **Criterios de inclusión**

Se consideran como parte de la población los ambientes de las instalaciones hospitalarias del área administrativa y consultorios externos de la institución de salud pública que concentran mayor concurrencia.

Para la segunda población se considera los pacientes que desean ingresar a los ambientes y tienen que formar colas para que los dejen pasar a realizar sus consultas.

- **Criterios de exclusión**

Se excluyen los ambientes que no permiten registrar visitas de atención y los ambientes de hospitalización debido a las restricciones horarias y de carácter normativo de la institución además de los ambientes de uso administrativo por los trabajadores.

Se excluye de la segunda población a las personas que forman parte del personal que labora en las instalaciones y tienen vía libre para entrar y salir sin pasar por el control de concurrencia.

#### **Población 1 ( $N_1$ )**

Se considera la población 2 a la cantidad de personas que ingresan a los diferentes ambientes que reciben consultas de las personas por diferentes motivos.

$$N_1 = \frac{130 \text{ Pacientes atendidos}}{\text{dia}} \times \frac{5 \text{ dias}}{\text{semana laboral}}$$

$$N_1 = 650 \text{ pacientes}$$

<b>Especialidad</b>	<b>Cantidad de pacientes (promedio)</b>
Cardiología	24%
Nefrología	19%
Neumología	38%
Oftalmología	6%
Otorrinolaringología	5%
<i>Medicina general</i>	8%
<b>TOTAL</b>	<b>100%</b>

**Tabla 1.** Porcentaje de Atención, elaborado por el autor

### **Muestra 1 ( $n_1$ )**

Dado que la Población es mayor que 30, entonces se aplicará la siguiente formula estadística considerando que nuestro universo es de 650 con un nivel de confianza del 95% ( $Z= 1.960$ ) y un margen de error del 7% (E) aplicaremos la formula:

$$n = \frac{N * Z_{\alpha}^2 * p * q}{e * (N - 1) + Z_{\alpha}^2 * p * q} = 145.38$$

- **Muestreo**

Se utilizará un muestreo probabilístico aleatorio simple.

### **3.4 Técnicas e instrumentos de recolección de datos**

- Técnicas
  - Observación
  - Análisis documental
- Instrumentos
  - Ficha de Observación
  - Ficha de Datos

### **3.5 Procedimientos**

- Mejorar el cumplimiento de las políticas de control de aforo:

Para mejorar el cumplimiento de las políticas de control de aforo en la institución se realizará la implementación y configuración del sistema en el ambiente con mayor concurrencia de personas de la institución y el sistema automáticamente alertará si se ha alcanzado el límite máximo de capacidad para el espacio y emitirá una alerta sonora que avisará al personal que el espacio está a su máxima capacidad, en el sentido de las políticas de control de aforo el sistema estará preparado para modificar su configuración y a través de esta modificar su operación en cuanto a la capacidad soportada en cada ambiente.

- Mejorar los tiempos de espera al actualizar los datos en tiempo real:

Al realizar el control de la cola de forma manual mediante un encargado el sistema actual consiste en contar y permitir el ingreso de  $n$  en  $n$  esto no es lo más adecuado dado que se tiene que esperar que salgan  $n$  personas para que otras  $n$  puedan ingresar, el llevar el control en pasos de 1 elemento de forma confiable permite un tráfico mucho más fluido de las personas de la cola.

- Ahorrar tiempos y costos del personal:

En el sistema manual de la institución se requiere de 3 personas que controlen el ingreso y estén constantemente verificando que se estén respetando las políticas de aforo en la institución en 2 turnos de 8 horas haciendo un total de 6 personas para esta labor las cuales podrían reducirse a 1, liberando al personal para otras tareas más críticas.

### **3.6 Método de análisis de datos**

El procesamiento y análisis de datos de datos en la investigación se realizará utilizando el método estadístico a nivel descriptivo como inferencial.

La estadística descriptiva abarca una variedad de técnicas y métodos que se utilizan para resumir, organizar y presentar datos de una muestra o población. Su objetivo principal es proporcionar una descripción completa y objetiva de los datos, lo que se logra a través de medidas como la tendencia central, la dispersión y la asimetría, entre otras. Es fundamental para la comprensión y la interpretación de los datos, y se aplica ampliamente en áreas como la investigación, la medicina, y la economía.

La estadística inferencial comprende un conjunto de técnicas y métodos que se emplean para extraer conclusiones e inferencias acerca de una población a partir de una muestra. Su relevancia es crucial para la toma de decisiones basadas en datos y para la investigación científica, ya que permite a los investigadores realizar predicciones y generalizaciones sobre una población más amplia a partir de una muestra, al mismo tiempo que se evalúa la probabilidad de que los resultados obtenidos sean significativos y no estén influenciados por el azar.

### **3.7 Aspectos éticos**

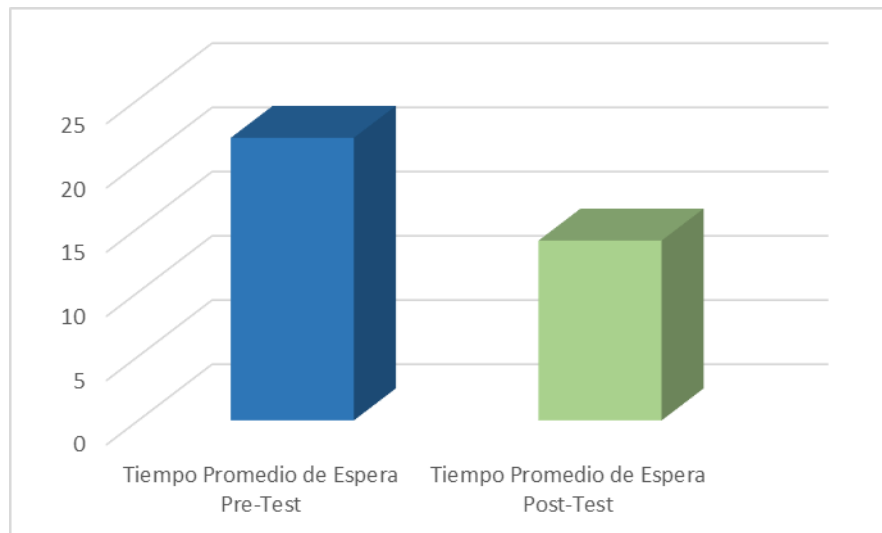
En los capítulos anteriores del presente proyecto se han realizado las debidas referencias y citas bibliográficas a cada autor del cual el presente proyecto se ha nutrido respetando siempre la autoría de cada investigador y utilizando el sistema de citas bibliográficas de la norma ISO-690 por disposición de la universidad, además con el uso del sistema de turnitin se garantiza que el contenido del proyecto es original en cuanto a las normas de la universidad, cumpliendo para las tesis de pregrado el porcentaje máximo aceptado del 30% encontrándose el presente trabajo por debajo de esta cifra. Cumpliéndose además con lo estipulado en el código de ética de la universidad Cesar Vallejo en cuanto al rigor científico, la honestidad, la competencia profesional y la responsabilidad durante el proceso de ejecución de la presente investigación.

#### IV. RESULTADOS

- **Análisis descriptivo**

- Para el indicador uno comprendido como el tiempo promedio que esperan las personas por día se hizo una medición de un turno de 4 horas en las cuales se registró en la ficha de observación el tiempo promedio de cada persona en la cola utilizando un cronometro y esperando a que le permitieran ingresar.

Durante la medición pretest se registró un tiempo promedio de 22 minutos en cola antes de ser atendidos y durante la medición post-test con el sistema de control se registró una mejora significativa de los tiempos de espera como se detalla a continuación:

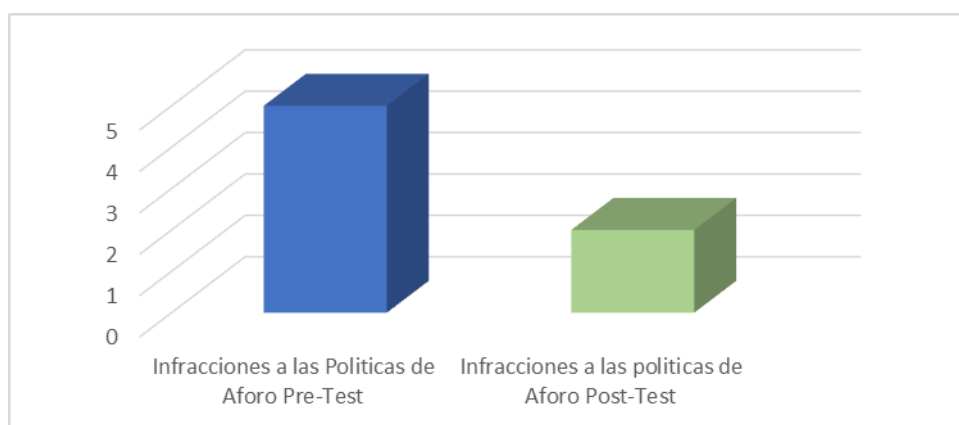


**Figura 1.** Resultado de tiempo promedio de espera. elaborado por el autor.

Estos resultados fueron obtenidos en el ingreso a la sala de espera de consulta externa por ser el área más visitada por pacientes en la institución de salud.

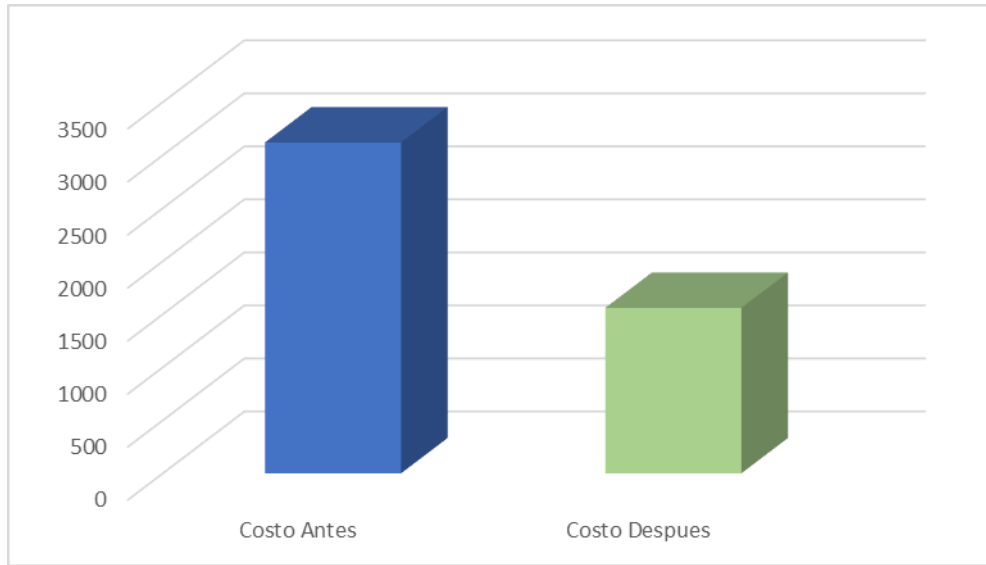
- Para el segundo indicador se registró durante el tiempo de medición en el escenario pretest y el escenario post-test durante el mismo turno de 4 horas, la cantidad de infracciones a la política de aforo establecida por la institución.

Por lo tanto según van entrando los pacientes muchas veces demoran en ser atendidos afectando esto el tiempo de espera de otros pacientes motivo por el cual el personal permite el ingreso de algunas personas extra en determinado momento bajo estas condiciones se ha medido la cantidad de veces por turno en las que se infringen las políticas de aforo y se ha promediado los resultados obtenidos durante la recolección de datos pretest y post test obteniendo una mejora en cuanto a la infracción de las políticas debido a que el sistema emite una alerta sonora y tiene un componente visual que permite conocer en todo momento la capacidad actual del ambiente sujeto a medición.



**Figura 2.** Resultado infracciones a las políticas de aforo elaborado por el autor

- Para evidenciar el impacto en el tercer indicador se ha valorado una fracción del ingreso percibido por los 6 trabajadores de la institución durante un turno y se ha sumado en pretest y luego de la implantación del sistema se evidencia de que solo es necesario 3 personas para realizar el control.



**Figura 3.** Resultado costos. Elaborado por el autor.

▪ **Análisis inferencial**

**Indicador:** Reducción de los Tiempos de espera en cola (Shapiron Wilk)

Prueba de Normalidad

Shapiro Wilk			
	Estadístico	gl	Sig.
TE_DIFERENCIA	0.974	15	0.917

**Tabla 2.** Prueba de Normalidad indicador 1 elaborado por el autor

Para la prueba de normalidad es necesario definir los siguiente:

H0: Los datos siguen una distribución normal ( $p < 0.05$ )

H1: Los datos no siguen una distribución normal ( $p \geq 0.05$ )

Se define el siguiente criterio para el análisis de la prueba de normalidad:

- Si  $P < 0.05$  entonces se rechaza la hipótesis nula y se acepta la hipótesis alterna.
- Si  $P \geq 0.05$  entonces se acepta la hipótesis nula y se rechaza la hipótesis alterna.

Decisión:

En vista de los resultados arrojados por el estadístico Shapiro Wilk se evidencia que el valor p es mayor a 0.05, habiendo obtenido un valor de significancia de 0.917 se acepta la hipótesis alternativa con respecto a la normalidad de los datos por lo tanto se indica que nuestros datos no siguen una distribución normal. En razón a lo anterior se aplicará una prueba no paramétrica.

