

SEGURIDAD DE LA VIVIENDA

COMPUTACIÓN UBICUA



GRUPO 4: LAB 12:00-14:00

Alberto Estévez Casarrubios
Álvaro Barragán Codesal
Eduardo Garzo Jiménez
Nicolae Alexandru Molnar

Índice:

Introducción	2
Análisis del problema	3
Objetivos y alcance del proyecto seleccionado	4
Mirilla inteligente:	4
Sensores de intrusión:	5
Conclusión:	5
Descripción de las ideas descartadas	6
Cierre de seguridad:	6
Medidas disuasorias:	7
Conclusión:	9
Tecnología a utilizar	9
Arquitectura a utilizar	9
Capa de percepción:	10
Capa de transporte:	10
Capa de procesamiento de datos:	11
Capa de aplicación:	11
Problema:	12
Arquitectura final:	12
Listado de Sensores utilizados	13
Sensor de proximidad:	13
Cámara corriente:	14
Sensor de movimiento:	14
Listado de Actuadores	15
LED	15
Foco más potente.	16
Microcontroladores a utilizar	17
Boceto del diseño del circuito con fritzing	18
Lenguajes de programación a utilizar	19
Software a utilizar	20
IDE de Arduino:	20
Entorno de PostgreSQL:	21
IDE de Android Studio:	21
Servidor web de Apache:	21
Características de la app móvil	22
Actividad de login:	22
Vista previa:	23
Actividad principal:	23
Vista previa:	24
Actividad de configuración:	25
Vista previa:	25
Características de la página web	26

Página principal:	26
Vista previa:	27
Log In:	27
Vista previa:	27
Vista previa:	28
Configuración:	28
Vista previa:	28
Biblioteca y Observación:	28
Planificación temporal, plan de desarrollo	29
Resumen y conclusiones	30
Bibliografía	31

Introducción

La seguridad en la vivienda es un asunto fundamental que nos preocupa a todos por igual, ya que deseamos un lugar en el que sentirnos seguros, con nuestros conocidos, así como poder dejar nuestros bienes en un sitio donde el riesgo de ser robados o dañados sea mínimo.

Siempre ha habido criminales que asaltan nuestros hogares, sin embargo, nunca hemos tenido una tecnología que nos pueda informar sobre los robos e incursiones de una forma tan eficiente como actualmente. Es por ello que debemos conocer las desventajas y riesgos que suponen no tener un sistema de seguridad, para así ser conscientes de todas las ventajas que tienen.

Entre los riesgos principales de no tener un sistema de seguridad encontramos:

- Que nuestra vivienda está más desprotegida y es más vulnerable ante los robos y asaltos.
- En las viviendas deshabitadas (ejemplo: casas de vacaciones las cuales están vacías la mayor parte del tiempo) no nos enteramos de si se ha producido un asalto o incluso una okupación hasta que ya es demasiado tarde. (No es lo mismo enterarte al instante de esa información, que 5 meses después).
- Sin sistemas de alarmas oficiales con captura de imagen no podemos demostrar a las autoridades competentes el hecho de que se haya producido una intrusión y así denunciar.

El documento presentado a continuación presenta la idea o prototipo de un sistema de seguridad inteligente, la mirilla inteligente.

El sistema tiene dos componentes principales, una mirilla integrada en la puerta principal, capaz de reconocer rostros y manos para detectar la presencia de una persona frente a la puerta, y un sistema de intrusión instalado en cualquier otra entrada o punto vulnerable distintos a la puerta principal.

Conforme avance la lectura del documento irá recibiendo y comprendiendo los detalles de la implementación del prototipo, los objetivos que se esperan cumplir y la metodología y tecnología utilizadas para ello.

Análisis del problema

Por todo lo que acabamos de mencionar es muy importante plantearse el hecho de incorporar un sistema de seguridad a nuestro hogar que nos asegure que nadie puede acceder a nuestra vivienda sin nuestra autorización para evitar cualquier daño o hurto.

El hecho de tener un buen sistema de seguridad en nuestra vivienda no solo actúa como un escudo protector tanto dentro como fuera de esta, sino también nos permite controlar su actividad en cualquier momento en tiempo real y desde cualquier lugar.

Podemos programar un modelo de inteligencia artificial preparado para mostrar toda la información necesaria en el sistema. Para ello vamos a utilizar una librería llamada numpy, con la que podemos realizar el cómputo de información de forma sencilla.

Este cómputo de información no será mínimo puesto que nuestro sistema estará encargado de proporcionar los servicios completos al cliente. Un nivel de atención superior al especificado por nuestros superiores. Esto nos proporciona cierto nivel de ejemplo. Además, un sistema de seguridad puede ofrecer un método disuasorio frente a posibles atacantes ya no solo por tener un sistema eficaz, sino por el simple hecho de mostrar que lo tienes.

Por esto es importante un sistema de seguridad en nuestra vivienda, porque no solo da seguridad a tu hogar, sino que tú también vives con esa seguridad y sensación de despreocupación al saber que tu casa está protegida todo el día.

Ante este problema nosotros hemos propuesto distintas soluciones de posibles sistemas de seguridad para tu vivienda como:

- Mirillas inteligentes que pueden “inteligentemente” detectar quien se acerca a tu puerta mediante un reconocimiento facial.
- Sensores de intrusión que permiten captar toda actividad interior o exterior de tu hogar con sensores de movimiento, sonido o cercanía de forma que puedas ser avisado tanto tú como la policía si ocurre cualquier actuación sospechosa.
- Por otro lado tenemos medidas disuasorias como focos de luz para cegar al atacante y poder evitar un posible robo.
- Incluso planteamos un cierre total de puertas, cajones o armarios de forma que estos queden bloqueados y no se puedan abrir en el caso de detectar alguna actividad sospechosa, con el fin de que el atacante, aunque consiga entrar en nuestra vivienda, no pueda llevarse nuestras pertenencias más valiosas.

Objetivos y alcance del proyecto seleccionado

Como objetivo principal de este proyecto vamos a diseñar y desarrollar un sistema de seguridad para viviendas que consta de una mirilla inteligente y conjunto de sensores para posibles intrusiones en un perímetro determinado.

Mirilla inteligente:

Esta idea tiene como objetivo detectar situaciones de intrusión que ocurran por la puerta principal, de manera que cualquier persona que se acerque a la misma será grabada, de forma que los propietarios del sistema reciban una notificación y la imagen en tiempo real de lo que ocurra en la puerta.

Para conseguirlo las soluciones que se nos han ocurrido implican los sensores de intrusión que pertenecen a nuestra segunda idea y además:

- Cámara con detección de rostros: de manera que dotemos a nuestro sistema de la capacidad de diferenciar rostros dentro de una imagen, y que envíe la misma al usuario, cuando detecte el rostro.
- Cámara corriente: que capture fotos, videos o imágenes en vivo y que los retransmite al usuario cuando detecte a personas o se le solicite.
- Sensor de proximidad: Un sensor capaz de detectar las situaciones en las que un objeto se encuentra lo suficientemente cerca de la puerta principal.
- Sensor de iluminación: Capaz de medir el nivel de luz que hay en la zona de la lente.
- Actuador de iluminación: Una fuente de luz que se encienda cuando el sistema detecte un nivel de iluminación bajo.

Esta propuesta es un sistema inteligente ya que el sistema debe analizar las imágenes que captura para saber si una persona (un rostro) está frente a la puerta. Para lograr este objetivo utilizamos un modelo de IA capaz de reconocer rostros y manos.

La mirilla inteligente es una propuesta que involucra los elementos anteriormente mencionados, así como la necesidad de transferir datos por una capa de transporte, procesar las imágenes capturadas en una capa de procesamiento y notificar los eventos sospechosos al cliente en la capa de aplicación.

Además, el sistema proporcionará un servicio de transmisión de imagen en vivo por el que la mirilla puede actuar como cámara web mostrando al cliente la actividad frente a su puerta.

Por tanto, desde el punto de vista del proyecto, creemos que la idea es suficiente para los requisitos del mismo. Sin embargo, si pensamos en la seguridad de una vivienda, las intrusiones no siempre se realizan por la puerta y por ello hemos decidido complementar la idea con la implementación de sensores de intrusión.

Sensores de intrusión:

El objetivo de los sensores es poder detectar cualquier tipo de actividad que se pueda producir alrededor de la vivienda, ya sea por movimientos cerca de la casa hasta posibles roturas de ventanas o entradas forzosas.

