

TRABALLO FIN DE GRAO  
GRAO EN ENXEÑERÍA INFORMÁTICA  
MENCIÓN EN ENXEÑARÍA DE COMPUTADORES

# **Dispositivo e aplicación Android para aumentar a seguridade nos desprazamentos en bicicleta**

**Estudiante:** Sergio Rodríguez Gayoso  
**Director/a/es/as:** Carlos Vázquez Regueiro

A Coruña, 21 de agosto de 2019.



*Dedicatoria*



### **Agradecementos**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.



## **Resumo**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

### **Palabras chave:**

- First itemtext
- Second itemtext
- Last itemtext
- First itemtext
- Second itemtext
- Last itemtext
- First itemtext





# Índice Xeral

---

<b>1</b>	<b>Introdución</b>	<b>1</b>
1.1	Obxectivos . . . . .	1
1.2	Metodoloxía . . . . .	2
1.3	Proposta . . . . .	2
1.4	Traballo relacionado . . . . .	3
1.5	Estrutura da memoria . . . . .	3
<b>2</b>	<b>Análise e planificación</b>	<b>5</b>
2.1	Análise . . . . .	5
2.2	Planificación . . . . .	7
2.2.1	Fase 1: Dispositivo principal . . . . .	7
2.2.2	Fase 2: Aplicación do dispositivo móbil . . . . .	7
2.3	Custos do proxecto . . . . .	8
<b>3</b>	<b>Deseño e implementación do dispositivo</b>	<b>9</b>
3.1	Esquema xeral do dispositivo HW . . . . .	9
3.1.1	Placas de desenvolvemento . . . . .	9
3.1.2	Luces led . . . . .	12
3.1.3	Cámara . . . . .	13
3.1.4	Alimentación e baterías . . . . .	14
3.1.5	Software a utilizar . . . . .	15
3.1.6	Caixa e ancoraxes a bicicleta . . . . .	16
3.2	Leds . . . . .	16
3.2.1	Conexión coa Raspberry Pi . . . . .	16
3.2.2	Software de Control . . . . .	17
3.3	Camara . . . . .	18
3.3.1	Lentes . . . . .	19
3.3.2	Captura de video . . . . .	19

3.4	Transmisión de video . . . . .	20
3.5	Recepción de ordes . . . . .	20
3.6	Conexión co servidor . . . . .	21
3.7	Autoarranque do servidor . . . . .	21
3.8	Alimentación e enerxía . . . . .	22
3.9	Carcasa e anclaxe . . . . .	24
3.10	Custo do dispositivo . . . . .	25
<b>4</b>	<b>Arquitectura e implementación da aplicación</b>	<b>29</b>
4.1	Esquema xeral da aplicación . . . . .	30
4.2	Actividade principal . . . . .	30
4.2.1	Layout . . . . .	31
4.2.2	Ciclo de vida da actividade . . . . .	31
4.2.3	Botones . . . . .	32
4.2.4	Sensores . . . . .	33
4.2.5	Superficie de video . . . . .	33
4.3	Comunicación co dispositivo . . . . .	33
4.3.1	Broadcast . . . . .	33
4.3.2	Conexión . . . . .	34
4.3.3	Transmisión de ordes . . . . .	34
4.4	Recepción de video . . . . .	34
4.5	Anclaxe a bicicleta . . . . .	34
<b>5</b>	<b>Evaluación experimental</b>	<b>37</b>
5.1	Consumo e autonomía . . . . .	37
5.2	Vídeo e lentes . . . . .	38
5.3	Visibilidade . . . . .	38
5.4	Estabilidade e consumo da aplicación . . . . .	38
<b>6</b>	<b>Conclusións</b>	<b>39</b>
6.1	Traballo futuro . . . . .	39
.1	Contido DVD . . . . .	41
.2	Requisitos e instalación . . . . .	41
<b>A</b>	<b>Glosario de acrónimos</b>	<b>43</b>
<b>B</b>	<b>Glosario de termos</b>	<b>45</b>

# Índice de Figuras

---

3.1	Diagrama de conexión dos leds. . . . .	18
3.2	Diagrama de funcionamento do circuío de alimentación. . . . .	24
3.3	Diagrama do dispositivo. . . . .	26
3.4	Esquema completo do dispositivo. . . . .	27
3.5	Ultimo desño da carcasa. . . . .	28
4.1	Desño 3D do soporte para o dispositivo móbil. . . . .	35



# Índice de Táboas

---

2.1	Custos monetarios do proxecto . . . . .	8
2.2	Tempo empregado no proxecto . . . . .	8
5.1	Comparativa de consumo de amperios . . . . .	38



# Introdución

---

Nos últimos anos os medios de transporte alternativos como son o caso de bicicletas, patinetes e similares están aumentando notablemente. As preocupacións medioambientais, os beneficios para a saúde e as vantaxes en termos de custo e eficiencia están a impulsar estes vehículos unipersoais. Porén, a falta de costume dos condutores xunto coas substanciais diferencias entre os vehículos e a falta de proteccións en caso de accidente dificultan a convivencia con automóviles nas mesmas vías e implican un risco engadido para a integridade física dos condutores destes vehículos. Unha das maneiras de reducir estes riscos é aumentar a visibilidade tanto da bicicleta por parte dos coches coma viceversa co obxectivo de aumentar as distancias e os tempos de reacción dos condutores.