**Indicador:** Reducción de la tasa de infracciones al aforo establecido por hora:

Prueba de Normalidad

Shapiro Wilk			
	Estadístico	gl	Sig.
DIFERENCIA	0.963	15	0.740

**Tabla 3.** Prueba de normalidad indicador 2, elaborado por el autor.



Para la prueba de normalidad con respecto a los datos de la cantidad de infracciones de aforo es necesario definir los siguiente:

H0: Los datos siguen una distribución normal ( $p < 0.05$ )

H1: Los datos no siguen una distribución normal ( $p \geq 0.05$ )

Se define el siguiente criterio para el análisis de la prueba de normalidad:

- Si  $P < 0.05$  entonces se rechaza la hipótesis nula y se acepta la hipótesis alterna.
- Si  $P \geq 0.05$  entonces se acepta la hipótesis nula y se rechaza la hipótesis alterna.

Decisión:

La prueba de Shapiro Wilk para este caso ha arrojado un valor de 0.740 de significancia (sig. 0.740) por lo tanto corresponde aceptar la hipótesis alterna ( $p \geq 0.05$ ) y rechazar la hipótesis nula, a raíz de estos resultados podemos evidenciar que los datos no siguen una distribución normal. A efecto de estos resultados corresponde ejecutar una prueba no paramétrica para el análisis de la prueba de hipótesis de estos datos.

### **Indicador: Reducción de costos en personal**

Prueba de Normalidad Costos del personal

	Shapiro Wilk		
	Estadístico	gl	Sig.
CP_DIFERENCIA	0.561	15	0.000

**Tabla 4.** Prueba de normalidad indicador 3, elaborado por el autor

Para la prueba de normalidad con respecto a los datos del costo de mantener el sistema de forma manual es necesario definir los siguiente:

H0: Los datos siguen una distribución normal ( $p < 0.05$ )

H1: Los datos no siguen una distribución normal ( $p \geq 0.05$ )

Se define el siguiente criterio para el análisis de la prueba de normalidad:

- Si  $P < 0.05$  entonces se rechaza la hipótesis nula y se acepta la hipótesis alterna.
- Si  $P \geq 0.05$  entonces se acepta la hipótesis nula y se rechaza la hipótesis alterna.

Decisión:

Según a prueba de normalidad de Shapiro Wilk nuestros datos siguen una distribución normal en este caso dado que su significancia ha arrojado el valor de 0.000 por lo tanto se aplicará una prueba estadística paramétrica.

▪ **Prueba de hipótesis**

- **Prueba de Hipótesis para el Indicador 1** – Tiempos de espera en cola.

H0 La implementación del sistema no tiene un impacto significativo reduciendo el tiempo de espera en una cola para una institución de salud en puno 2022.

H1 La implementación del sistema tiene un impacto significativo reduciendo el tiempo de espera en una cola para una institución de salud en puno 2022.

<b>.Estadísticos de Prueba</b>	
	TE_DESPUES - TE_ANTES
Z	-2.797 <sup>b</sup>
Sig. asintótica(bilateral)	0.005

**Tabla 5.** Prueba de hipótesis indicador 1, elaborado por el autor.

- Si  $P < 0.05$  entonces se rechaza la hipótesis nula y se acepta la hipótesis alterna.
- Si  $P \geq 0.05$  entonces se acepta la hipótesis nula y se rechaza la hipótesis alterna.

Por lo tanto, en razón a la evidencia obtenida sobre los datos utilizando la prueba de Wilconxon podemos afirmar que la implementación del sistema tiene un impacto significativo sobre el indicador de los tiempos de espera en el control de la concurrencia en una institución de salud de puno en el año 2022 dado que el valor de significancia para  $p < 0.05$  habiendo obtenido un valor de significancia (Sig) de 0.005.

- **Prueba de Hipótesis para el indicador** cantidad de infracciones a las políticas de aforo

H0 La implementación del sistema no tiene un impacto significativo reduciendo la cantidad de infracciones que se producen a las políticas de aforo para una institución de salud en puno 2022.

H1 La implementación del sistema tiene un impacto significativo reduciendo la cantidad de infracciones que se producen a las políticas de aforo para una institución de salud en puno 2022.

Estadísticos de Prueba	
	TI_DESPUES - TI_ANTES
Z	-3,192 <sup>b</sup>
Sig. asintótica(bilateral)	,001

**Tabla 6.** Prueba de hipótesis del indicador 2 elaborado por el autor

- Si  $p < 0.05$  entonces se rechaza la hipótesis nula y se acepta la hipótesis alterna.
- Si  $p \geq 0.05$  entonces se acepta la hipótesis nula y se rechaza la hipótesis alterna.

Decisión:

Para el indicador de cantidad de infracciones a la normativa de aforo propuesta por la institución se obtuvo un valor de  $p < 0.05$  (sig = 0.001) por lo tanto se rechaza la hipótesis nula y se acepta la hipótesis alterna en cuyo caso corresponde inferir de que la implementación del sistema de control de aforo tiene un impacto significativo reduciendo la cantidad de infracciones a las políticas de aforo.

- **Prueba de Hipótesis para el indicador – Reducción de costos del personal**

- H0 La implementación del sistema no tiene un impacto significativo reduciendo los costos en el personal involucrado en el control de aforo para una institución de salud en puno 2022.
- H1 La implementación del sistema tiene un impacto significativo reduciendo los costos en el personal involucrado en el control de aforo para una institución de salud en puno 2022..

	t	gl	Sig. (bilateral)
CP_ANTES - CP_DESPUES	14.666	14	0.000

**Tabla 7.** Prueba de hipótesis del indicador 3, elaborado por el autor

- Si  $p < 0.05$  entonces se rechaza la hipótesis nula y se acepta la hipótesis alterna.
- Si  $p \geq 0.05$  entonces se acepta la hipótesis nula y se rechaza la hipótesis alterna.

Decisión:

Ante la evidencia mostrada a raíz de la prueba estadística T-Student se puede afirmar que la implementación del sistema de control de aforo tiene un impacto significativo reduciendo los costos para el personal involucrado en el control de aforo de una institución de salud en Puno para el año 2022.

## V. DISCUSIÓN

Se ha definido como indicador el tiempo de espera en la cola antes de ingresar al ambiente correspondiente y se ha conseguido reducir el tiempo desde un promedio de 22 minutos en cola hasta 14 minutos con la implementación del sistema esto consiste en una reducción de 8 minutos en promedio para el sistema, la mejora en la gestión de la cola se debe a la implementación del dispositivo electrónico que permite un conteo preciso de entradas y salidas a través de una barrera ultrasónica en este caso en comunicación con un dispositivo Android, la posibilidad de controlar un flujo de personas con el desarrollo de un sistema de control y monitoreo de aforo ha sido abordada con anterioridad por (Ballester Mainar 2021) cuya solución ha sido sometida a pruebas que indican que el desarrollo e implementación de un dispositivo para el control de aforo es posible mediante la combinación de múltiples sistemas de transmisión y centralización de datos además guarda relación en cuanto al contexto en el que se desarrolla su proyecto, entre los desafíos más relevantes en este tipo de proyectos está la elección de las tecnologías y el tipo de planeación que requieren estos proyectos dado que pueden enfocarse en distintas tecnologías y conseguir el mismo objetivo en común que es el poder tener información en tiempo real sobre el aforo de las personas en ambientes que requieren restricción.

La propuesta de sistema de Safe-Events (Gallego Ferrer 2016) en su investigación aborda el mismo objetivo de mejorar los procesos de aforo de eventos por medio de la implementación de un sistema de brazaletes que son escaneados con tecnología RFID aunque su sistema tiene la ventaja de ser más preciso con la información manejada sobre cada entidad que ingresa al sistema esta requiere que se le entregue un dispositivo que la persona pueda portar y que tendría que ser de uso personal por lo tanto sale de los márgenes de un sistema sencillo que solo pueda gestionar entradas y salidas de personas son necesariamente conocer más datos.

La implementación de sistemas de control de aforo puede mejorar significativamente los tiempos de espera. Por lo tanto, la reducción de los tiempos de espera podría ser explicada por la mejora en la eficiencia del

proceso de ingreso, la automatización del sistema, la optimización del personal y el mejoramiento de la gestión de la institución.

## VI. CONCLUSIONES

1. El sistema de control de aforo basado en un microcontrolador Arduino y una comunicación con un dispositivo Android a través de la nube busca como objetivo principal evaluar el impacto de la implementación y uso de esta tecnología al servicio de una mejor gestión de espacios controlados y medidas de aforo, este objetivo primario y los indicadores elegidos para medir este impacto han permitido llegar a las siguientes conclusiones:
2. Se ha reducido el tiempo de espera en cola de las personas para ingresar al ambiente en el que el sistema ha sido implementado, ello implica una mejora en promedio del tiempo de espera gracias a la automatización del proceso frente al sistema manual de conteo de personas.
3. Se ha conseguido reducir las infracciones a las políticas de aforo de la institución gracias a las alertas sonoras e indicadores visuales que presenta el sistema cuya información previa a la implementación del sistema arrojaban valores que oscilaban entre 4 a 8 infracciones a las políticas de aforo con un promedio de 5 infracciones por día y bajo el mismo escenario se ha realizado la medición post implementación del sistema y se ha dado seguimiento obteniendo un promedio de 2 infracciones a las políticas de aforo por día, evidenciando una reducción en promedio de 3 infracciones diarias.
4. Con respecto al indicador de reducción de costos de personal, se ha hecho seguimiento de la cantidad de personas que son necesarias para gestionar el sistema manual y verificar constantemente que no se esté rebasando las normativas de aforo y se ha conseguido evidenciar de que en lugar de dedicar 6 personas a la labor de controlar los aforos luego de la implementación del sistema solo son requeridas 3 personas reduciendo de esta manera el coste de personal en la institución.



## **VII. RECOMENDACIONES**

En general se recomienda que se reemplace la forma en que se comunica el sensor con el dispositivo móvil dado que para el proyecto se ha utilizado comunicación serial en base al puerto USB del equipo.

Además, se recomienda optimizar un poco más la frecuencia de lectura de los sensores dado que se experimentaron algunos problemas en fase de desarrollo con respecto a la absorción de los materiales de la onda ultrasónica, esto se pudo corregir mejorando el algoritmo y la frecuencia en que los dispositivos efectúan las operaciones de envío de onda y lectura.

Sin embargo, para el correcto funcionamiento es necesario calibrar correctamente el retorno de la onda ultrasónica considerando una superficie plana desde la que rebota.

## REFERENCIAS

- ANDROID, [sin fecha]. Descripción general sobre el modo de accesorio USB. [en línea]. Disponible en:  
<https://developer.android.com/guide/topics/connectivity/usb/accessory?hl=es-419>.
- BALLESTER MAINAR, F., 2021. Diseño y desarrollo de un sistema de monitorización y limitación en tiempo real del aforo en espacios cerrados. [en línea]. S.l.: Universitat Politècnica de València. [Consulta: 30 junio 2021].  
Disponible en: <https://riunet.upv.es/handle/10251/158740>.
- CALVO REBOLLO, E., 2017. Universidad de Jaén. [en línea]. S.l.: [Consulta: 1 julio 2021]. Disponible en: <http://tauja.ujaen.es/jspui/handle/10953.1/6660>.
- GALLEGO FERRER, M.M., 2016. Safe Events: control de aforos mediante RFID. [en línea]. S.l.: Universitat Politècnica de València. [Consulta: 30 junio 2021].  
Disponible en: <https://riunet.upv.es/handle/10251/89155>.
- HEATH, S., 2003. *Embedded systems design*. second edi. S.l.: Newnes. ISBN 978-0-7506-5546-0.
- ILYAS, N.; SHAHZAD, A.; KIM, K., 2020. Convolutional-Neural Network-Based Image Crowd Counting: Review, Categorization, Analysis, and Performance Evaluation. *Sensors*, DOI <https://doi.org/10.3390/s20010043>.
- J. M. HUGHES, 2016. *Arduino: A Technical Reference* [en línea]. 1 st. S.l.: o really. [Consulta: 5 julio 2021]. ISBN 9781491934494, 1491934492.  
Disponible en: <https://www.vitalsource.com/pe/en-us/products/arduino-a-technical-reference-j-m-hughes-v9781491934494>.
- KADAV, A. y SWIFT, M.M., 2012. Understanding modern device drivers. *International Conference on Architectural Support for Programming Languages and Operating Systems - ASPLOS*, pp. 87-98. DOI 10.1145/2150976.2150987.
- KEITH STILL, MARINA PAPALEXI, YIYI FAN, D.B., 2013. Place crowd safety, crowd science? Case studies and application. *Journal of Place Management and Development*, vol. 13, pp. 385-417. DOI <https://doi.org/10.1108/JPMD-10->

2019-0090.

MANUEL BÁEZ, ÁLVARO BORREGO, J.C., LUIS CRUZ, MIGUEL GONZÁLEZ, F.H., DAVID PALOMERO, JOSÉ RODRÍGUEZ DE LLERA, D.S. y MARIAM SAUCEDO, PILAR TORRALBO, Á.Z., 2016. *Introducción a Android*. S.I.: E.M.E. Editorial ©. ISBN 978-84-96285-39-5.

MOZILLA, [sin fecha]. Generalidades del protocolo HTTP - HTTP | MDN. [en línea]. [Consulta: 14 julio 2021]. Disponible en: <https://developer.mozilla.org/es/docs/Web/HTTP/Overview>.