Para conseguir detectar estos movimientos y que se nos informe de los mismos hemos pensado en:

- Sensores de movimiento: sensores infrarrojos que detectan obstáculos, activándose cuando algo pasa por delante del láser, pudiendo instalarse en cualquier parte de la casa.
- Sensores de cercanía: que emiten ondas ultrasónicas en una dirección, de manera que si estas ondas vuelven tras encontrarse con algún obstáculo, sabiendo incluso la distancia a la que está el obstáculo del sensor. Siendo útil para colocarlo en las puertas de entrada para controlar las mismas.
- Sensores de sonido: que tengan prefijado un umbral de decibelios a detectar, de forma que en el momento en el que se supere la intensidad predeterminada se producirá una señal, siendo útiles para detectar si se rompe alguna ventana.

Además de estos sensores habría un sistema de notificaciones que nos comuniquen la información recogida por los mismos, a través de una aplicación móvil que permita al propietario recibir alertas.

Esta era la idea principal de la implementación de los sensores de intrusión. Tiene como ventajas su simpleza en cuanto a implementación, su bajo coste y su alta escalabilidad ya que para incrementar la magnitud del sistema basta con replicar el módulo y conectarlo al sistema.

La principal desventaja recae a la hora de dotar al sistema de inteligencia, para hacerlo podríamos generar un modelo de Inteligencia Artificial que sea capaz de adaptarse a los horarios del cliente. Esta idea de inteligencia permitiría activar y desactivar los sensores de intrusión en función de si el sistema predice que el usuario estará en casa o no.

El principal inconveniente de esta idea de inteligencia es que las personas somos distintas y existirá un periodo de tiempo en el que el sistema se equivoque porque aún se está adaptando al cliente. Esto es incómodo y poco profesional por lo que descartamos esta idea de inteligencia.

Conclusión:

Una vez explicadas las ideas de proyecto por separado y observado que el sistema de intrusión no tiene inteligencia por sí mismo, hemos decidido que la mejor solución inteligente al problema de la seguridad en la vivienda es combinar la mirilla inteligente con el sistema de intrusión.

De esta manera, tendremos un control inteligente de la entrada de la vivienda sin perder el control sobre los puntos más vulnerables de nuestro hogar. Además conseguimos una propuesta escalable puesto que podemos aumentar el tamaño del sistema añadiendo mirillas a más puertas o más sensores de intrusión, ya que ambos son modulares e independientes.

Para concluir, la idea del proyecto es una mirilla con detección de intrusión cuya inteligencia se basa en el reconocimiento facial de la mirilla.

Descripción de las ideas descartadas

De todas las ideas presentadas en este proyecto, hemos decidido descartar las siguientes:

Cierre de seguridad:

Una idea cuyo objetivo es cerrar y bloquear puertas y/o cajones para, una vez realizada la intrusión, intentar impedir el acceso del intruso a otras salas de la casa y bloquear la apertura de ciertos muebles para impedir el robo de objetos de su interior.

Para lograr este objetivo se han propuesto varias soluciones.

Con objeto de impedir el acceso del intruso a otras salas se ha propuesto un sistema que, por medio de un motor instalado en cada puerta, cierre las puertas y aplique un “pestillo” que imposibilite su apertura. Esta idea quedó descartada debido a que la capacidad del motor de cerrar la puerta depende directamente del tipo de puerta, y un modelo de cierre no podría ser aplicado a cualquier puerta. Además, necesitaríamos un motor capaz de cerrar la puerta incluso si el intruso estuviese tratando de impedirlo, es decir, un motor que ejerciese más fuerza que un humano. Estos motores existen pero su integración en las puertas de un hogar es muy complicada y costosa.

La siguiente idea planteada para cumplir el objetivo consiste en diseñar un mecanismo de “cierre de cajones”, este mecanismo consistirá de un motor capaz de mover un pestillo que impida o permita la apertura de un cajón que puede contener objetos de valor. Haciendo uso solamente de un motor, un pestillo y un hueco en el que encajar el pestillo, al intruso le resultaría mucho más complicado abrir el cajón de valor, puesto que para hacerlo deberá romper el objeto usado como pestillo.

La idea del cierre de cajones nos parece una buena solución para el problema a la que podemos añadir el uso de una aplicación o página web para controlar manualmente el cierre. Esta implementación implicaría un módulo WiFi con el que la placa Arduino pueda comunicarse con el servidor encargado de gestionar la web o con el dispositivo Android del usuario.

Los inconvenientes de esta idea son: El sistema por sí solo no puede detectar la intrusión y aplicar el cierre automáticamente, para ello debería ser otro sistema el que le transmita esa información o el propio usuario el que accione el cierre de forma manual. Además, este sistema carece de inteligencia ya que el cierre es un actuador que sólo sabe abrirse o cerrarse en función de las órdenes del usuario, no analiza ningún tipo de información y actúa en consecuencia.

Para solventar esta carencia de inteligencia se ha propuesto utilizar un sistema de detección de proximidad, como podría ser NFC o RTID. Un sistema capaz de abrir y cerrar los cajones en función de la proximidad del cliente. También se ha propuesto diseñar un modelo de inteligencia artificial capaz de aprender los hábitos horarios del cliente y de cerrar los cajones automáticamente por las noches o en los horarios en los que la casa se queda vacía.

El sistema de detección de proximidad ha sido descartado ya que presentaría una incomodidad a la hora de tener invitados, los cuales necesitan ser acompañados por el dueño del hogar para poder abrir cualquier cajón o mueble cerrado por el pestillo. Si bien es cierto que se podría implementar un botón para activar/desactivar el sistema de proximidad, creemos que la inteligencia del sistema se vería afectada.

En cuanto a la IA capaz de acomodarse a los horarios del cliente, nos parece una buena idea ya que los horarios de las personas van cambiando a lo largo del año y no sería inteligente gestionar el cierre con los mismos datos todo el año. La implementación de esta idea implicaría el uso de una base de datos que proporcionará los datos a la IA y la misma,

en función de sus datos y la hora actual del sistema, será capaz de cerrar o abrir los cajones.

El inconveniente más importante que plantea esta idea es que el usuario deberá ser el encargado de introducir, por medio de la aplicación Android o la página web, los datos para entrenar a la IA. Esta es una tarea que puede resultar incómoda y obliga al usuario a tener una franja de tiempo en la que el sistema aún se está acomodando a sus horarios y se comporta de manera incorrecta.

Para concluir, aunque es un proyecto que nos gustaría implementar, dotarlo de inteligencia hace que se vuelva incómodo para la práctica y más costoso en comparación con el resto de ideas que hemos planteado.

Medidas disuasorias:

Una idea cuyo objetivo es disuadir a un intruso que está intentando entrar o ya ha entrado en el hogar del cliente.

Para ello se deberán utilizar medidas disuasorias como una alarma de sonido estrepitoso o una fuente de luz muy potente que reduzca la visibilidad de los intrusos que ya han invadido el hogar y asustarlos para prevenir la llegada a zonas más vulnerables y, además, alertar a los habitantes de la vivienda de esa intrusión.

Además de las medidas explicadas en el párrafo anterior, también es común utilizar un sistema de humo antirrobo que reduce la visibilidad del asaltante y lo disuade de seguir con la intrusión.

Las ventajas de esta propuesta son su modularidad, no tener un dispositivo de disuasión no afecta al rendimiento de los demás; su simplicidad y fácil implementación, con el diseño adecuado la inserción de este sistema en el hogar no sería complicada ni presentaría mucha incomodidad para los habitantes del mismo; por último, el sistema permitirá que los vecinos o transeúntes cercanos puedan notar la intrusión en el hogar y llamar a las autoridades si los habitantes del hogar no pueden hacerlo.

Para implementar el sistema podemos desarrollar una aplicación Android o una página web que ofrezca la capacidad de activar el sistema de disuasión manualmente en caso de fallo de los sensores de intrusión que los activan.

El principal problema del proyecto es su dependencia de otro sistema encargado de detectar la intrusión en el hogar. Esto obliga a que su instalación esté condicionada por factores externos y por tanto, el mercado del producto se ve reducido.

A pesar de que su implementación sea sencilla y, con un diseño adecuado, discreta, el coste es elevado a la hora de usar medidas de disuasión verdaderamente eficientes como luces de alta potencia o sistemas de humo antirrobo. El sistema se deberá instalar en una

habitación, lo que significa que para ampliar su rango de actividad a toda la vivienda habría que replicar el sistema en las distintas habitaciones y eso dificultará la discreción y la comodidad de los habitantes de la vivienda.