Varios estudos mostran que o uso de luces en bicicletas, especialmente as dinámicas permiten que estas sexan divisadas a maiores distancias tanto de día como de noite. A tese *The Nighttime Conspicuity Benefits of Static and Dynamic Bicycle Taillights* estuda os beneficios de diferentes luces traseiras de bicicleta de noite. Conclúe que as luces que se moven co ciclista, como as colocadas nos nocellos, son as que mais visibilidade aportan pero cando o ciclista deixa de pedalea estes beneficios se perden polo que o uso de unha luz fixa cun patrón de palpado pode ser a mellor opción en tódalas circunstancias.

Por outra parte a falta de retrovisores na maioría de bicicletas implica que o ciclista debe xirase cada vez que quere saber o que está a acontecer tras el facendo que así perda momentaneamente a visión do que ocorre diante e incluso o equilibrio en ciclistas non experimentados.

Para paliar estes problemas expónse unha solución baseada nun ou varios dispositivos dotados de luces e cámara xunto a outro de control e visualización.

## 1.1 Obxectivos

Os obxectivos principais de este proxecto serán dous: O primeiro de desenvolvemento de dous dispositivos diferenciados, un dotado de cámara e luces que se poderá colocar en

diferentes lugares da bicicleta e do que se poderán utilizar unha ou varias unidades o mesmo tempo que disporá de alimentación propia ou compartida. Un segundo dispositivo de interacción co usuario para o manexo do primeiro e a visualización do vídeo capturado. Tamén será necesario o hardware que permita a comunicación dos dispositivos xa sexa por cable ou sen fíos.

O segundo consistirá en desenvolver o software que permita o funcionamento dos dispositivos. O control das luces para indicar posición aumentar a visibilidade ou indicar manobras coma a freada ou xiro. O control do vídeo permitindo activalo e desactivalo cando se desexe. As posibles automatizacións como o acendido de luces cando haia pouca visibilidade ou a indicación automática da freada. As interfaces de interacción co usuario que permitan o control e a visualización sen distraccións da conducción. A integración e comunicación entre os dispositivos en tempo real e de forma transparente para o usuario.

Se comenzará coa análise das posíbeis solucións contemplando as diferentes opcións de hardware dispoñibles tendo en conta o custo, o tamaño, as capacidades de funcionamento, as restricións de compatibilidade, as restricións no software a utilizar, a dispoñibilidade e a dificultade de uso polo usuario final.

Se realizará a implementación da solución elixida e se someterá a probas nun entorno real para comprobar o cumprimento dos requisitos establecidos.

## **1.2 Metodoloxía**

A forma de traballo consistirá en seguir as directrices dadas polo método da enxeñaría partindo de estudo de traballos relacionados anteriores e continuando co análise, deseño implementación e avaliación do sistema implementado.

No apartado de deseño, debido a que o proxecto se estrutura en diferentes compoñentes, optarase por unha metodoloxía Top-Down que consiste en, partindo dos requisitos xerais, dividir o sistema en módulos independentes que se implementarán e probaran por separado. A cada un destes módulos se lle aplicara unha metodoloxía iterativa baseada en prototipos engadindo en cada incremento novas funcionalidades a o prototipo previamente desenvolvido, implementado e validado.

## **1.3 Proposta**

Neste traballo propoñerase unha solución baseada nun microordenador colocado baixo a sela, a Raspberry Pi Zero, que contará cunha cámara e unhas serie de luces leds conectadas. Para o control e a visualización do vídeo en directo realizarase unha aplicación android que se executara nun telefono situado no guiador da bicicleta.



## 1.4 Traballo relacionado

Existen no mercado dispositivos con funcións similares pero poucos integra tódalas funcións. É o caso da cámara Fly6 da compañía Cycliq, un combo de cámara traseira máis luces que se poden controlar dende o móbil, pero non permite o streaming en directo do vídeo, solo a gravación. Conta con características interesantes como a estabilización da imaxe, resistencia a auga e un tamaño moi compacto ou seu prezo e de 179 euros. Outra opción existente é a cámara Hexagon que se financiou mediante croudfunding no ano 2017 pero non chegou a produción. Este dispositivo si que contaba con streaming de vídeo en directo, xunto con acendido automático das luces en caso de freada, a aplicación tamén se encarga de gravar a posición gps no itinerario ou compartila en tempo real, conta cun segundo dispositivo con botóns para o control do dispositivo, e de chegar a producirse o seu prezo estaría entre os 100 e 200 euros.