PURDUM, J., 2015. Introduction. *Beginning C for Arduino*. S.I.: Apress, pp. 1-22.

RUSSELL, D.J., 2010. *Introduction to embedded systems: Using ANSI C and the Arduino development environment*. S.I.: Morgan & Claypool. ISBN 9781608454983.

SÁNCHEZ, A., ALBERTO, A., GODOY, A. y DANTE, M., 2017. Desarrollo de una red de sensores inalámbricos para la detección de automóviles en estacionamientos privados utilizando sensores Xbee DigiMesh. *Universidad Peruana de Ciencias Aplicadas (UPC)* [en línea]. S.I.: Universidad Peruana de Ciencias Aplicadas (UPC). [Consulta: 28 junio 2021]. Disponible en: <http://hdl.handle.net/10757/621900>.

TAN, C.J., MOHAMAD-SALEH, J., ZAIN, K.A.M. y AZIZ, Z.A.A., 2017. Review on firmware. *ACM International Conference Proceeding Series*. S.I.: Association for Computing Machinery, pp. 186-190. ISBN 9781450352895. DOI 10.1145/3132300.3132337.

## ANEXOS

### Anexo 1 - Matriz de consistencia de la investigación

Título: “Sistema inteligente para el Control de aforo en Institución de Salud Pública, Puno 2023”

Autor: Ruelas Paredes, Rusbel Oliver

Problema	Objetivo	Hipótesis	Variable
<p>General:</p> <p>¿En qué medida el desarrollo de un sistema un sistema para el control de aforo en espacios públicos concurridos impacta en el control de flujo de personas en la institución de salud pública en Puno en el año 2022?</p>	<p>General:</p> <p>Mejorar el control de concurrencia de personas en institución de salud pública mediante el desarrollo de un sistema para el control de aforo en espacios públicos concurridos en el año 2022.</p>	<p>Alternativa (H<sub>a</sub>):</p> <p>“El desarrollo de un sistema de control de aforo inteligente si mejora el control de flujos de personas en la institución de salud pública de Puno el año 2022”</p>	<p>Independiente:</p> <p>Sistema de control de aforo inteligente.</p>
<p>Específicos:</p> <p>Se cumple parcialmente las políticas de restricción de aforo.</p> <p>El sistema manual de contar las personas por el numero de personas que entra y esperar a q el mismo numero salga no es lo suficientemente rápido ni efectivo.</p> <p>El sistema manual para controlar el aforo es costoso en tiempo y personal</p>	<p>Específicos:</p> <p>Mejorar el cumplimiento de las políticas de control de aforo.</p> <p>Mejorar los tiempos de espera al actualizar los datos en tiempo real.</p> <p>La automatización del sistema permitirá ahorrar tiempos y costos del personal</p>	<p>Nula (H<sub>0</sub>):</p> <p>“El sistema de control de aforo no mejora significativamente el control del flujo de personas en la institución de salud pública de puno el año 2022”</p>	<p>Dependiente:</p> <p>Control de flujo de personas.</p>

Metodología			
<p>Tipo de investigación:</p> <p>Aplicada</p>	<p>Población (N):</p> $N_1 = \frac{130 \text{ Pacientes atendidos}}{\text{dia}} \times \frac{5 \text{ dias}}{\text{semana laboral}}$ $N_1 = 650 \text{ pacientes}$	<p>Técnicas de recolección de datos:</p> <p>Observación</p>	<p>Método de análisis de datos:</p> <p>Estadística descriptiva</p> <p>Estadística inferencial</p> <p>Deductivo (enfoque cuantitativo)</p>
<p>Diseño de investigación:</p> <p>Pre experimental</p>	<p>Muestra (n):</p> $n = \frac{N * Z_{\alpha}^2 * p * q}{e * (N - 1) + Z_{\alpha}^2 * p * q} = 145.38$	<p>Instrumentos de recolección de datos:</p> <p>Ficha de observación</p>	<p>Aspectos éticos:</p> <p>Se respetará el derecho a la propiedad intelectual (Originalidad de la investigación - Reporte Turnitin).</p> <p>Se tomará en cuenta el Código de ética vigente de la Universidad César Vallejo.</p> <p>Adicionalmente, se usará para la redacción de la investigación el Sistema de Normas ISO-690.</p>

## Anexo 2 - Matriz de operacionalización de variables

Variable	Definición conceptual	Definición operacional	Dimensión subvariable	Indicador	Escala de medición
Independiente: Sistema de control de aforo inteligente.	"Un sistema de control de aforo inteligente es un sistema que utiliza tecnologías avanzadas, como sensores y análisis de datos, para monitorizar y gestionar el número de personas que acceden a un espacio determinado en tiempo real." (Ilyas, N.; Shahzad, A.; Kim 2020).	Se mide utilizando el aspecto de calidad de funcionalidad	calidad	funcionalidad	ordinal
Dependiente: Control de flujo de personas.	"el uso de barreras físicas, señalización, personal y otros medios para controlar y dirigir el flujo de personas a través de un espacio o un sistema de transporte" (Keith Still, Marina Papalexi, Yiyi Fan 2013).	Se dividirá en 3 dimensiones considerando el tiempo, efectividad y costo-beneficio	Tiempo	Tiempo que demora en actualizar el registro de control de personas.	Razón
				Tasa de infracciones a la restricción de aforo por día	Razón
			Costo - Beneficio	Costo de implementación y manutención del sistema por quincena	Razón

### Anexo 3 - Método de juicio experto

Apellidos y nombres del experto: Torres Villanueva Marcelino

Título profesional y/o Grado académico: Ingeniero de Sistemas - Maestro.

Fecha: 19/12/2022

Título del proyecto de investigación:

“Sistema inteligente para el Control de aforo en Institución de Salud Pública, Puno 2023”

Autor: Ruelas Paredes Rusbel Oliver

#### Evaluación de la metodología de desarrollo de un sistema inteligente de aforo

Mediante el Método de juicio experto, Usted tiene la facultad de calificar las metodologías involucradas, mediante unas series de criterios con puntuaciones especificadas al final de la tabla. Así mismo le exhortamos en la correcta determinación de la metodología para desarrollar la solución propuesta en el presente proyecto de investigación y, también si hubiese algunas sugerencias:

Ítem	Criterios	Metodologías		
		RUP	SCRUM	XP
1	Tiempo de desarrollo	1	3	2
2	Información	3	3	2
3	Requerimientos	1	3	2
4	Complejidad	1	3	2
5	...	...	...	...
Total		6	12	8

La escala a evaluar es de: 1 - Malo, 2 - Regular, 3 - Bueno

Sugerencias:

Firma del experto

#### Criterios de evaluación de las metodologías propuestas

Ítem	Criterio	Descripción
1	Tiempo de desarrollo	Es el tiempo que toma el desarrollo completo del software.
2	Información	Es la cantidad de información disponible sobre la metodología.
3	Requerimientos	Es la cantidad de requerimientos que exige la metodología.
4	Complejidad	Es el nivel de abstracción del estudio de la metodología.
5	...	...

Fuente: Elaboración propia

### Anexo 3 - Método de juicio experto

Apellidos y nombres del experto: Guevara Ruíz, Ricardo Manuel

Título profesional y/o Grado académico: Ingeniero de Sistemas - Maestro.

Fecha: 16/12/2022

Título del proyecto de investigación:

“Sistema inteligente para el Control de aforo en Institución de Salud Pública, Puno 2023”

Autor: Ruelas Paredes Rusbel Oliver

#### Evaluación de la metodología de desarrollo de un sistema inteligente de aforo

Mediante el Método de juicio experto, Usted tiene la facultad de calificar las metodologías involucradas, mediante unas series de criterios con puntuaciones especificadas al final de la tabla. Así mismo le exhortamos en la correcta determinación de la metodología para desarrollar la solución propuesta en el presente proyecto de investigación y, también si hubiese algunas sugerencias:

Ítem	Criterios	Metodologías		
		RUP	SCRUM	XP
1	Tiempo de desarrollo	1	3	2
2	Información	3	3	2
3	Requerimientos	1	3	2
4	Complejidad	1	1	2
5	...	...	...	...
Total		<b>6</b>	<b>10</b>	<b>8</b>

La escala a evaluar es de: **1** - Malo, **2** - Regular, **3** - Bueno

Sugerencias:

Firma del experto

#### Criterios de evaluación de las metodologías propuestas

Ítem	Criterio	Descripción
1	Tiempo de desarrollo	Es el tiempo que toma el desarrollo completo del software.
2	Información	Es la cantidad de información disponible sobre la metodología.
3	Requerimientos	Es la cantidad de requerimientos que exige la metodología.
4	Complejidad	Es el nivel de abstracción del estudio de la metodología.
5	...	...

Fuente: Elaboración propia



### Anexo 3 - Método de juicio experto

Apellidos y nombres del experto: Agreda Gamboa, Everson David

Título profesional y/o Grado académico: Ingeniero de Sistemas - Doctor.

Fecha: 20/12/2022

Título del proyecto de investigación:

"Sistema inteligente para el Control de aforo en Institución de Salud Pública, Puno 2023"

Autor: Ruelas Paredes Rusbel Oliver

#### Evaluación de la metodología de desarrollo de un sistema inteligente de aforo

Mediante el Método de juicio experto, Usted tiene la facultad de calificar las metodologías involucradas, mediante unas series de criterios con puntuaciones especificadas al final de la tabla. Así mismo le exhortamos en la correcta determinación de la metodología para desarrollar la solución propuesta en el presente proyecto de investigación y, también si hubiese algunas sugerencias:

Ítem	Criterios	Metodologías		
		RUP	SCRUM	XP
1	Tiempo de desarrollo	1	1	2
2	Información	3	3	2
3	Requerimientos	1	2	2
4	Complejidad	1	3	2
5	...	...	...	...
Total		6	9	8

La escala a evaluar es de: 1 - Malo, 2 - Regular, 3 - Bueno

Sugerencias:

Firma del experto

#### Criterios de evaluación de las metodologías propuestas

Ítem	Criterio	Descripción
1	Tiempo de desarrollo	Es el tiempo que toma el desarrollo completo del software.
2	Información	Es la cantidad de información disponible sobre la metodología.
3	Requerimientos	Es la cantidad de requerimientos que exige la metodología.
4	Complejidad	Es el nivel de abstracción del estudio de la metodología.
5	...	...

Fuente: Elaboración propia

## Anexo 4. Instrumentos de recolección de datos

Fecha de observación:	1/10/2022
Hora de inicio de la observación:	8:00 a.m.
Hora de finalización de la observación:	1:00 p.m.

### FICHA DE OBSERVACIÓN

Objetivo: Observar el comportamiento de los usuarios y la utilización de los diferentes ambientes del establecimiento de salud para identificar patrones y tendencias en la afluencia y el control de aforo.

Ambiente observado: Sala de espera del servicio de consulta externa.

Observador: Rusbel Ruelas

<b>Comportamiento de los usuarios:</b>	
Total de usuarios ingresados:	32
Total de usuarios egresados:	31

Tiempo de permanencia promedio:	22 minutos
Momentos de mayor afluencia:	entre 10:15 a.m. y 10:30 a.m.
Momentos de menor afluencia:	entre 11:10 am. y 11:30 am.
Tasa de ocupación promedio:	85%

<b>Comportamiento del personal:</b>	
Total de personal de atención:	3 personas
Tiempo de respuesta promedio:	3 minutos
Tiempo de atención promedio:	8 minutos
Tiempo de descanso promedio:	5 minutos
Tiempo de espera promedio del personal:	2 minutos

#### Observaciones adicionales:

-Se observó que el personal de atención realiza la medición de temperatura a todos los usuarios antes de permitir su ingreso.

-Se observó que en ocasiones se producen aglomeraciones en la entrada y salida de la sala de espera, lo que podría dificultar el control de aforo.

-Se observó que algunos usuarios no respetan la distancia mínima de seguridad y se agrupan en áreas comunes del establecimiento.

## Anexo 7. Tabla de datos

Tabla de Tiempo Promedio de Espera datos resumidos:

Dias	T1	T2	Diferencia
1	20	18	2
2	13	11	2
3	9	16	-7
4	21	18	3
5	30	15	15
6	22	12	10
7	18	20	-2
8	31	13	18
9	25	19	6
10	19	10	9
11	22	8	14
12	36	15	21
13	24	12	12
14	13	13	0
15	30	16	14
Promedio	22.2	14.4	

### Tiempos en Minutos

T1 = Tiempo de espera en cola pretest

T2 = Tiempo de espera en cola post-test

Tiempo Promedio de Espera Pre-Test 22

Tiempo Promedio de Espera Post-Test 14

Tabla : Promedio de Infracciones por día

Dias	T1	T2	Diferencia
1	8	0	8
2	3	0	3
3	5	3	2
4	8	2	6
5	5	5	0
6	3	0	3
7	8	3	5
8	6	2	4
9	5	5	0
10	9	4	5
11	4	0	4
12	3	1	2
13	7	3	4
14	8	4	4
15	6	5	1
Promedio	5	2	

**Infracciones por Día**

Infracciones a las Políticas de Aforo Pre-Test 5

Infracciones a las políticas de Aforo Post-Test 2

Tabla Indicador 3 Costo de trabajadores por día.