Los intrusos pueden tener conocimiento previo del sistema de disuasión y llevar equipación que les proteja de las luces, los sonidos y el humo. Esto implica que nuestro sistema de disuasión funcionará sólo contra aquellos asaltantes que no conozcan su existencia en el hogar.

Este sistema no necesita una arquitectura de 4 etapas puesto que no tendría datos que transmitir al servidor. La capa de percepción del mismo estaría compuesta de actuadores activados por un servidor o por sensores de otro sistema. No hay datos que almacenar que no sean registros (logs) de las activaciones del sistema de disuasión y las aplicaciones o páginas web actuarían únicamente de “sensores” con los que el usuario puede activar el sistema.

Por último, la naturaleza del sistema hace prácticamente imposible dotarlo de inteligencia, la inteligencia del mismo sólo se podría conseguir si lo combinamos con alguna de las otras ideas propuestas para el proyecto, haciendo que esos sistemas sean los sensores de este sistema.

Conclusión:

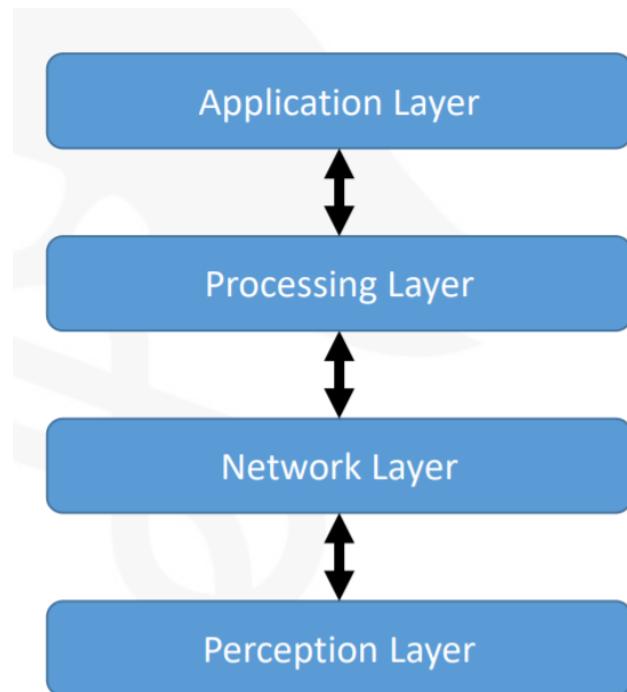
Para concluir esta sección, consideramos que estas ideas por sí solas no cumplen los requisitos y los objetivos que perseguimos con nuestro proyecto. Sin embargo, combinaciones como Sistema de intrusión+Cierre de seguridad o Mirilla Inteligente+Medidas de disuasión son combinaciones más cercanas a nuestros objetivos.

Estas dos combinaciones se han descartado frente a la combinación Mirilla Inteligente+Sistema de Intrusión debido a que el modelo SI+CS tiene poco margen de inteligencia y el modelo MI+MD sólamente proporciona seguridad en las puertas en las que el sistema está instalado.

Tecnología a utilizar

Arquitectura a utilizar

La arquitectura utilizada para implementar el prototipo está basada en una arquitectura de 4 capas, que sigue el enfoque ascendente del siguiente esquema:



Capa de percepción:

Es la capa donde se encuentran todos los sensores, actuadores, y controladores que usaremos para obtener información del entorno e interactuar con él.

Los sensores y actuadores que serán utilizados serán explicados más adelante. Sin embargo, nos surge un problema a la hora de elegir si vamos a trabajar con un ESP32, Arduino o una Raspberry Pi. Este problema involucra otras capas de la arquitectura así que lo dejaremos para el final de esta sección.

La capa de percepción hará uso de los sensores de proximidad para detectar una situación en la que un posible intruso tape la cámara, en este caso, el sensor de proximidad alertará de esta situación que podría convertirse en un intento de intrusión.

Se instalarán sensores de movimiento en las ventanas o zonas de la vivienda que puedan suponer una vulnerabilidad para la seguridad, estos módulos se conectarán al controlador principal (la mirilla) mediante Wi-Fi notificando cualquier movimiento.

El módulo de cámara, integrado en la mirilla, realizará una grabación constante de imágenes, imágenes que analizará para detectar la presencia de una persona. Si se detecta una persona frente a la puerta, se enviará la imagen que ha causado la detección a la capa de transporte para su posterior llegada a la capa de procesamiento.

En situaciones de oscuridad la cámara nos sería inútil, por lo que se instalará un actuador, una bombilla, capaz de iluminar la zona de captura cuando se detecte que la iluminación no es suficiente. Para esta detección utilizaremos un sensor de iluminación integrado en la mirilla con su foco de detección paralelo a la lente de la cámara, de esta manera, si alguien tapa la lente la cámara detectará la falta de luz.

Capa de transporte:

Capa encargada de conectar los sensores de la capa de percepción con los servidores para garantizar la transmisión de los datos recogidos a la capa de procesamiento.

Para lograr el objetivo descrito utilizaremos Wi-Fi, en concreto, utilizaremos el protocolo MQTT, protocolo que hace uso de la capa de transporte TCP y SSL para garantizar una transmisión segura, privada y libre de pérdidas.

El protocolo utilizado ofrece un servicio de mensajería PubSub, es decir, un sistema en el que existen publicadores, los cuales publican información en un hilo específico, y suscriptores, los cuales ven o descargan información de los hilos a los que están suscritos. Cabe destacar que la información de un hilo solo aparecerá visible para aquellos suscriptores suscritos a ese hilo.

Hemos seleccionado este protocolo por su simplicidad a la hora de implementarlo, porque es ligero, apropiado para dispositivos de baja potencia como los usados en nuestro proyecto y, además, está optimizado para ofrecer servicio a un gran número de clientes de forma simultánea.

Para garantizar la transmisión segura, privada y libre de pérdida de mensajes implementaremos un nivel de QoS (calidad de servicio) 2, estableciendo un enlace de cuatro pasos entre cliente y bróker (servidor que gestiona los mensajes).

Capa de procesamiento de datos:

Capa encargada de recoger los eventos del sistema, otorgando la capacidad de procesarlos, almacenarlos y actuar en consecuencia a ellos.

Esta capa será la encargada de procesar la información recibida de la capa de percepción. Esta información consiste en imágenes relevantes (que contienen caras y/o manos) e información sobre el estado de los sensores de movimiento, proximidad e iluminación.

Esta información será procesada por el servidor, bróker puesto que utilizamos MQTT, y en función de la misma será almacenada, transformada o se realizará alguna acción como consecuencia, como por ejemplo notificar un intento de intrusión.

También nos puede llegar información por parte de la aplicación, donde el usuario puede cambiar preferencias (desactivar sensores de movimiento) o puede solicitar el borrado de algún elemento de la base de datos.

Esta información es también procesada por el bróker, actuando en consecuencia modificando la base de datos o transmitiendo, por medio de la capa de transporte, la orden de parada a un sensor.

Finalmente, en esta capa se encuentran el bróker, encargado de gestionar el procesamiento y transmisión de datos; el gestor de bases de datos, encargado de gestionar las órdenes, queries, utilizadas para añadir o eliminar elementos de la base de datos; y el servidor web, encargado de almacenar, gestionar y comunicar la página web con el bróker y la base de datos.

Capa de aplicación:

Es la capa encargada de mostrar la información y el estado del sistema al usuario. Lo hará mediante una aplicación móvil para Android y una página web, descritas más adelante, encargadas de gestionar una interfaz de usuario e interactuar con el cliente, actuando según sus peticiones o notificando algún evento.

La capa de aplicación se comunicará con la de procesamiento usando la misma arquitectura de red que la comunicación procesamiento-percepción. Es decir, los dispositivos de la capa de aplicación se suscribirán a los hilos que necesitan y publicarán en el hilo de peticiones android y web, respectivamente.

Problema:

La inteligencia de nuestro proyecto se basa en su totalidad en la capacidad para analizar la imagen de la cámara y detectar la presencia de rostros y/o manos en la imagen. Este análisis se realiza mediante Inteligencia Artificial, en concreto, un algoritmo que analizará constantemente la imagen recibida por la cámara.

Surge el siguiente problema: ¿Dónde se ejecutará el modelo de IA, en el bróker o en la capa de percepción?