Días	T1	T2	Diferencia
1	240	120	120
2	240	120	120
3	240	120	120
4	240	120	120
5	240	120	120
6	120	60	60
7	120	60	60
8	240	120	120
9	240	120	120
10	240	120	120
11	240	120	120
12	240	120	120
13	120	60	60
14	120	60	60
15	240	120	120
Total	208	104	

**Promedio de Costo en 15 días en Soles**

Costo Antes 3120

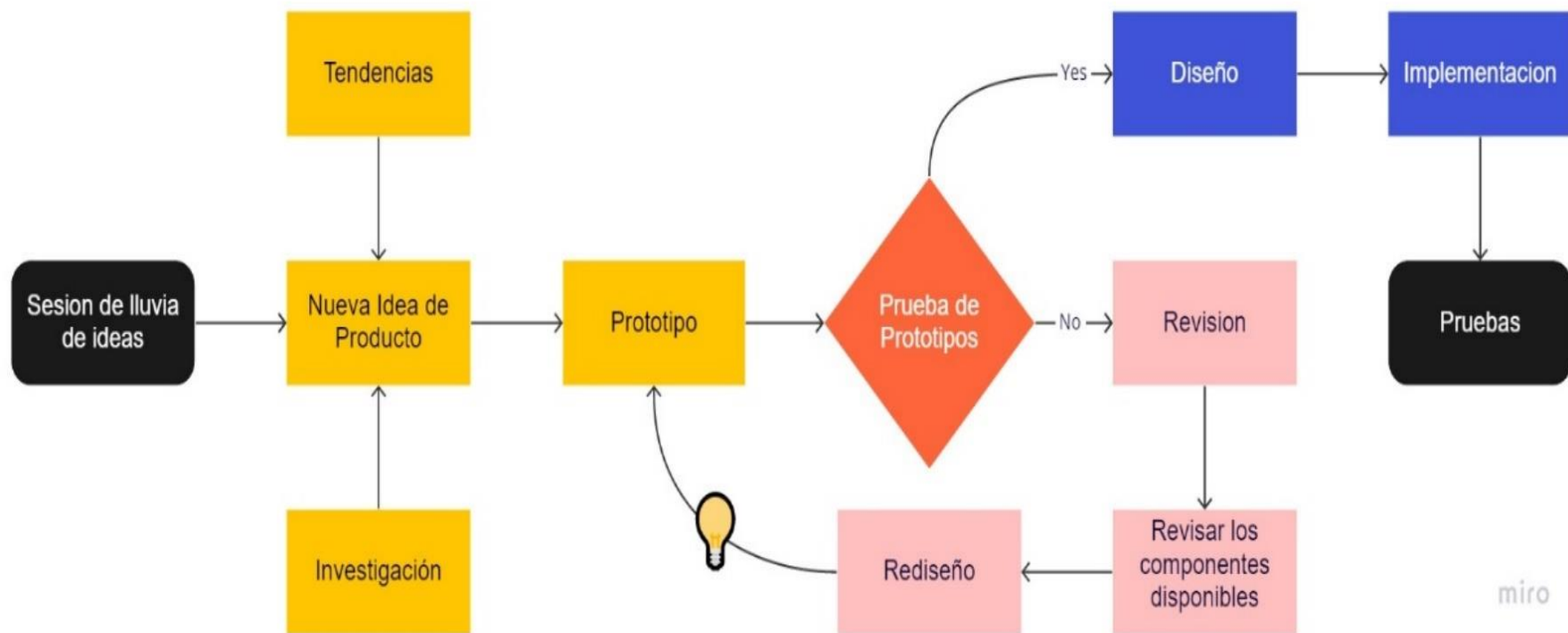
Costo Despues 1560

## **Anexo 8. Solución propuesta**

### Fase de Concepto

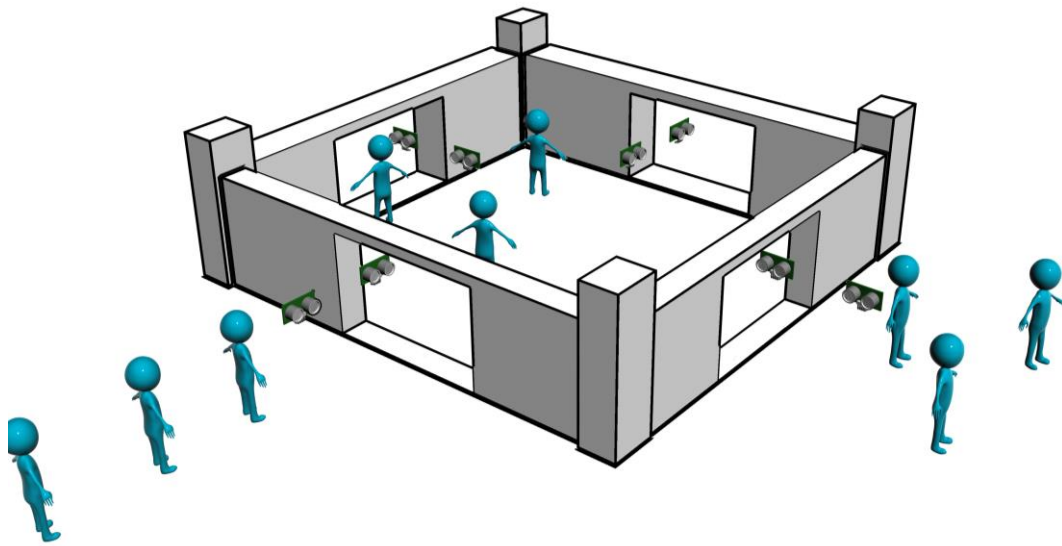
Teniendo en cuenta el problema y situación específica para abordarlo se da comienzo a las primeras caracterizaciones del sistema basados en sesiones de lluvia de ideas en las cuales surgen las primeras propuestas de organización del sistema, en esta etapa se prueba y se refina los componentes a formar parte del sistema para probar mínimamente una viabilidad del producto con respecto a la integración de sus componentes y la posibilidad económica de adquirirlos, además se considera también un tiempo prudencial y realista para la conclusión de cada etapa de forma satisfactoria.

En esta etapa se prueban diversos sensores y configuraciones, para este proyecto se consideró en primer lugar el uso de sensores infrarrojos sin embargo su aplicación fue descartada al notar en las primeras etapas de pruebas que no respondía bien ante colores cercanos al negro además de que su costo era más elevado y presentaba otros problemas como la reflexión de las señales en superficies reflectantes produciendo errores en cantidades considerables. En la misma línea de análisis se considera el uso de sensores ultrasónicos por su tiempo de vida extenso, bajo costo y alta estabilidad frente a la interrupción del entorno, En un primer momento se considera la configuración del sistema en base a un solo sensor haciendo uso de su capacidad de detección de distancia en centímetros, pero debido a la velocidad de funcionamiento requerido para hacer las lecturas y el ángulo necesario para realizar una lectura precisa es que se descarta esta configuración y se adapta una configuración de doble sensor destinando uno a contar el ingreso y otro a contar el egreso determinando mediante el firmware el orden de activación de cada sensor para de esta manera determinar con exactitud si era contado como una entrada al local o una salida, el proceso de conceptualización del prototipo se muestra en la figura 4 y 5.



miro

**Figura 4.** Desarrollo del prototipo elaborado por el autor.



**Figura 5.** Maqueta del concepto. Fuente: Elaboración propia

### **Desarrollo de Producto a nivel sistema**

Para el desarrollo del sistema se propone 4 niveles de desarrollo, en la ilustración 4 se evidencia el esquema propuesto que se llevara a cabo como guía para el desarrollo de los componentes necesarios, así como el uso de las librerías requeridas por otras partes de la aplicación.

Esbozar las partes principales del sistema también es un buen punto de partida para la aplicación de SCRUM dado que en base a estos datos podemos planear el orden de los sprints y las tareas necesarias para la estimación del tiempo de desarrollo.

### **Desarrollo para el microcontrolador atmega328p**

Esta etapa es determinante para el proyecto por lo que se inició desde aquí para el desarrollo, además del firmware es necesario desarrollar el dispositivo electrónico que comunicará los sensores y el microcontrolador con el driver en Android, en consecuencia, se inició un prototipado partiendo de la electrónica y la estabilidad de los componentes haciendo pruebas de estabilidad y compatibilidad. Acto

seguido se realizaron los diagramas de flujo necesarios para el funcionamiento y el algoritmo que permite operar ambos sensores además de la interfaz de comunicación del dispositivo para establecer las señales enviadas y recibidas por el microcontrolador.

### **Desarrollo para la aplicación de comunicación con android**

Habiendo separado las etapas del desarrollo, el diseño y programación de la aplicación en Android se puede realizar de manera separada con la condición de que se ejecuten algunas pruebas de integración en algún punto de la entrega de la aplicación para comprobar la funcionalidad de ambas partes por separado y en conjunto en diferentes momentos.

La aplicación para Android es un punto crítico en el proyecto, ya que esta aplicación enlaza la comunicación con el dispositivo y la interpretación de los datos y también la comunicación con el servidor para el procesamiento de la información del aforo en tiempo real,

La herramienta utilizada para el desarrollo de la aplicación de Android es IntelliJ-IDEA como se muestra en la ilustración 6 la cual es una IDE para desarrollo generalmente usada en java pero también soporta el desarrollo en Android por medio de la instalación de algunos plugins y el Android SDK, aunque fue considerada en primera instancia la aplicación recomendada por Google para desarrollo en Android nombrada Android Studio; sin embargo esta tuvo que ser reemplazada debido a la velocidad de ejecución de las rutinas de inicialización y cierta inestabilidad probablemente debido a incompatibilidad con el sistema operativo de la máquina de desarrollo.



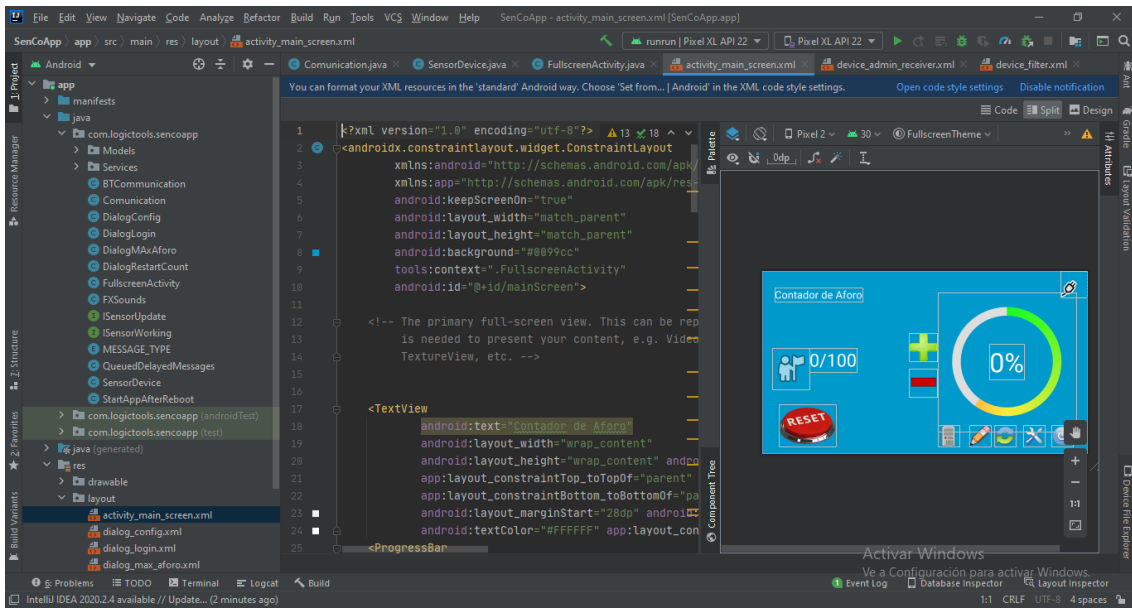
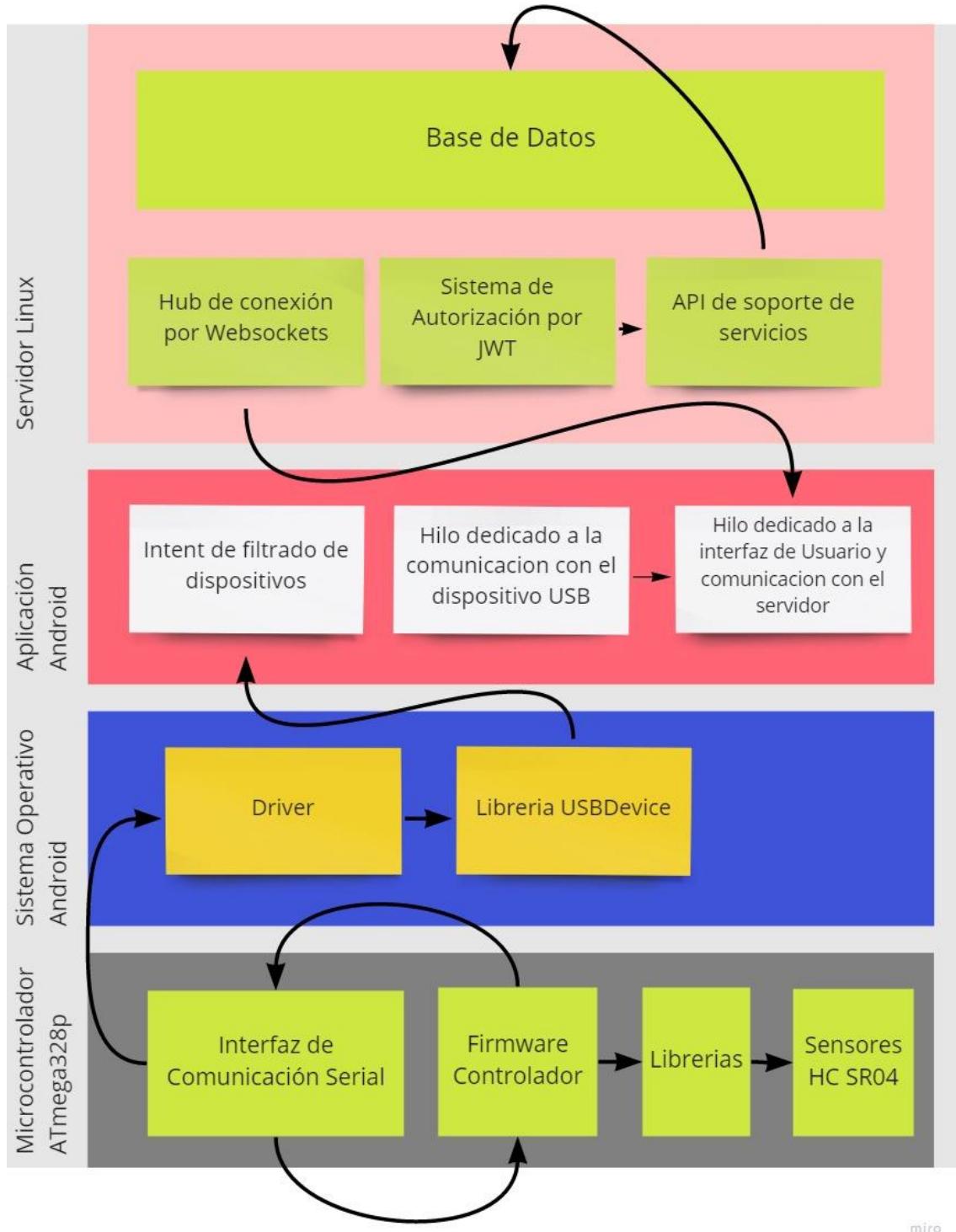