Realizar el análisis en el bróker implica una transmisión constante de imágenes entre los sensores y el servidor. La calidad de este análisis depende de la tasa de transferencia de la red y del módulo Wi-Fi del controlador así como la de la tarjeta de red del servidor.

Otra opción es realizar el análisis directamente en el controlador, pero esto limita la elección del microcontrolador a una única opción, la Raspberry Pi. La RPi es el único controlador con la capacidad de ejecutar un código en Python que realice el cómputo del modelo de IA.

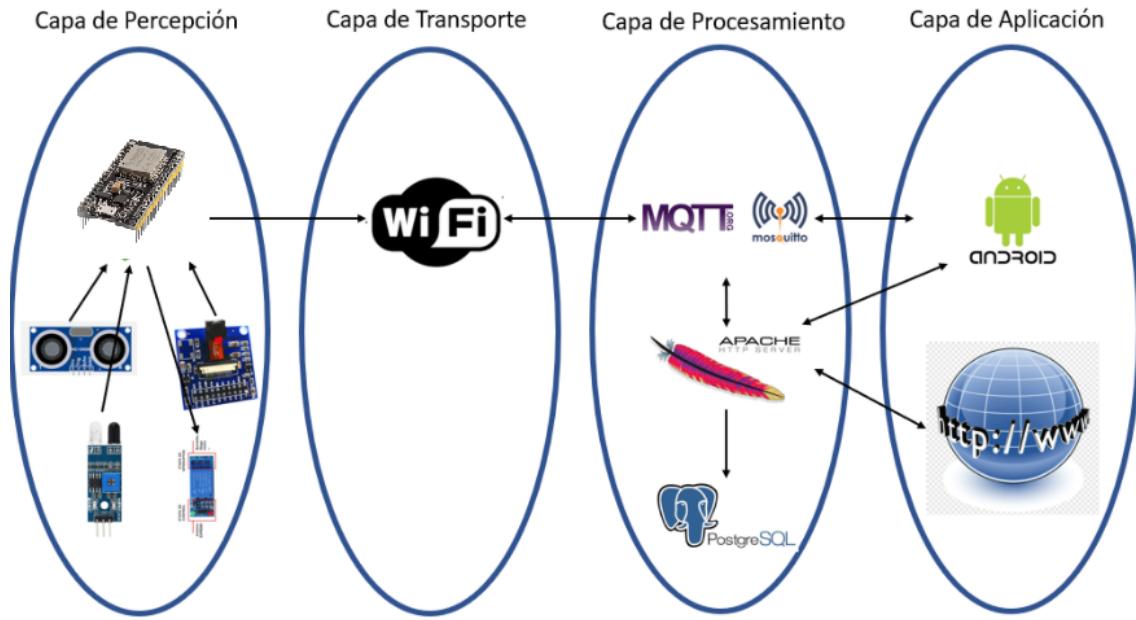
El análisis se realizaría directamente sobre la imagen recibida de la cámara y sólo se transmitirán las imágenes relevantes. La calidad de este análisis depende del algoritmo y los recursos de la RPi.

Tras analizar lo explicado anteriormente hemos decidido mantener una arquitectura de 4 capas en la que sólo se transmitirán imágenes cuando el sensor de proximidad detecta un objeto a menos de una distancia determinada.

De esta manera reducimos la transferencia de imágenes a las situaciones en las que hay alguien frente a la puerta, lo que aligera la carga en la red y reduce el trabajo computacional de la capa de procesamiento.

Arquitectura final:

Después de todos los detalles explicados anteriormente podemos concluir un diseño de arquitectura como el que sigue:



Listado de Sensores utilizados

< Listado de los sensores utilizados en la capa de percepción. Cada sensor listado deberá tener una descripción de su funcionalidad, imágenes del mismo y un listado de las librerías a utilizar si las hubiera >

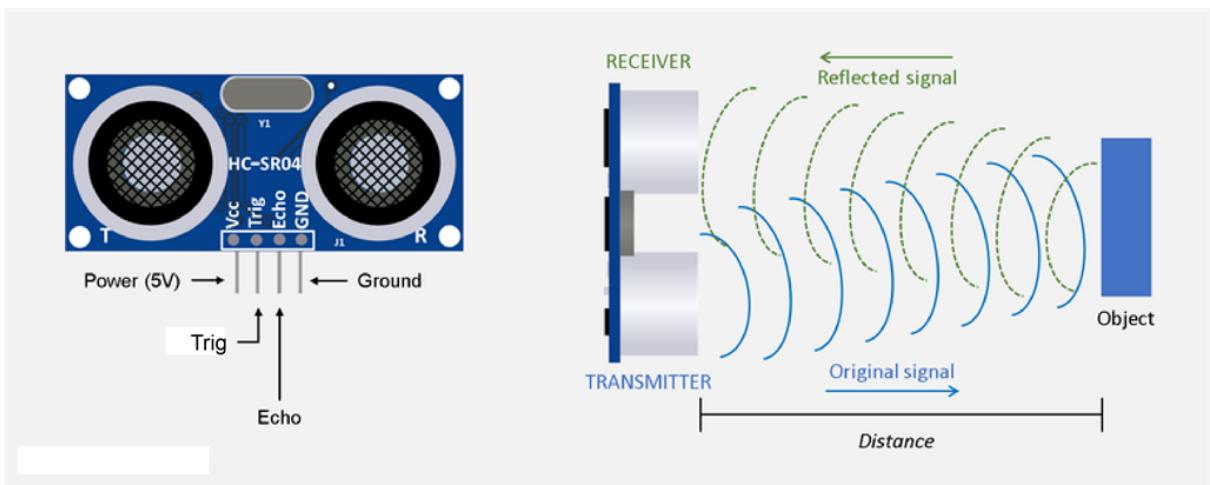
En esta sección enumeraremos los sensores que serán utilizados en la capa de percepción:

Sensor de proximidad:

Como sensor de proximidad usaremos un sensor de ultrasonidos **HC-SR04**, un módulo utilizado para medir la distancia al objeto más cercano frente al sensor.

La medida se obtiene por medio de ondas de ultrasonido, el módulo tiene un emisor y un receptor con los que en función del tiempo que tarda la onda en llegar al receptor podemos calcular la distancia al objeto.

El cálculo se puede hacer mediante una librería característica del módulo o mediante unas operaciones simples en arduino, hemos decidido no usar la librería ya que las operaciones son muy simples y podemos programar nuestra propia función.

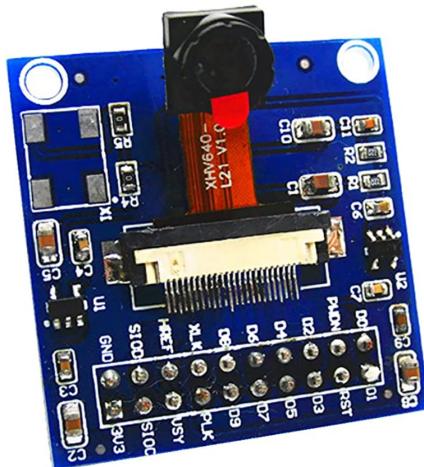


Cámara corriente:

Para la captura de imágenes vamos a utilizar un módulo cámara **OV2640** de Arduino, una cámara de 2 megapixeles que trabaja a poco voltaje y puede capturar imágenes hasta a 15 fotogramas por segundo.

El módulo cámara será utilizado para transmitir imágenes de manera constante al servidor donde serán procesadas y almacenadas.

Para trabajar con este módulo haremos uso de la librería ARDUCAM en arduino, una librería que facilita el tratamiento y la transferencia de imágenes con arduino.



Sensor de movimiento:

Para controlar el movimiento en las ventanas o zonas distintas de la puerta principal haremos uso de sensores de movimiento, en concreto utilizaremos sensores infrarrojos para detectar cualquier tipo de movimiento en las zonas mencionadas.

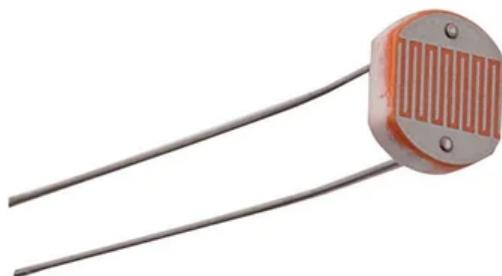
Para controlar el módulo mediante arduino no hay necesidad de utilizar ninguna librería, con un simple analogRead podemos detectar el movimiento frente a él.



Célula fotovoltaica:

Una célula fotovoltaica encargada de medir el nivel de iluminación en la zona de grabación. Este componente se conecta a un pin analógico y con un simple AnalogRead podemos obtener el nivel de iluminación detectado.

De esta manera, nuestro sistema de percepción puede conocer la iluminación a la que está sometida la lente y reaccionar encendiendo el actuador explicado más adelante.



Listado de Actuadores

En cuanto a los actuadores en la capa de percepción utilizaremos únicamente un módulo que permita emitir luz.

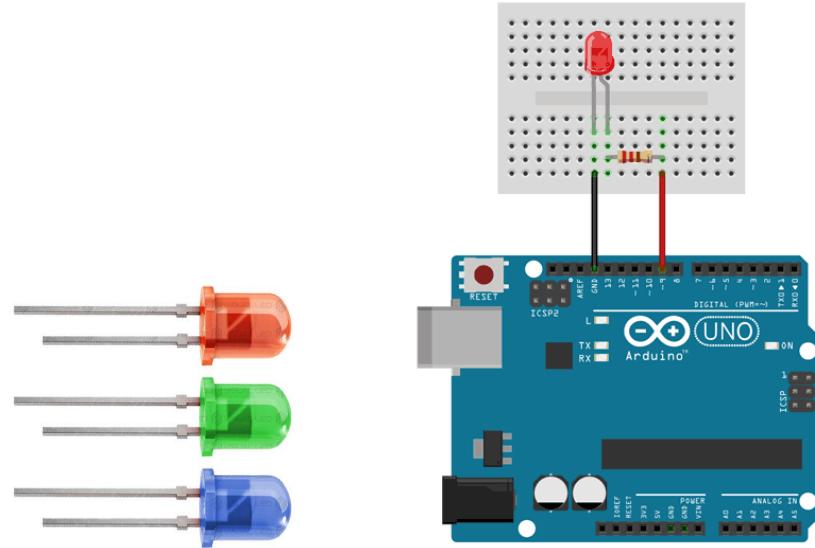
Para nuestro proyecto este módulo cumplirá la función emitir una fuente de luz con el objetivo de reducir la visibilidad de intrusos para prevenir los robos y lograr disuadir a los posibles asaltantes.

Podemos darle dos enfoques:

LED

Podemos realizar una simulación con un diodo emisor de luz LED. Esto sería una forma de simular un gran foco de luz pero adaptado a nuestro pequeño prototipo.

Realizarlo de esta forma nos da muchas ventajas como puede ser la simplicidad, ya que no haría falta trabajar con voltajes altos (como podría ser en una bombilla) o programar ningún módulo más avanzado, o que sea más económico, los LED no suponen ningún gasto económico.



No es necesario el uso de librerías.

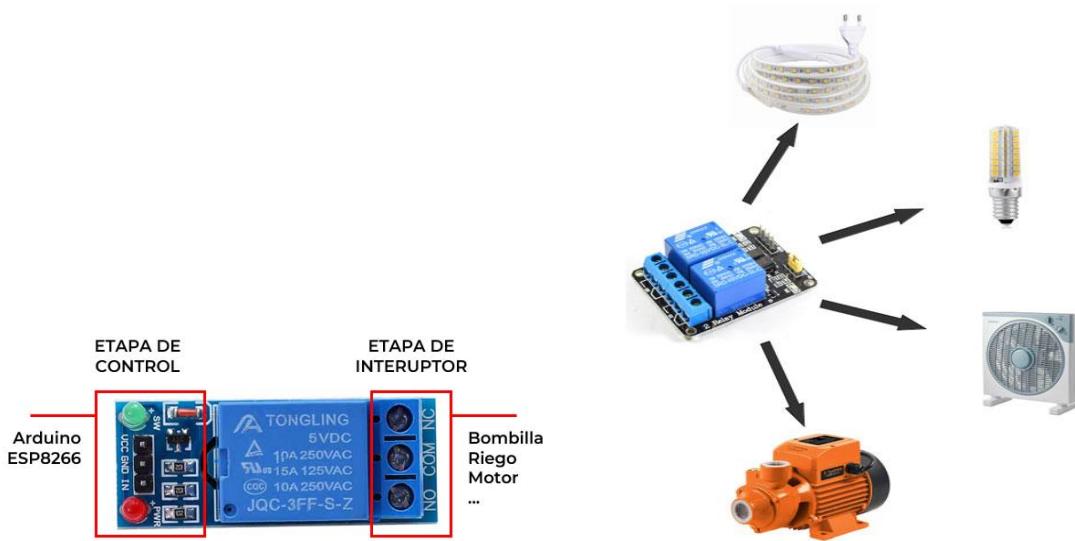
El LED se enciende mediante el comando digitalWrite. Se indica HIGH para encender el pin y LOW para apagarlo.

Ej: `digitalWrite(LED,HIGH)`.

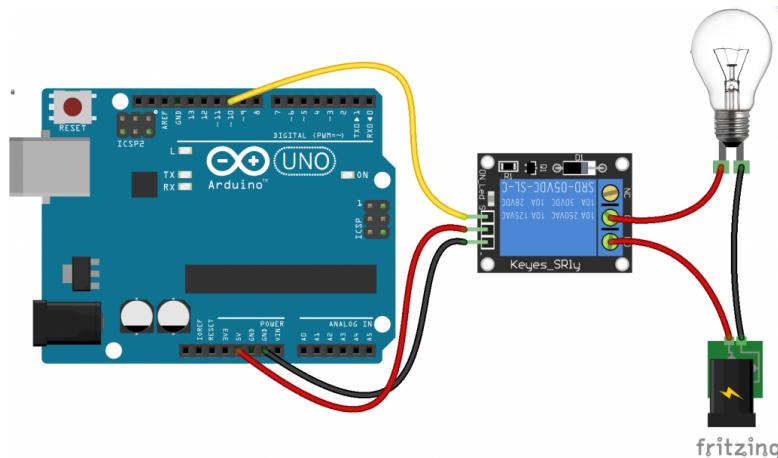
Foco más potente.

Por otra parte, podemos ir un poco más allá y utilizar algún tipo de bombilla o linterna más potente. La única ventaja es que dará más luz e incluso puede llegar a cegar dependiendo de la potencia de la linterna.

En este caso necesitaríamos un relé. Un relé (relay) es un dispositivo que es utilizado para controlar el encendido y apagado de dispositivos que requieren altos voltajes o altas corrientes mediante una señal de corriente continua como las proporcionadas por los pines de arduino.



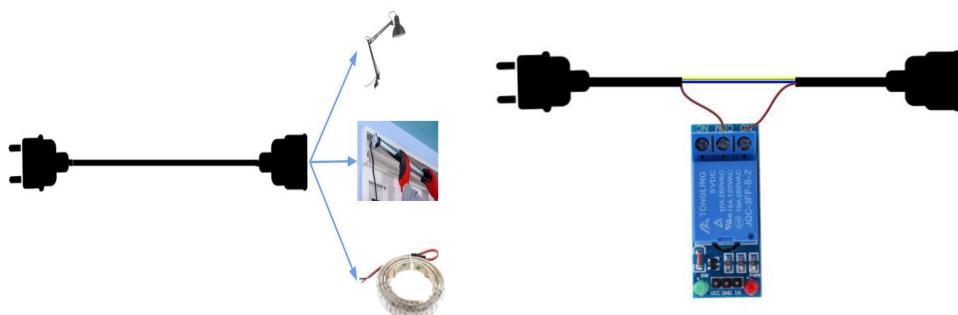
La etapa de control es la encargada de controlar la corriente que circula por la bobina o el inductor. Trabaja con corriente continua y el voltaje y amperaje de las salidas digitales suele ser suficiente para controlar el relé con Arduino. La otra etapa funciona como un interruptor de corrientes y voltajes altos. Es donde conectamos la bombilla, un motor o cualquier dispositivo que queramos controlar.



Tampoco necesitaría librerías ya que en cuanto a código es igual de sencillo que un LED.

Para conectarlo a la corriente necesitamos un alargador de forma que un extremo del cable irá conectado a un enchufe y el otro extremo al dispositivo que quieras controlar.

De la forma:



Todo esto es un proceso demasiado costoso en cuanto a tiempo y complejidad. Por esto al no ser una parte importante del proyecto, sino que simplemente es un añadido de acción-reacción no profundizaremos mucho más en este enfoque.