Figura 6. Captura de pantalla de IntelliJ-IDEA fuente: Elaboración propia



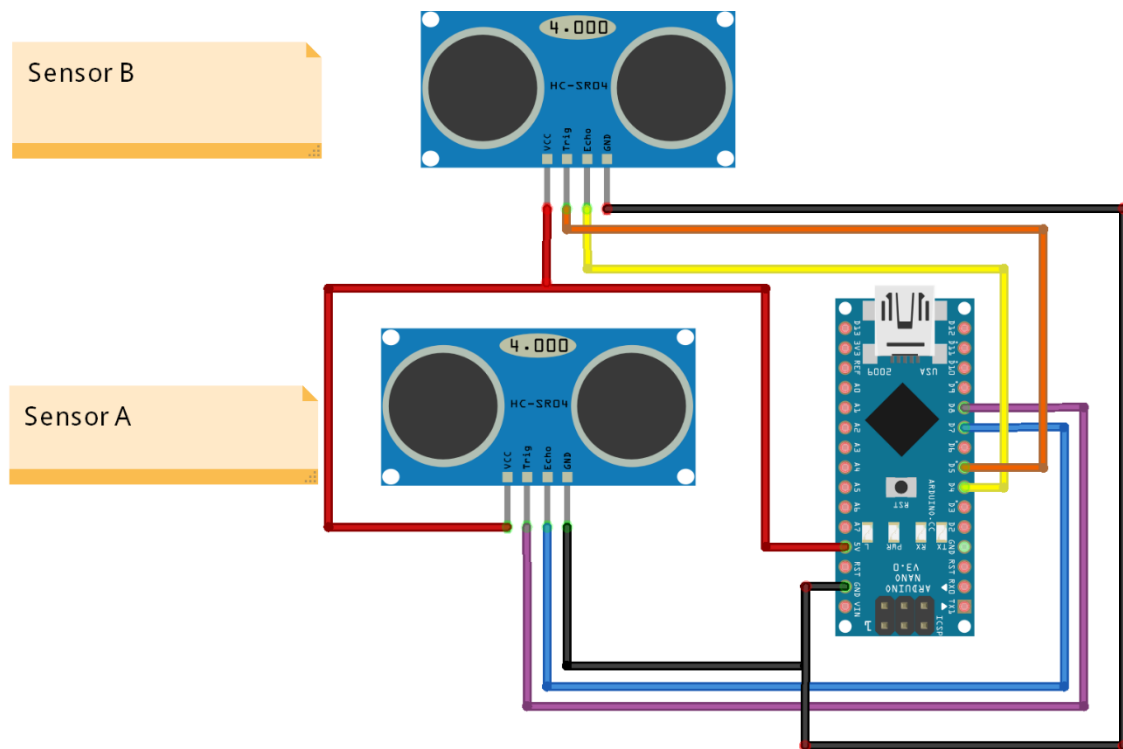
**Figura 7.** Diagrama de afinidad de componentes del sistema, elaborado por el autor.

### Desarrollo a nivel hardware

El desarrollo del hardware necesario para el dispositivo electrónico se inicia con un prototipo en mente y una configuración elegida, en este caso se ha optado por una

configuración de doble sensor para poder realizar de mejor manera el registro de un ingreso o una salida de manera independiente por las razones elegidas páginas arriba.

En la Figura 8 se muestra el diagrama de conexiones utilizado para el hardware, cabe resaltar que el consumo de cada sensor en modo de trabajo es de 15mA según el fabricante por lo tanto para los 2 sensores la alimentación requerida es de 30mA, el Arduino nano rev3 utilizado en el prototipo tiene un soporte para alimentar dispositivos externos de 40mA por cada pin y un máximo de 200mA como máximo con estabilidad, si fuera necesario entregar más energía se necesitaría una alimentación externa; sin embargo este no es el caso.

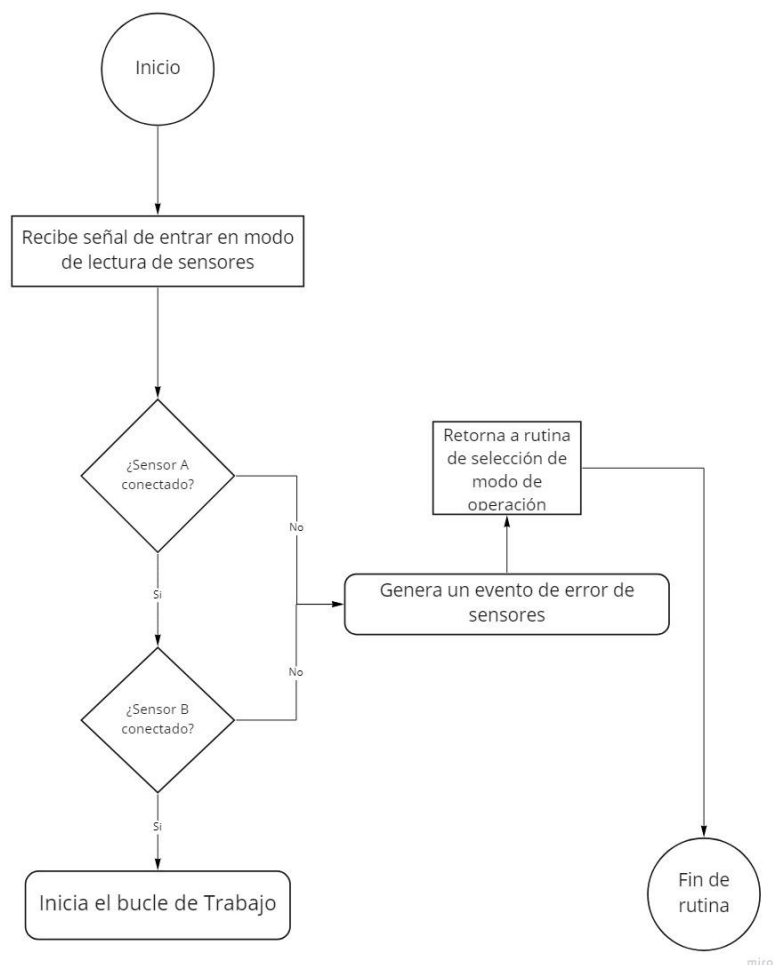


**Figura 8.** Diagrama de conexión de los sensores, elaborado por el autor.

El desarrollo del firmware para el microcontrolador ATmega328p fue dividido en rutinas de funcionamiento la cuales fueron esbozadas en lo siguientes flujogramas en los cuales se trata de explicar el flujo de ejecución del programa que permite gobernar los sensores de manera estable y funcional, más adelante estos

algoritmos fueron refinados y organizados en el código de programación utilizando el lenguaje de c++ y las librerías de Avr y Arduino.

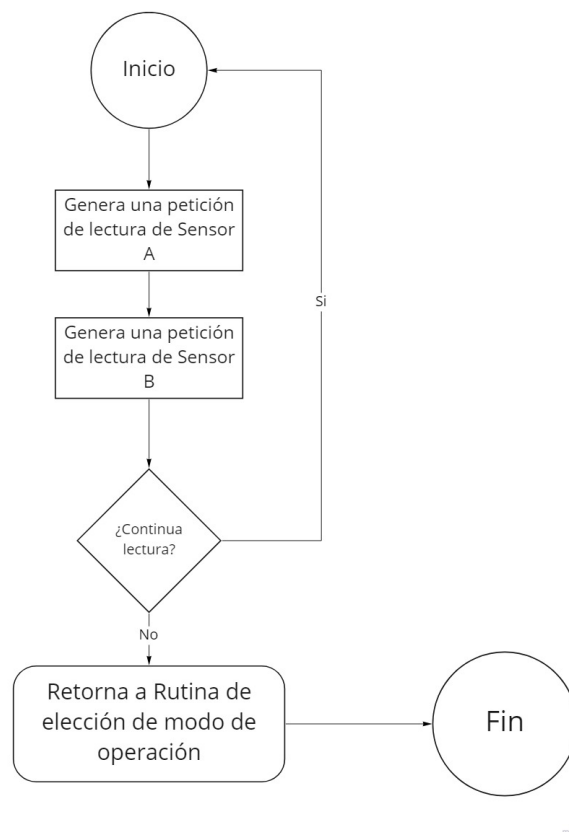
En el flujograma de la figura 9 inicializamos las operaciones de lectura de los sensores con una comprobación antes de entrar en modo de trabajo, esto es necesario debido a que en funcionamiento los sensores entran en un bucle de lectura del que solo se puede salir reiniciando el dispositivo o enviando una señal externa mediante la interfaz serial.



**Figura 9.** Flujograma de rutina de comprobación de sensores, elaborado por el autor

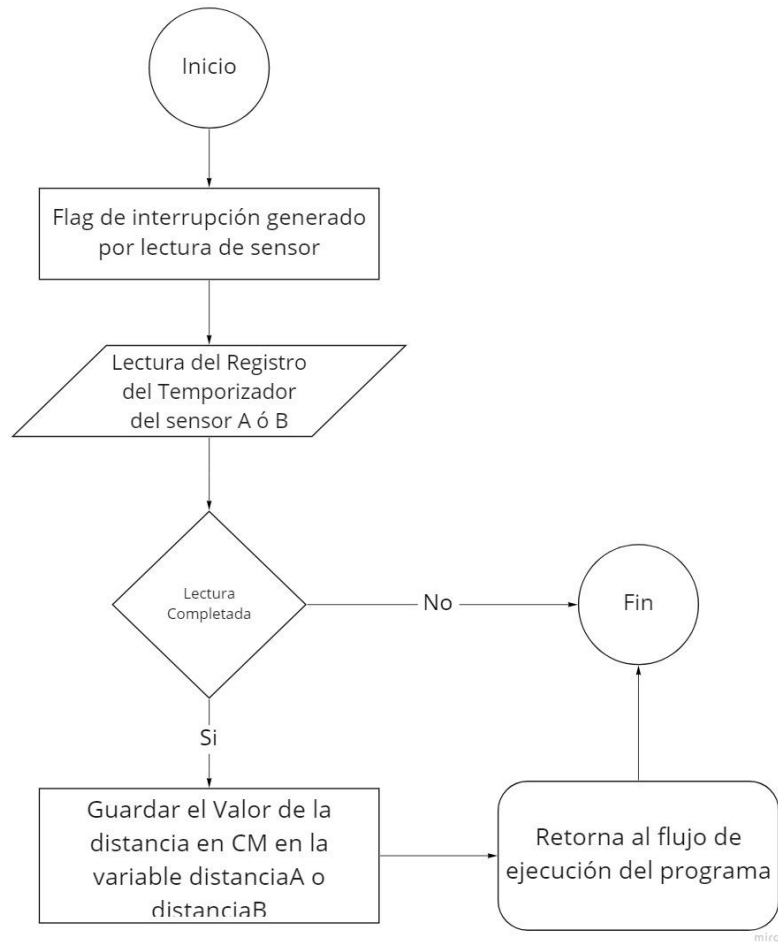
La rutina de operación en el ciclo de funcionamiento del sensor se muestra en el flujograma de la ilustración 10, es un bucle continuo de lectura que depende de una variable condicional, esta variable puede ser cambiada mediante interrupción de software desde otro punto del programa en el bucle principal el cual no se ha

considerado en los flujogramas para evitar confusión en el funcionamiento, para detallar el funcionamiento en esta parte es necesario aclarar que los sensores no pueden ser leídos en cada iteración del bucle debido a las restricciones técnicas de su funcionamiento físico, esto debido a que las ondas ultrasónicas tienen un margen de retorno en el tiempo recomendado por el fabricante de aproximadamente 29ms esta recomendación está basada en la constante de velocidad del sonido manejada a nivel de la física con una estimación de 29 ms para la ida y retorno de una señal, en consecuencia a esta restricción los sensores son medidos utilizando una rutina de interrupción de software la cual avisa al flujo principal del programa que se ha obtenido una lectura o delo contrario si no ha recibido una lectura en una ventana de tiempo determinada se registra la lectura como perdida o absorbida y se informa al flujo principal de que el sensor esta disponible para leer nuevamente.