Microcontroladores a utilizar

Para el prototipo deseado se nos presentan tres opciones de microcontroladores: una tarjeta Arduino Uno junto con un módulo Wi-Fi, un microcontrolador ESP32 con Wi-Fi integrado o una placa Raspberry Pi.

El precio del RPi resulta elevado y la capacidad computacional de un Arduino o un ESP32 es más que suficiente para el prototipo propuesto. Por tanto, de entre las opciones restantes hemos decidido quedarnos con el microcontrolador ESP32 ya que al tener wifi integrado se adapta perfectamente a nuestras necesidades y podemos programarlo con Arduino.

El ESP32 es un microcontrolador que se conecta a una protoboard con todos sus pines, una vez colocado podemos darle corriente por sus pines de 5V y GND. El resto de componentes se conectarán, mediante la protoboard, a los pines GX. Por último, para cargar el programa arduino en el chip usaremos un cable de datos tipo C conectado al ordenador.



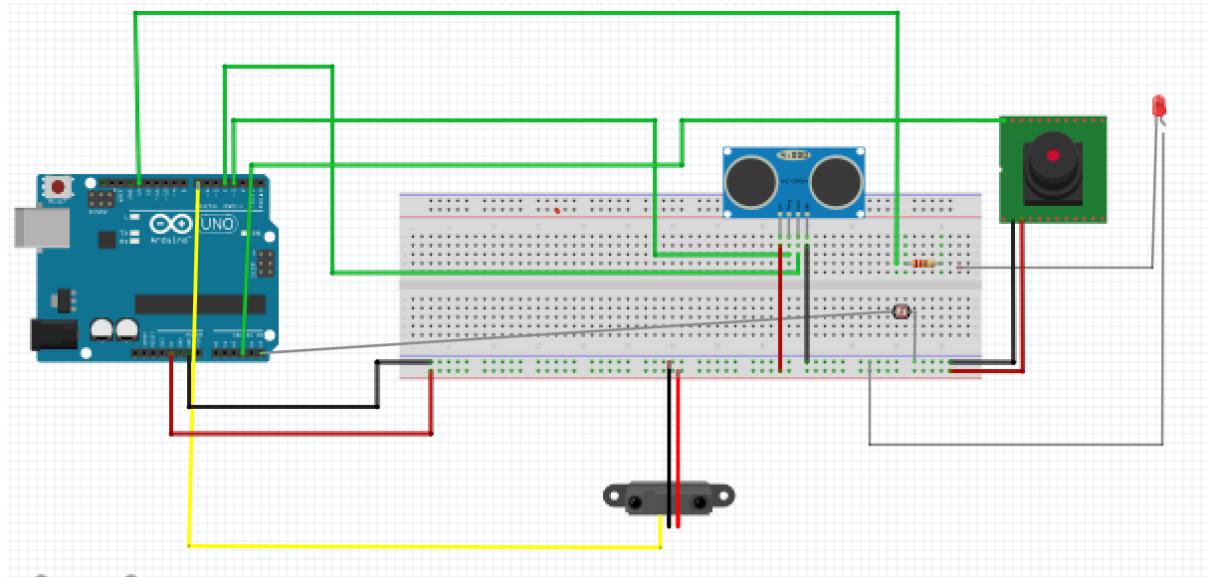
Boceto del diseño del circuito con fritzing

Hemos utilizado la herramienta fritzing para diseñar un boceto del circuito del prototipo. Por desgracia fritzing no contiene exactamente los componentes que vamos a utilizar así que hemos utilizado otros lo suficientemente parecidos para suplantarlos en este boceto.

Nótese que las conexiones son meramente simbólicas, para dar una idea de las conexiones del sistema. En el boceto se usa un Arduino Uno que en la práctica sería un dispositivo ESP32, también usamos un sensor de proximidad infrarrojo cuando en la práctica usamos un sensor de movimiento infrarrojo. Finalmente, el módulo cámara utilizado es distinto al seleccionado para el prototipo.

Una vez especificados los cambios en el boceto, podemos describir el funcionamiento del hardware. Los módulos cámara, ultrasonidos y fotocélula funcionan juntos, y los datos

recibidos por ellos podrían encender el led que hace alusión a una linterna. Por último, se nos presenta un detector de movimiento infrarrojo que analizará la actividad en zonas distintas a la puerta principal.



Lenguajes de programación a utilizar

Para hablar de los distintos lenguajes de programación que vamos a utilizar en este proyecto, tenemos que tener claro qué Software vamos a utilizar en cada desarrollo y para ese Software hablar del lenguaje:

- Para programar en Arduino: el lenguaje estándar, y en el que se basa es C++, teniendo una gran similitud a la hora de tratar la mayoría de los tipos de datos (array, bool, char, float...). Hemos preferido el uso de este, debido a que todos estamos familiarizados con el mismo al haberlo usado en proyectos anteriores, además como características principales del mismo como ventajas para nuestro proyecto cabría resaltar:
 - La compatibilidad con bibliotecas: ya que hay muchas funciones que están disponibles para todos a través de bibliotecas y que nos ayudarían a escribir código rápidamente.
 - Compilación: al ser necesario compilar el código de bajo nivel antes de ejecutarse, hace que sea más eficiente con el hardware.
 - Un uso fácil: ya que como hemos comentado ya hemos trabajado con C++, y además hay una gran comunidad en la que podemos apoyarnos para buscar información o en la resolución de algunos problemas que se nos puedan plantear durante el proyecto.
- Para programar en la base de datos usaremos el sistema de PostgreSQL: usaremos SQL al ser uno de los lenguajes más utilizados para trabajar con conjuntos de datos y la relaciones entre ellos. Es un tipo de lenguaje que nos permitirá gestionar una base de datos prácticamente en su totalidad podríamos el lenguaje en 3 tipos de sentencias como :

- Lenguaje de Definición de Datos (DDL): ya que puede definir y crear los objetos que soportan las bases de datos.
 - Lenguaje de Manipulación de Datos (DML): ya que nos sirve para manejar los datos almacenados en las tablas , a nivel de campos o registros.
 - Lenguaje de Control de Datos (DCL): ya que lo podemos usar para controlar las funciones de administración y control de las bases.
- Para el desarrollo de la aplicación en Android usaremos el entorno de Android Studio: de manera que el código que usaremos será Java ya que como el caso de C++ estamos más familiarizados con el mismo al haber trabajado anteriormente con este. Además como características ventajosas de este lenguaje podemos hablar de:
 - Es Open Source: ya que disponemos de una gran cantidad de funcionalidades base con el uso de bibliotecas y además encontramos multitud de código de terceros que podemos usar en cualquier momento.
 - Por su versatilidad: ya que es una plataforma que nos permite la creación de aplicaciones de escritorio e interfaces de una forma muy sencilla.
 - Multiplataforma: ya que Java lo podemos usar para que funcione en cualquier servidor y sistema operativo.
- Para la integración de inteligencia en nuestro sistema (de manera que sea un sistema inteligente): usaremos Python. Este es un lenguaje multiplataforma, multiparadigma y de propósito general, es decir, que lo podemos usar en varios ámbitos tanto para hacer páginas web como para desarrollar programas inteligentes haciendo uso de sus bibliotecas como Numpy,que son utilizadas para computación científica y aprendizaje automático. Es una opción flexible al ser independiente de la plataforma en la que se use. Además también es un lenguaje con el que ya hemos trabajado numerosas veces, y que nos resulta mucho más intuitivo y simple a la hora de programar. De manera que como características principales encontramos:
 - Es multiplataforma: al ser usado en diversos tipos de sistemas.
 - Es multiparadigma: funciona tanto para la programación orientada a objetos como funcional, o imperativa.
 - De libre distribución.
 - Tiene varias librerías como pandas que nos sirven para leer rápidamente una gran cantidad de datos.

Software a utilizar

Como software que vamos a utilizar hablaremos de manera más extensa de los entornos mencionados anteriormente, así como del servidor web que vamos a usar.

IDE de Arduino:

Atendiendo a su definición Arduino es una plataforma de desarrollo de hardware y software de código abierto, en la parte hardware encontramos las placas, y toda clase de sensores y actuadores. Por la parte del software encontramos un programa de aplicación, que consta de un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). Como características principales del mismo encontramos:

- Es una plataforma de código abierto y que cuenta con una gran comunidad para que la reutilización de código sea más sencilla.
- Multiplataforma: ya que se puede utilizar tanto en Windows, como en MacOS y Linux
- Que tiene un entorno de programación simple y claro.