**Figura 10.** Rutina de operación de flujo de programa al leer los sensores elaborado por el autor.

La ilustración 11 enuncia el funcionamiento de las rutinas de interrupción de software necesarias para la actualización de las variables que guardan los datos de distancia leídos por el microcontrolador, la operación de estas rutinas depende de los temporizadores internos del microcontrolador cuyos registros son desbordados luego de una cantidad de ciclos de ejecución avisándonos de que es momento de leer los registros de estado para la lectura de los sensores, estos temporizadores generan interrupciones al microcontrolador el cual pone en pausa su funcionamiento para saltar a la ejecución de las mencionadas rutinas en el orden que aparezcan, el relevante en este punto comprender el funcionamiento de la memoria y el stack pointer del microcontrolador, dado que la pila en ejecución para el programa principal es almacenada en un registro rápido junto a otros registros y esto puede alterar el funcionamiento y la estabilidad del programa por lo tanto se recomienda que la ejecución de las rutinas de interrupciones sea lo más ligera y sencilla posible para este caso lo único que hace la rutina es comprobar si ya se ha completado la lectura del sensor con una variable booleana y de ser así escribir estos datos en una variable no volátil para su posterior lectura e interpretación en otra parte del flujo del programa principal, una vez terminadas todas las operaciones de las rutinas de interrupción se retorna al flujo principal en donde se había quedado el stack pointer.



**Figura 11.** Rutina de interrupción de lectura de sensores, elaborado por el autor

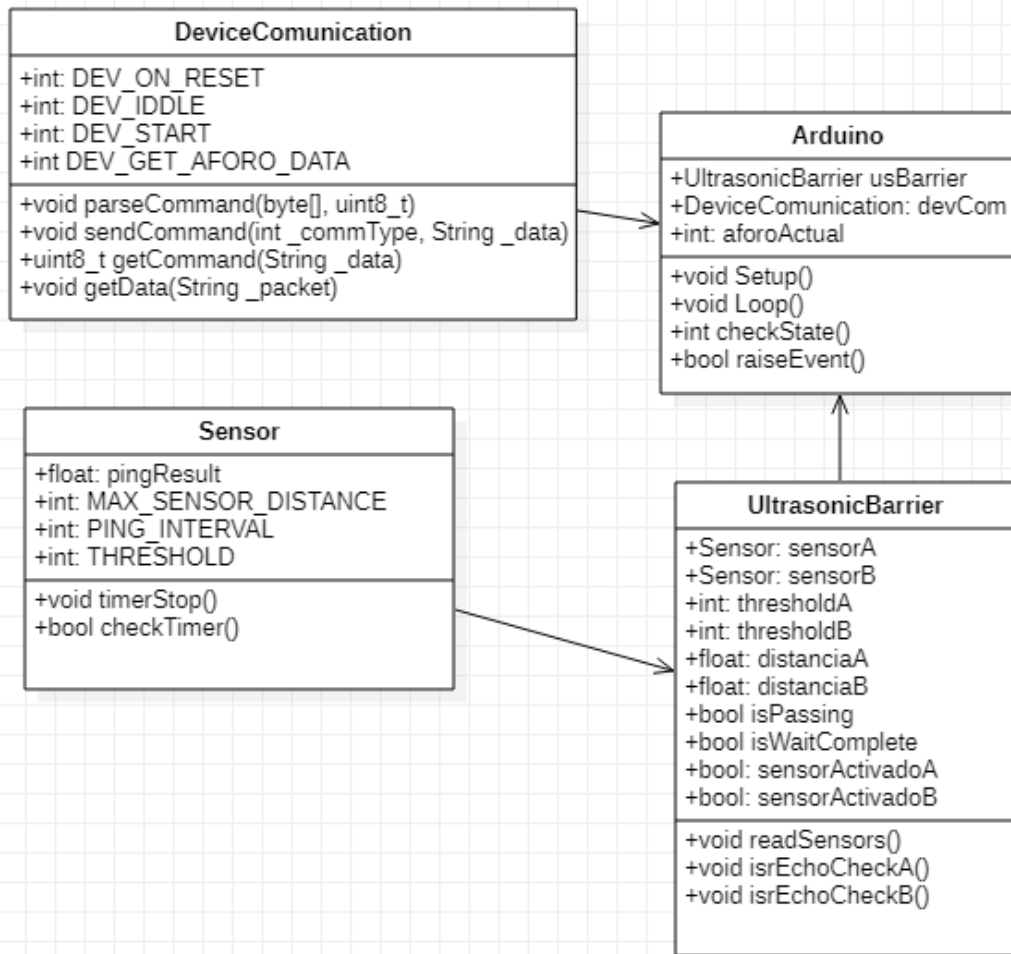
Finalmente el flujograma mostrado en la ilustración 12 describe el funcionamiento operativo de la parte principal del dispositivo, el funcionamiento de la barrera ultrasónica, para conseguirlo se ha hecho uso de múltiples variables de control y buffers de umbral de sensibilidad de respuesta de los sensores basado en múltiples condicionales anidadas, la operatividad básica del algoritmo podemos describirla en 3 estados, el primero corresponde a un estado de reposo en el cual el micro controlador está leyendo continuamente la señal de rebote de la barrera cuando está disponible hasta que hay un cambio suficientemente alto basado en el umbral de sensibilidad para detectar si el sensor A o el sensor B han detectado algo, si se produce un cambio considerable en el umbral se pasa al siguiente estado el cual sería de detección, en dicho estado se han activado las variables que indican si el sensorA o el sensorB han sido activados dentro del umbral de lectura y se pone en





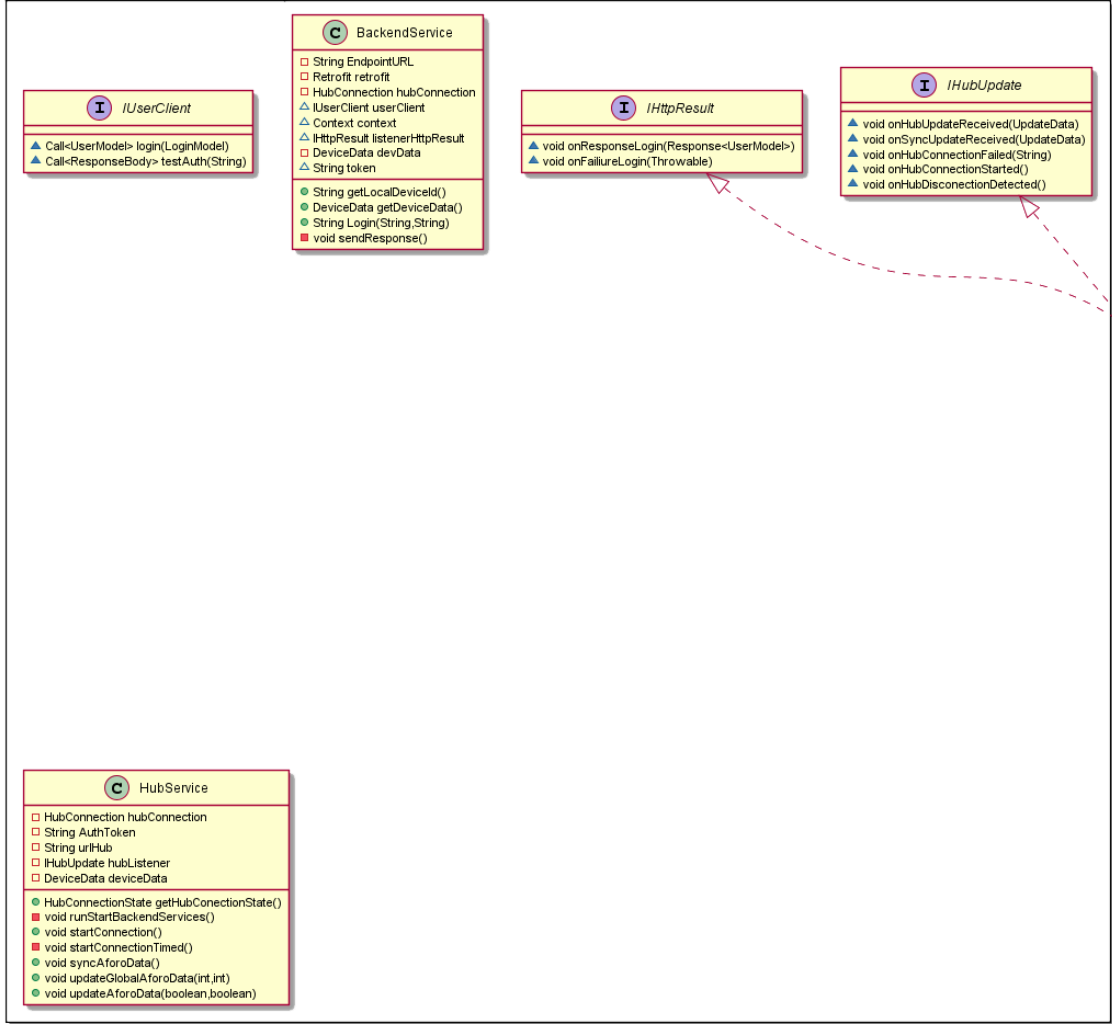
## Desarrollo a nivel software

Diagrama de clases Dispositivo barrera ultrasónica para el MICROCONTROLADOR

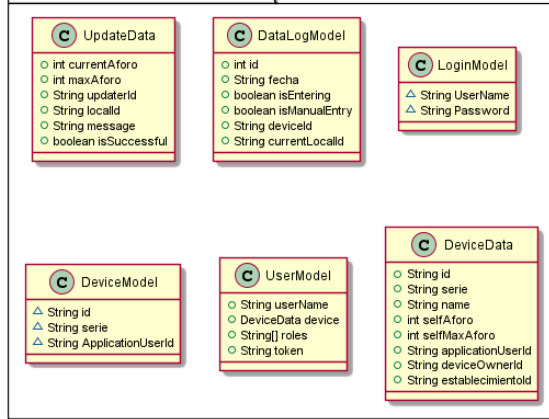


**Figura 13.** Diagrama de clases para implementación de barrera ultrasónica. Fuente: Elaboración propia

**com.logictools.sencoapp.Services**



**com.logictools.sencoapp.Models**





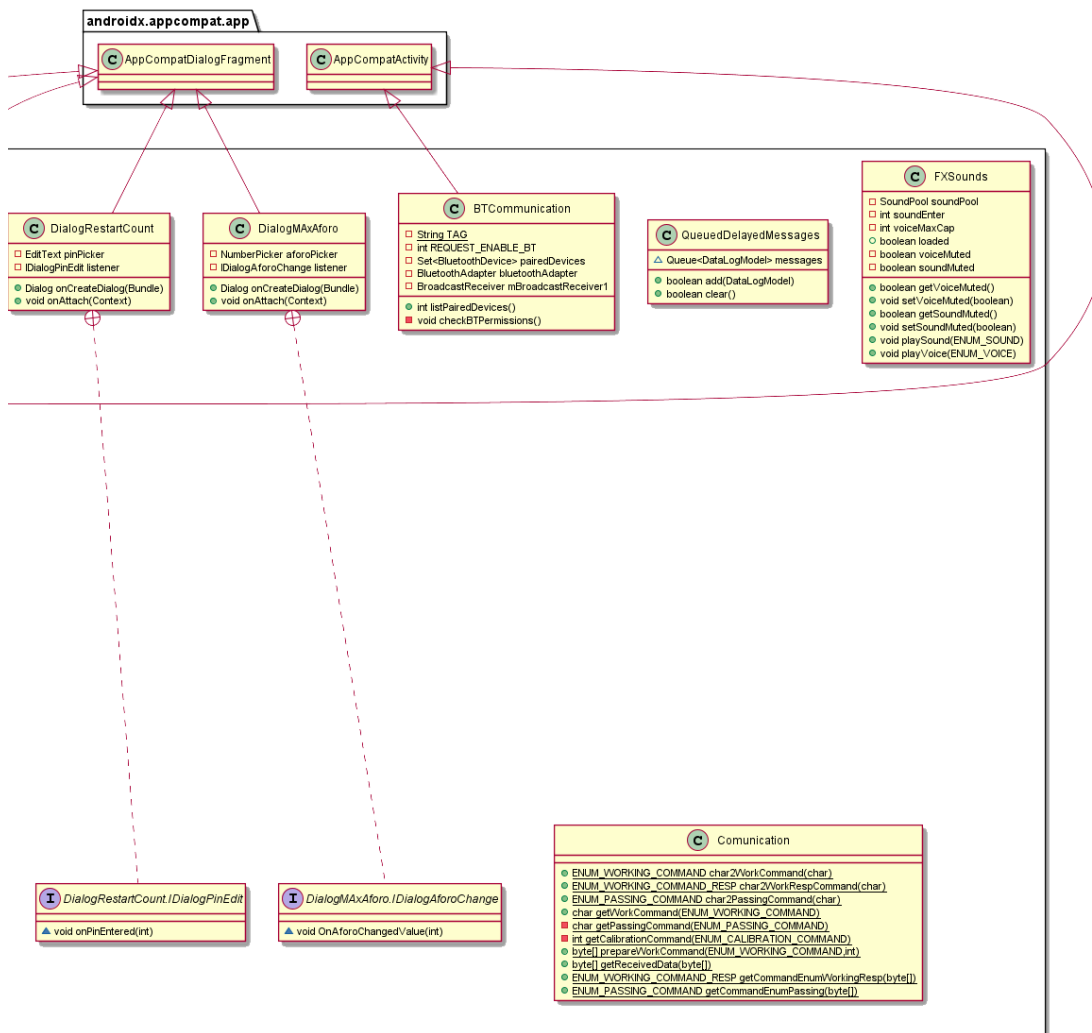


Figura 14. Diagrama de clases despliegue en Android elaborado por el autor

### Diagrama de clases para el desarrollo de la API backend del servicio

La arquitectura aplicada en el desarrollo de la API de soporte de servicios está basada en una disgregación por capas de las responsabilidades y reglas del negocio y otra capa encargada de las transacciones con la base de datos otra capa dedicada a la exposición de los servicios a través de los controles