Y además destacar lo económicos que pueden llegar a ser sus componentes a la hora de desarrollar cualquier tipo de proyecto.

[Entorno de PostgreSQL:](#)

PostgreSQL es un sistema para gestionar bases de datos de muy alto nivel, cuya licencia es totalmente abierta, además es consistente y tolerante a fallos. Entre sus principales características encontramos:

- Alta concurrencia: ya que puede atender a varios clientes al mismo tiempo.
- Soporte a triggers: permitiendo que definamos eventos y generemos acciones asociadas a las tablas cuando estos se lancen.
- Ofrece mecanismos de programación orientada a objetos.
- Da soporte a gran cantidad de lenguajes: siendo capaz de trabajar con funciones internas que se ejecuten en el servidor, que hayan sido escritas en diversos lenguajes como C, C++, Java, Python.

[IDE de Android Studio:](#)

Es el entorno de desarrollo integrado oficial para el desarrollo de aplicaciones para Android. Tiene plantillas integradas que nos facilitan la creación de diseños y código. Un espacio virtual que nos sirve para ejecutar y probar las aplicaciones en tiempo real. Como fases que veremos a la hora de desarrollar nuestra aplicación en Android studio tendremos:

- La configuración del entorno: ajustando el entorno de desarrollo y realizando la conexión a los elementos donde instalaremos la app, y crear los dispositivos virtuales de Android.
- Configuración del proyecto y desarrollo: creando los módulos que contengan los recursos para la aplicación y los archivos de código fuente.
- Hacer las pruebas, depuración y construcción de la aplicación, construyendo el proyecto en un paquete/s depurables .apk que podremos instalar en nuestros dispositivos, utilizando un sistema de construcción basado en Gradle, el cual es un sistema que automatiza la construcción de nuestro código.

[Servidor web de Apache:](#)

Apache es un software que se ejecuta en un servidor, cuyo objetivo es que podamos establecer conexión entre un servidor y los navegadores, mientras se envían archivos entre ellos como en la estructura cliente-servidor. Es un software multiplataforma, pudiendo funcionar tanto en Unix como en Windows. Además tiene una estructura basada en módulos, lo que nos permite activar o desactivar funcionalidades cuando queramos, estos módulos pueden ser de seguridad, almacenamiento en caché, autenticación de contraseña...

Como características principales de Apache vemos que es:

- De código abierto y gratuito.
- Flexible debido a su estructura basada en módulos
- Multiplataforma

Características de la app móvil

El desarrollo de la aplicación móvil para Android consistirá en un sistema sencillo basado en Actividades, entre las cuales las más importantes serán la actividad de “Inicio de Sesión” y “Monitorización”. Además, el sistema ofrecerá una actividad “Configuración” que otorga la capacidad de personalizar la aplicación añadiendo el nombre del cliente, preferencias de alertas y configuraciones sobre la captura de imágenes (si quiere grabar o sólo hacer una captura).

La aplicación se programará haciendo uso de Android Studio, en concreto por medio del lenguaje de programación Java. Se comunicará con el sistema a través de internet para recibir información del mismo y notificar cambios en la configuración o respuestas a notificaciones.

Presentamos un boceto simple del diseño de la aplicación móvil:

Actividad de login:

Actividad en la que el usuario introducirá su email y contraseñas y, si la información proporcionada es válida, se le concederá el inicio de sesión a su perfil con sus correspondientes configuraciones.

Vista previa:



Email _____

Password _____

LOG IN

REGISTER

Actividad principal:

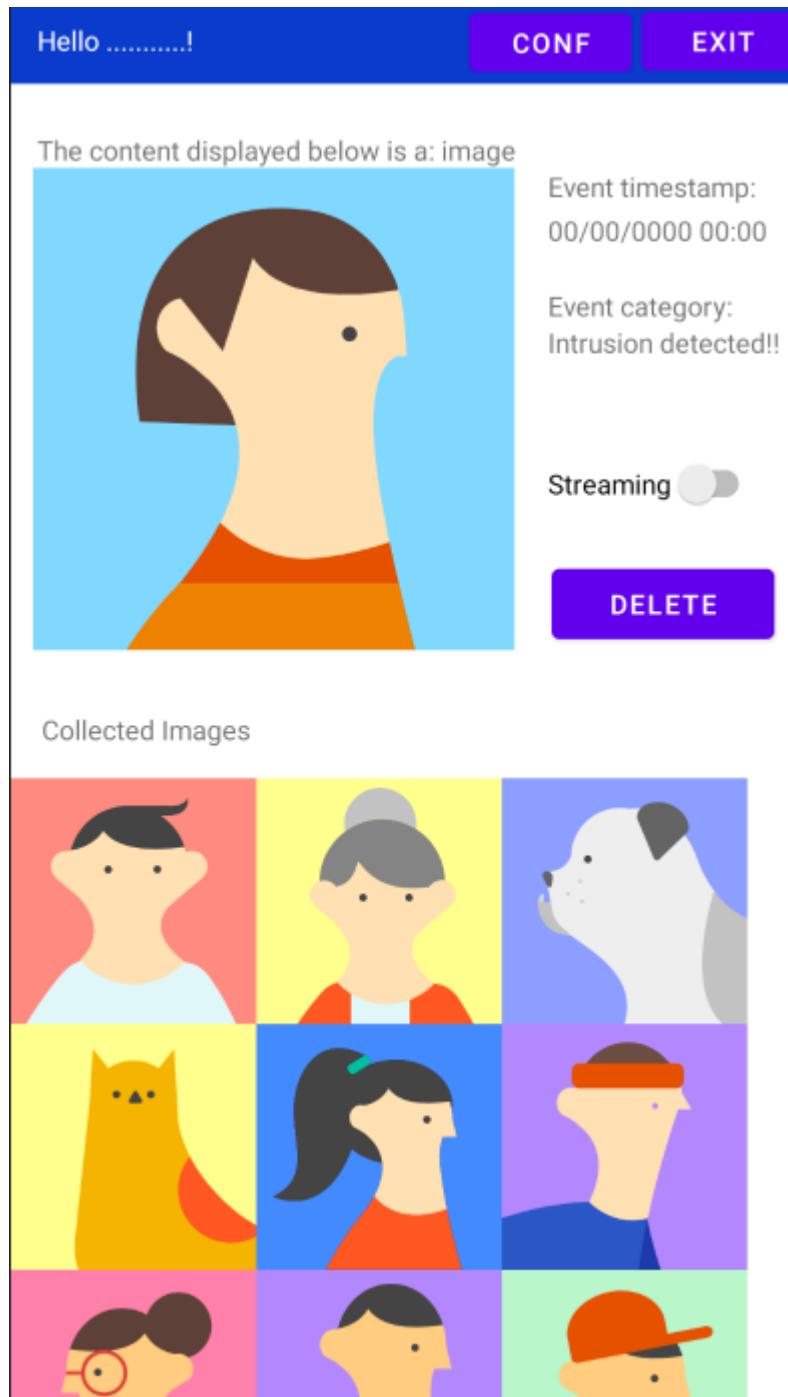
Actividad en la que se presenta al usuario una vista de la imagen en vivo (Streaming) o una imagen de la galería con información como el momento de la captura y una descripción inteligente de la situación frente a su puerta.

El usuario puede cambiar a modo Streaming usando el switch button presente en la interfaz, esto hará que el botón delete desaparezca ya que solo sirve para borrar imágenes de la galería.

La galería se muestra en la parte inferior de la actividad, ofreciendo, ordenadas por fecha de captura, todas las imágenes y vídeos capturadas por la mirilla.

Finalmente el usuario puede acceder a la actividad de configuración por medio del botón “Conf” o cerrar sesión pulsando “Exit”

Vista previa:



Actividad de configuración:

Se presentan al usuario dos secciones: configuración de información personal, donde el usuario podrá modificar su información personal (nombre, correo, contraseña) y la configuración de la aplicación, donde el usuario podrá especificar sus preferencias acerca del comportamiento del sistema en su vivienda.

Una vez realizadas las modificaciones necesarias el usuario puede guardar y actualizar la configuración del sistema pulsando “Save & Update”, volver a la actividad principal pulsando “back” o cerrar sesión en la aplicación pulsando “exit”.