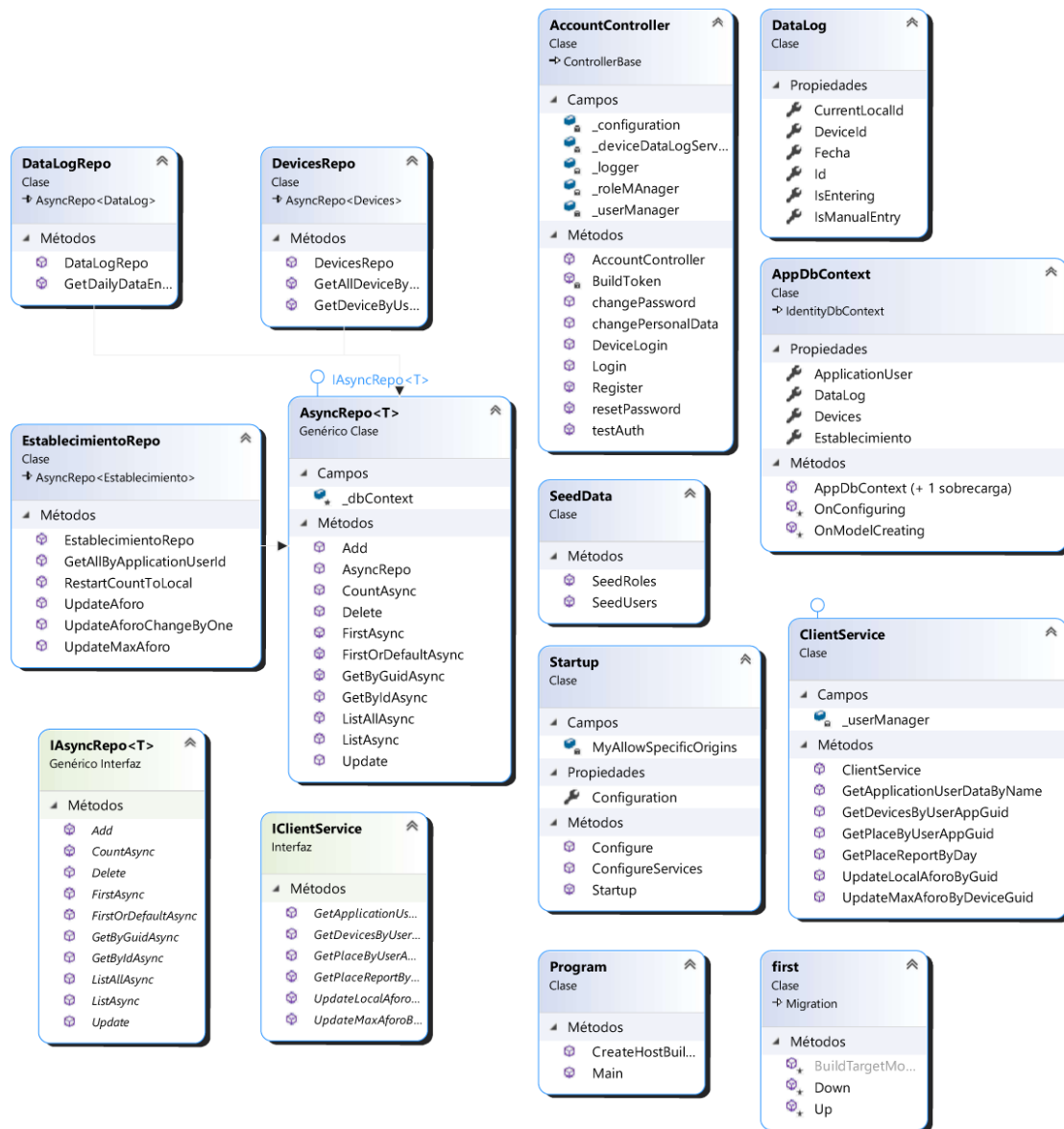


Figura 15. Diagrama de clases para la API parte 1



Figura 16. Diagrama de clases para la API parte 2

## Mockups para el desarrollo de la interfaz gráfica en la aplicación Android

Para la visualización del sistema en una app un método de común uso en etapas tempranas del desarrollo es la creación de mockups para guiar el posicionamiento de los componentes en la interfaz gráfica y tener una vista previa de como lucirá el sistema terminado.

En la ilustración 17 se muestra el mockup que contiene los elementos principales de la interfaz gráfica, este es el componente central de la interfaz gráfica y desde aquí se puede controlar todos los aspectos a los que el usuario puede acceder para controlar los datos y el funcionamiento del dispositivo. El elemento más importante de esta vista es la barra de progreso circular y la información en porcentaje y números enteros de la situación actual de aforo en el sistema.



**Figura 17.** Mockup Pantalla principal. Fuente: elaboración propia

La ilustración 18 muestra el cuadro de dialogo que permite configurar en modo encendido o apagado de las tres configuraciones básicas de preferencias del usuario. Estos datos son guardados y cargados en cada arranque del sistema y facilitan la adaptación del algún elemento del sistema al usuario.



**Figura 18.** Mockup Configuraciones básicas de preferencias. Fuente: Elaboración propia

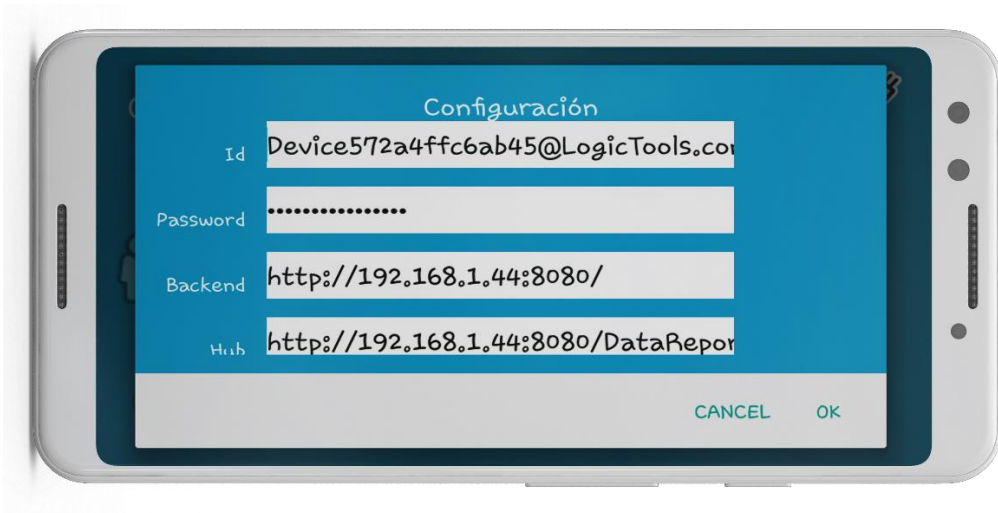
El control del cálculo del aforo máximo en números enteros se muestra en la ilustración 19 y hace uso de un selector deslizable hacia arriba o hacia abajo para la configuración de la capacidad máxima de personas que admite el espacio que será gestionado.



**Figura 19.** Mockup selección de aforo máximo. Fuente Elaboración Propia

Para el funcionamiento del sistema en modo enjambre necesitamos configurar algunos datos para que la aplicación pueda autenticar directamente a los servicios en la nube y obtenga desde la misma la configuración en su modo de operación para ese efecto en la ilustración 20 se muestra los elementos que necesitará el cuadro de dialogo.





**Figura 20.** Mockup configuración de los credenciales y rutas a los servicios. Fuente: Elaboración propia

El mockup presentado en la ilustración 21 muestra el cuadro de dialogo de confirmación de cierre de la app.

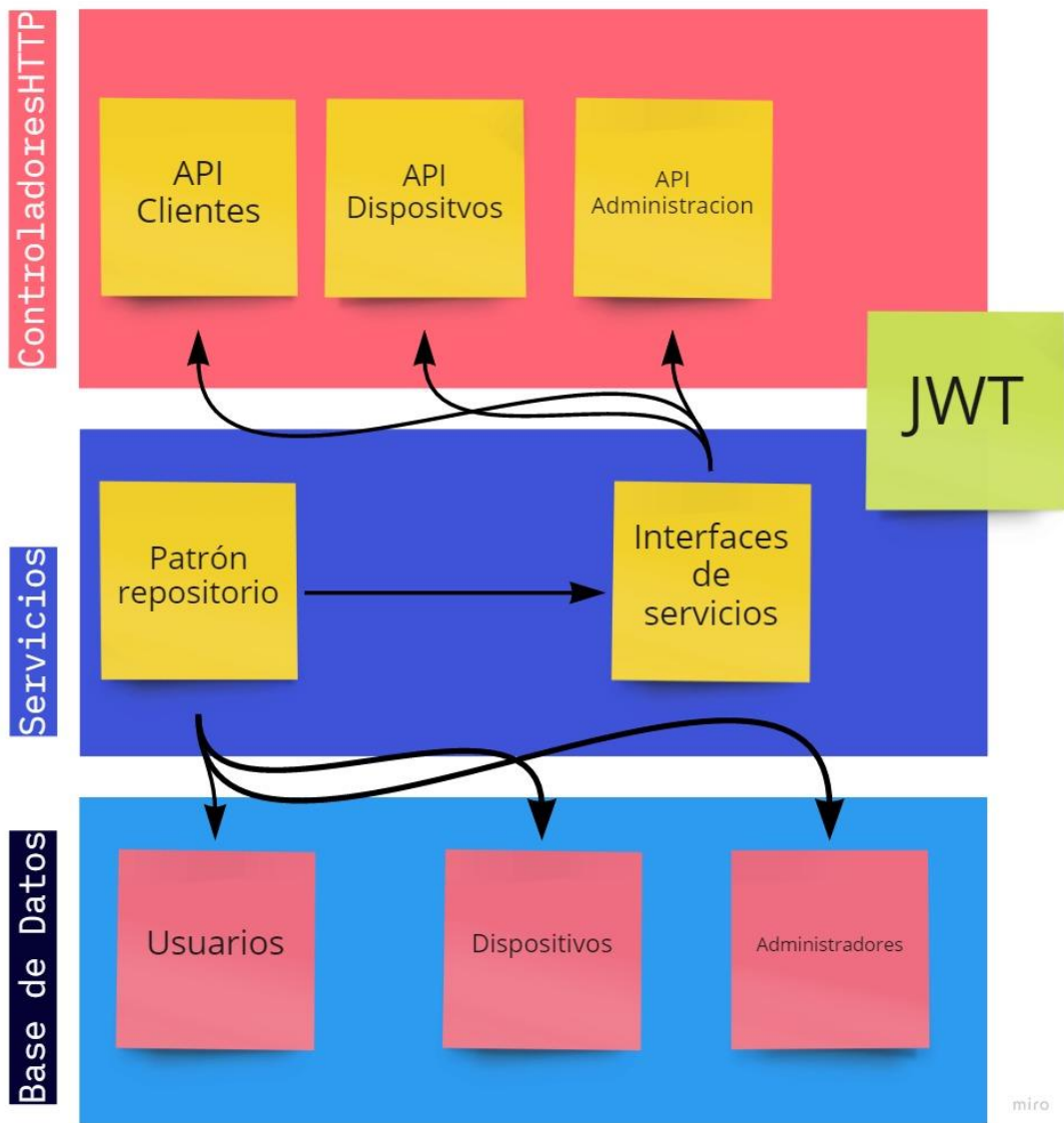


**Figura 21.** Mockup Confirmación de cierre de app. Fuente: Elaboración propia

## Desarrollo la API rest

El desarrollo del servicio backend fue ejecutado utilizando tecnología de Microsoft .Netcore 3.1 la cual puede ser desplegada en Linux sin mayores inconvenientes, la comunidad recomienda el uso del kestrel de Linux implementado detrás de una configuración de proxy inverso para incrementar la estabilidad y sobre todo la seguridad de la aplicación. Para el caso de la API rest, se procedió en la programación del sistema utilizando la IDE de Microsoft visual studio 2019 community edition, a continuación, se muestran los endpoints expuestos por la API.