Vista previa:

The screenshot shows a mobile application interface for configuration. At the top, there is a blue header bar with the text "Hello!" on the left and "BACK" and "EXIT" buttons on the right. Below the header, the title "Personal settings" is displayed. The form contains four input fields: "User Name:" with a blank line, "Email:" with a blank line, "Password:" with a blank line, and "Confirm password:" with a blank line. Below these is a section titled "App settings" with four toggle switches: "Send notifications" (on), "Capture photos" (on), "Life Stream Capture" (off), and "Capture videos" (off). At the bottom is a large purple button labeled "SAVE & UPDATE".

Hello

BACK EXIT

Personal settings

User Name:

Email:

Password:

Confirm password:

App settings

Send notifications

Capture photos

Life Stream Capture

Capture videos

SAVE & UPDATE

Cuando la aplicación sea cerrada por el usuario usando los botones del sistema operativo, esta cerrará sesión y ejecutará en segundo plano un hilo encargado únicamente de recibir información del servidor y crear notificaciones. Las notificaciones solo tendrán la opción de ser pulsadas, acción que abrirá la aplicación pidiendo al usuario que inicie sesión y llevándolo al evento notificado.

Características de la página web

La página web será un sitio web destinado a monitorizar la actividad del sistema. En esta cada usuario del sistema tendrá un usuario y contraseña que usará para acceder a la web. Una vez en esta el usuario podrá acceder a diferentes secciones:

- Configuración: en este apartado de la web podremos personalizar la recepción de las imágenes ya sea grabar o obtener fotos.
- Biblioteca: esta será la página principal de la web donde podremos encontrar ordenadas por fecha todas las capturas y grabaciones que se han ido almacenando.
- Observación: aquí podremos ver en nuestro monitor , a tiempo real, lo que está grabando nuestra cámara.

La web será desarrollada mediante la plataforma Wix.

Con URL: <https://www.seuria.es>

Cabe añadir que a diferencia de la aplicación las notificaciones no llegarán hasta que uno no se registre en la página web por lo tanto la web se enfocará más en monitorizar la mirilla.

Página principal:

Al acceder a la web, antes de acceder a nuestra cuenta, encontraremos una introducción y exposición de nuestro producto. Esta página tiene el objetivo de convencer a otros usuarios que no forman parte del sistema a que se unan a él.

Por otra parte, desde esta página tendremos la opción de registrarnos o de iniciar sesión desde el botón de “Estar seguro” situado en la esquina superior derecha.

Vista previa:

The screenshot shows the homepage of the Securia website. At the top left is the logo 'Securia'. At the top right are three buttons: 'Home' (in red), 'Importancia de la seguridad', and 'Estar seguro' (in a red box). The main title 'Por qué Securia?' is centered at the top in a large, bold, white font. Below the title is a paragraph of text in a smaller white font. At the bottom left is a red button labeled 'Saber más'. At the bottom right is a small red square icon with a white speech bubble symbol. The background features a dark blue gradient with a subtle grid pattern.

Log In:

Una vez en el sitio web podremos acceder a nuestra cuenta a través de un Log in verificando nuestro email y contraseña.

Vista previa:

Email
Introduzca su dirección de email

Clave
Introduzca su clave

Aceptar

Registro:

En el caso de no estar registrado el usuario accede a una pantalla de registro del usuario en la cual añadiendo sus datos personales junto a la clave del producto podrá empezar a formar parte del sistema.

Vista previa:

Regristro de usuario

Primer nombre Apellidos

Email * Compañía

Teléfono Clave

Clave de producto Confirmar clave

Aceptar

Configuración:

En este apartado podremos cambiar nuestros datos personales y el comportamiento del sistema.

Una vez finalizadas las modificaciones, clicamos en el botón “Aceptar” para guardar los cambios.

Vista previa:

<p>Configuración del Sistema</p> <p>Capturar fotos SI Capturar videos NO Capturar en vivo SI</p>	<p>Configuración Datos Personales</p> <p>Primer nombre Apellidos <input type="text"/> <input type="text"/> Email * Compañía <input type="text"/> <input type="text"/> Teléfono Clave <input type="text"/> <input type="text"/></p> <p style="background-color: red; color: white; text-align: center; padding: 5px;">Aceptar</p>
--	---

Biblioteca y Observación:

Por último, una vez ya iniciada la sesión, con su respectiva cuenta, accederemos directamente al modo biblioteca de archivos.

El usuario podrá intercambiar con solo un click entre el modo Biblioteca, en el que puede ver cada archivo (los cuales estarán ordenados por fecha para facilitar su seguimiento) recogido por la mirilla en el pasado, o el modo Observación donde podrá ver por vía streaming, es decir, en directo, lo que está ocurriendo en la puerta de su vivienda a través de la mirilla.

(El boceto nos pareció demasiado complicado para los conocimientos actuales, consistiría en una sección que muestre una serie de imágenes en un scroll que podríamos agrandar. Además, el boceto incluiría un botón con el que cambiar al modo observación, en el que ver la imagen en vivo que está siendo grabada por la mirilla)

Planificación temporal, plan de desarrollo

Semanas	Objetivos	Responsable	Anotaciones
1	Comprar componentes Diseñar IU app android Diseño del sistema de archivos y BBDD	Todos Alberto, Edu Álvaro	--- --- Solo modelo relacional
2	Diseño del circuito hardware Implementación del sistema de archivos	Alex, Edu Álvaro	Boceto ---
3	Configuración del servidor Apache Sensor de proximidad y cámara	Alberto Alex	Para web general Módulos independientes

4	Sensor de movimiento Sensor de detec. luz	Alberto Álvaro	Módulos independientes Módulos independientes
5	Juntar sensor de iluminación con foco de luz Comunicación Wi-Fi con broker Ensamblaje del programa Arduino	Edu Alex Todos	--- --- Juntado de los módulos independientes
6	Conectar Apache y Unix Sistema PubSub Mosquitto Conectar Apache y Mosquitto	Álvaro, Alberto Alex, Edu Todos	--- --- ---
7	Actividad login Actividad configuración	Alberto, Edu Alvaro, Alex	Implementación backend Implementación backend
8	Diseño del modelo de reconocimiento facial	Alex	Diseño e implementación
9	Diseño de IU web Montaje de circuito arduino	Álvaro, Alex Todos	--- Montaje y pruebas
10	Actividad monitorización Inicio de sesión y registro web	Todos Álvaro, Alex	Implementación backend Implementación backend
11	Biblioteca y observación web Página principal web Notificaciones android	Todos Álvaro, Alex Alberto, Edu	Implementación backend Implementación backend Implementación backend
12	ENTREGA	Todos	---

Resumen y conclusiones

La vivienda es el lugar que la gente habita, generalmente, con sus familias y guardan pertenencias de gran valor económico y personal; y donde quieren sentirse seguros. Por tanto, es un lugar el cual se quiere mantener seguro de robos, intrusiones, etc, por lo que la seguridad de la vivienda es un asunto que preocupa a todos. Pero siempre ha habido gente con la intención de acceder a viviendas ajenas para robar o con otros motivos de desagrado para los habitantes de las mismas. Así que hay un gran interés por asegurar las viviendas para evitar dichas intrusiones.

Nuestro proyecto se centra en dos sistemas, la mirilla inteligente que avisará al usuario si hay alguien delante de la puerta ofreciéndole imagen en tiempo real de lo que está sucediendo allí; y de los sensores de intrusión que detectan cualquier intrusión en la vivienda y notifica al usuario con toda la información obtenida de la intrusión.

Como hemos podido ver, el sistema consta de una arquitectura de 4 capas capaz de mantener la seguridad de la vivienda de manera inteligente haciendo uso de una cámara y distintos sensores de movimiento, proximidad, etc. Para poder gestionar el sistema y

hacerlo cómodo para el cliente el sistema consta de una aplicación móvil y una página web con las que se podrá monitorear toda la actividad en la puerta principal y las zonas controladas por los sensores.

Bibliografía

- [Cámara inteligente montada](#)
- [Módulo cámara a USB](#)
- [Módulo cámara a Arduino](#)
- [Top 10 mejores cámaras Arduino](#)
- [Módulo Bluetooth que Alex ya ha utilizado](#)
- [Relé Wifi Arduino](#)
- [Módulo ultrasonidos Arduino](#)
- [Vídeo Sistema de alarma y sensor ultrasónico con Arduino](#)
- [Vídeo para automatizar una máquina de humo con Arduino](#)
- [Máquina de humo](#)
- [Demostración Cámara OV2640](#)
- [Módulo infrarrojo Arduino](#)
- [Ejemplo de código arduino con IR](#)
- [Descripción general de notificaciones Android](#)