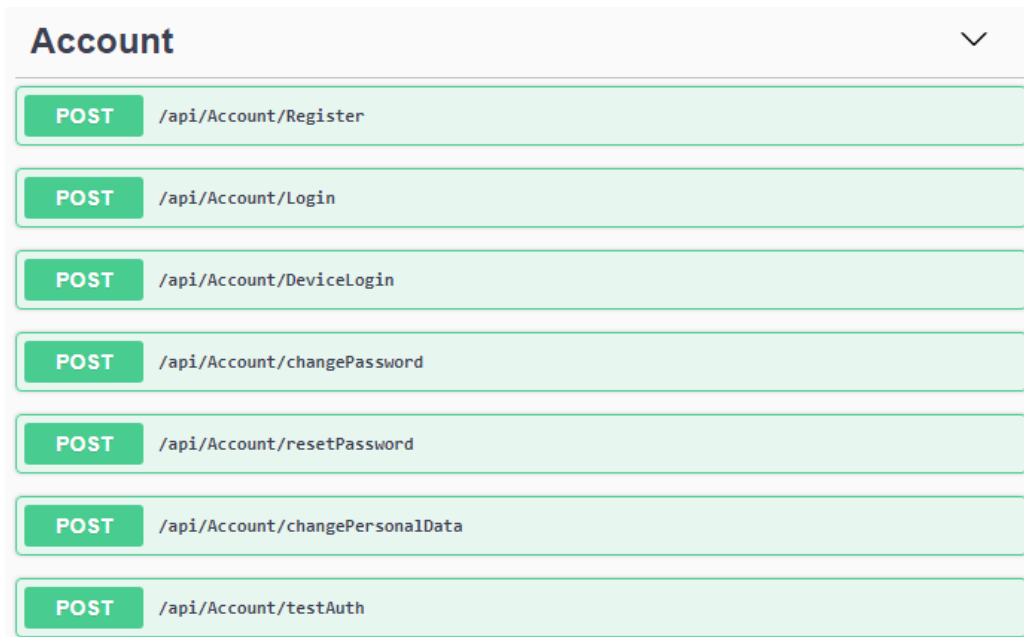
La arquitectura propuesta se muestra en la ilustración 22 se desarrolla en base a 3 capas claramente enunciadas y con responsabilidades directas se hace uso del patrón repositorio y el patrón Servicio para la implementación de las clases, el uso del middle-ware para la autenticación al sistema en base a JWT para cada parte protegida del sistema y también para el acceso a los dispositivos, además del patrón inyección de dependencia utilizado para mejorar la administración y escalabilidad del sistema propuesto en su parte backend.



**Figura 22.** Arquitectura de la API REST de servicios; fuente: elaboración propia

En las ilustraciones 23, 24 y 25 se muestran algunos end-points expuestos por la API, por seguridad y complejidad de los datos expuestos se ha elegido el método

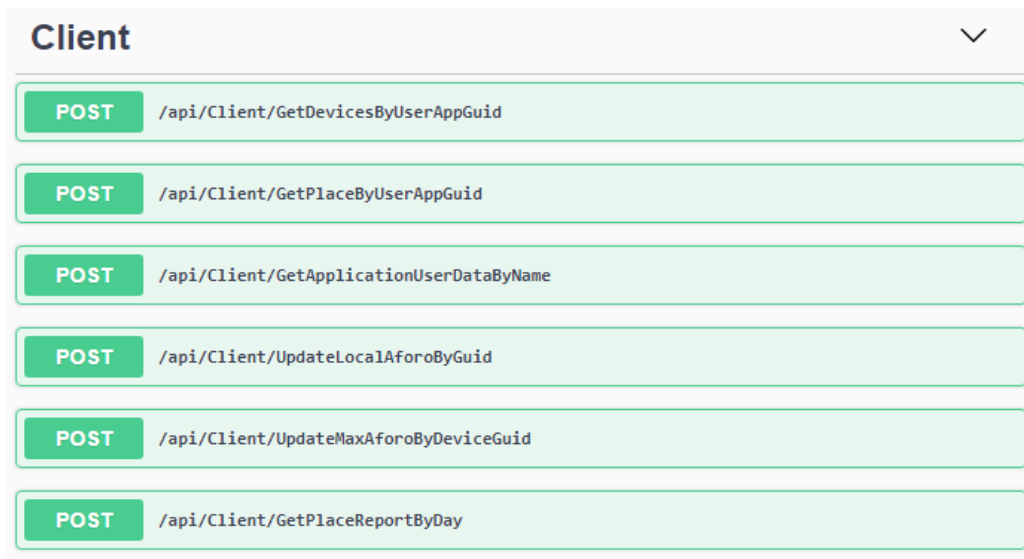
post para la transferencia y recepción de peticiones a estos end-points los cuales devuelven y reciben respuestas y peticiones en JSON



The screenshot shows a list of seven POST endpoints for the Account API. Each endpoint is displayed in a light green box with a dark green 'POST' label on the left and the endpoint path on the right. The endpoints are: /api/Account/Register, /api/Account/Login, /api/Account/DeviceLogin, /api/Account/changePassword, /api/Account/resetPassword, /api/Account/changePersonalData, and /api/Account/testAuth. The title 'Account' is at the top left, and a dropdown arrow is at the top right.

Method	Endpoint
POST	/api/Account/Register
POST	/api/Account/Login
POST	/api/Account/DeviceLogin
POST	/api/Account/changePassword
POST	/api/Account/resetPassword
POST	/api/Account/changePersonalData
POST	/api/Account/testAuth

**Figura 23.** End-points HTTP del api de cuentas.



The screenshot shows a list of six POST endpoints for the Client API. Each endpoint is displayed in a light green box with a dark green 'POST' label on the left and the endpoint path on the right. The endpoints are: /api/Client/GetDevicesByUserAppGuid, /api/Client/GetPlaceByUserAppGuid, /api/Client/GetApplicationUserDataByName, /api/Client/UpdateLocalAforoByGuid, /api/Client/UpdateMaxAforoByDeviceGuid, and /api/Client/GetPlaceReportByDay. The title 'Client' is at the top left, and a dropdown arrow is at the top right.

Method	Endpoint
POST	/api/Client/GetDevicesByUserAppGuid
POST	/api/Client/GetPlaceByUserAppGuid
POST	/api/Client/GetApplicationUserDataByName
POST	/api/Client/UpdateLocalAforoByGuid
POST	/api/Client/UpdateMaxAforoByDeviceGuid
POST	/api/Client/GetPlaceReportByDay

**Figura 24.** End-points del API de clientes.

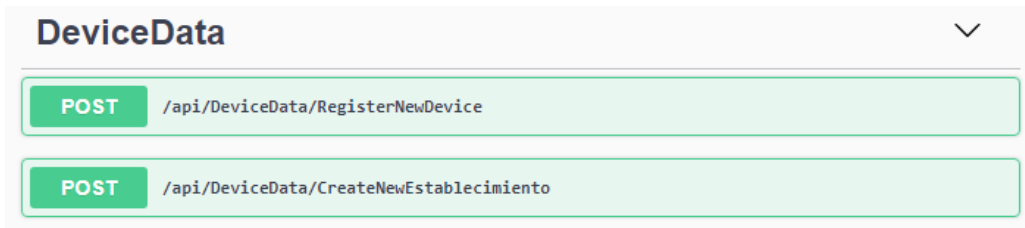


Figura 25. End-points para los datos de los dispositivos.

## Fragmento del Manual de Usuario

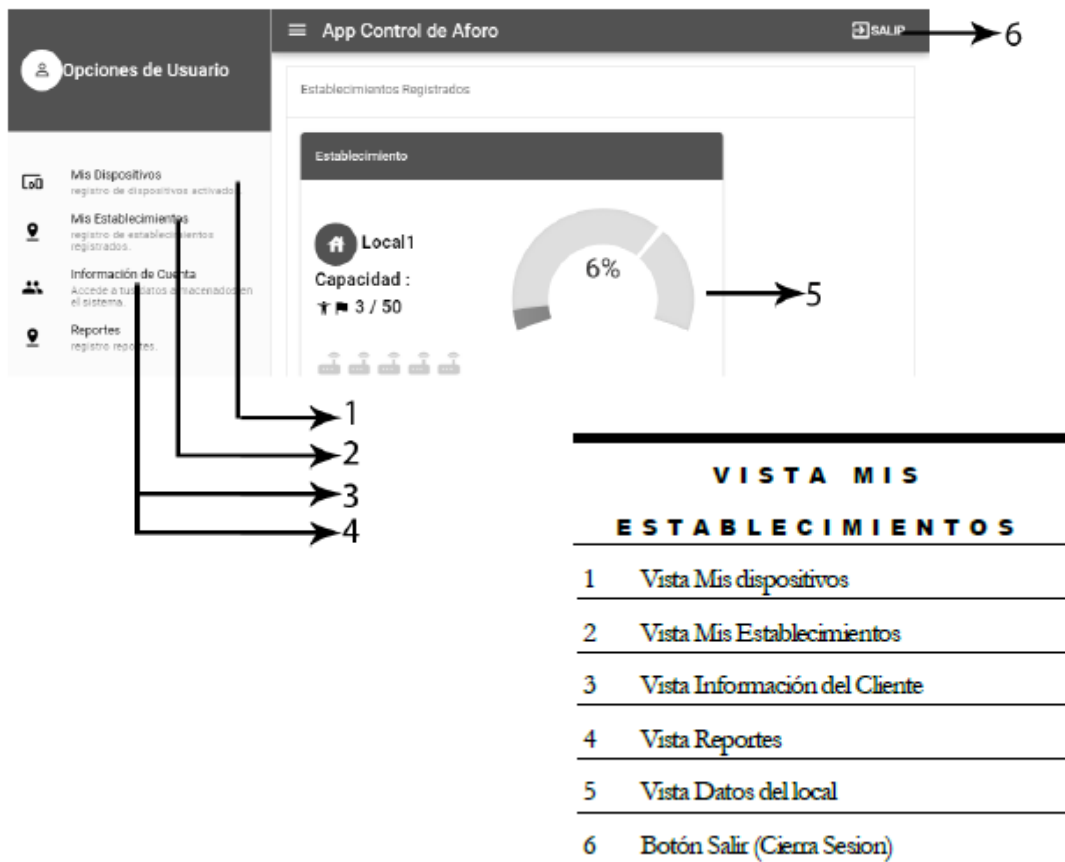
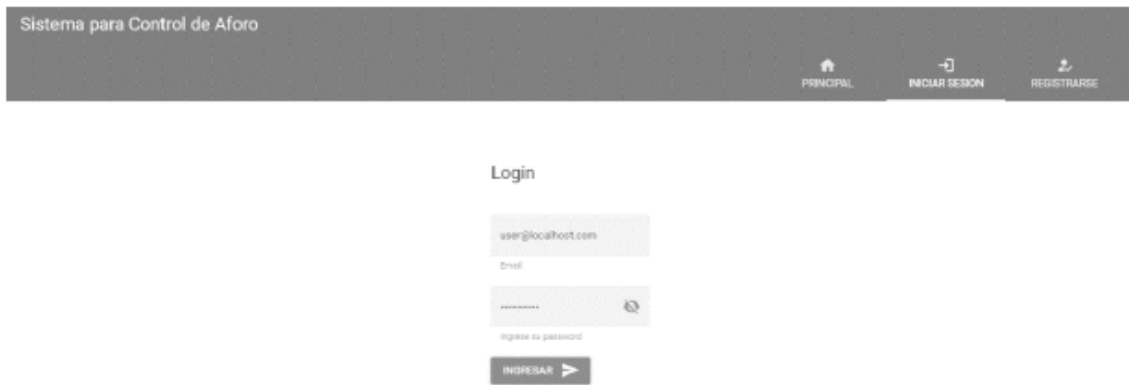


Figura 26. Vista Web



**Figura 27.** Vista Web Login



**Figura 28.** Vista Web Landing Page



**UNIVERSIDAD CÉSAR VALLEJO**

**FACULTAD DE INGENIERÍA Y ARQUITECTURA  
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS**

### **Declaratoria de Autenticidad del Asesor**

Yo, AGREDA GAMBOA EVERSON DAVID, docente de la FACULTAD DE INGENIERÍA Y ARQUITECTURA de la escuela profesional de INGENIERÍA DE SISTEMAS de la UNIVERSIDAD CÉSAR VALLEJO SAC - LIMA ATE, asesor de Tesis titulada: "Sistema inteligente para el Control de aforo en una Institución de Salud Pública, Puno 2023", cuyo autor es RUELAS PAREDES RUSBEL OLIVER, constato que la investigación tiene un índice de similitud de 13.00%, verificable en el reporte de originalidad del programa Turnitin, el cual ha sido realizado sin filtros, ni exclusiones.

He revisado dicho reporte y concluyo que cada una de las coincidencias detectadas no constituyen plagio. A mi leal saber y entender la Tesis cumple con todas las normas para el uso de citas y referencias establecidas por la Universidad César Vallejo.

En tal sentido, asumo la responsabilidad que corresponda ante cualquier falsedad, ocultamiento u omisión tanto de los documentos como de información aportada, por lo cual me someto a lo dispuesto en las normas académicas vigentes de la Universidad César Vallejo.

LIMA, 22 de Febrero del 2023

<b>Apellidos y Nombres del Asesor:</b>	<b>Firma</b>
AGREDA GAMBOA EVERSON DAVID <b>DNI:</b> 18161457 <b>ORCID:</b> 0000-0003-1252-9692	Firmado electrónicamente por: AGREDA el 27-03- 2023 08:58:24

Código documento Trilce: TRI - 0534648