Non te esquezas de incluír bibliografía e de referenciarla no texto. É moi importante (non abuses de páxinas web) !!

## 1.5 Estrutura da memoria

Este documento estrutúrase en seis capítulos e un anexo. Neste primeiro capítulo expónse a motivación os requisitos e as lineas xerais do proxecto. O capítulo 2 explora as posibles alternativas dispoñibles analizando as vantaxes e desvantaxes de cada unha delas fronte a solución elixida. O capítulo 3 relata o proceso e os detalles da implementación do dispositivo a construír xunto co seu software. O capítulo 4 expón a implementación da aplicación android que se utilizará para o control e a visualización. O capítulo 5 describe as probas realizadas e a análise dos seus resultados. O sexto e derradeiro capítulo recolle as conclusión obtidas trala realización deste proxecto.



# Analise e planificación

---

Neste capítulo partírase dos requisitos do proxecto para analizar as posibles funcionalidades planificar a implementación e elixir a solución a implantar.

## 2.1 Análise

O obxectivo do proxecto é a creación de dous dispositivos que realizarán as seguintes funcións:

- **Informar da posición da bicicleta mediante luces**

Esta función realizarase no dispositivo principal. Para elo terá que contar con luces leds e capacidade para controlalas, tamén dispor de capacidade de comunicación co segundo dispositivo e dispoñer dunha fonte de alimentación con capacidade suficiente para facer funcionar as luces.

- **Informar das manobras e estado do vehículo mediante luces**

Para realizar esta función os requisitos son os mesmos da función anterior o que engadiremos a necesidade de sensores para detectar cambios no movemento da bicicleta, para sinalizar freadas ou accidentes, e cambios na luz do ambiente para acender as luces cando as condicións lumínicas non sexan favorables.

- **Captura de vídeo do que sucede detrás do vehículo**

Esta función tamén se realizará no dispositivo principal e necesitará os requisitos de comunicación e alimentación enerxética xa citados. Contará cunha cámara para capturar as imaxes e necesitará a capacidade para procesalas e transmitilas en tempo real.

- **Entrada de ordes do usuario para o control de luces e vídeo**

Esta función se executa no segundo dispositivo, para realizala será necesario un mecanismo de entrada, como poden ser botóns, pulsadores ou pantalla táctil. Tamén se necesitará capacidades de procesamento, comunicación e alimentación enerxética.

- **Reproducción do vídeo en tempo real**

Tamén a realizar no dispositivo dous, esta función necesita, a maiores do anteriormente citado unha pantalla na que poder ver o vídeo, e capacidades abondo para reproducilo a tempo real.

A maiores destes requisitos funcionais contase cos seguintes obxectivos:

- **Pequeno tamaño e potabilidade**

Co motivo de poder dispoñer o dispositivo da bicicleta, xa sexa para cargalo ou por motivos de seguridade o deixar a bicicleta aparcada na rúa, priorizarase por un deseño portátil. O ideal é que os dispositivos poidan separase da bicicleta con facilidade e que o seu tamaño permita gardalos no peto. Este, xunto cos requisitos funcionais, é o principal motivo polo que optarase por utilizar un telefono móbil como segundo dispositivo, xa que a maioría de persoas dispoñen de un en todo momento, a así evitárase ter que levar un dispositivo a maiores.

- **Independencia entre dispositivos**

Buscarase que os dispositivos poidan conectarse sen fíos para obter unha maior independencia entre os dispositivos e evitar ter que colocar cables na bicicleta. Tamén se estudará a posibilidade de utilizar unha conexión cableada para diminuír o consumo enerxético dos dispositivos, neste caso optarase preferiblemente por unha conexión USB por compatibilidade co telefono móbil.

- **Batería e alimentación**

Para poder alimentar o dispositivo principal necesitarase dunha batería con capacidade abonda para poder utilizalo polo menos un día de uso sen ter que recargala. Tamén se estudará a posibilidade de incluír sistemas de aceso e apagado de ser preciso.

- **Sinxeleza e capacidade de actualización**

Pretendese desenvolver un sistema robusto e simple para facilitar o seu mantemento e poder actualizalo de forma sinxela. Especialmente o software ha de ser o mais simple posible para poder permitir incorporar novas funcionalidades no futuro.

## 2.2 Planificación

Seguindo a metodoloxía Top-Dow o desenvolvemento do proxecto dividirase en dúas fases correspondentes a os dous dispositivos. Comezase polo desenvolvemento do dispositivo principal seguido da aplicación no dispositivo móbil é a comunicación entre ambos. Unha vez desenvolvidas as funcións principais iterarase entre as fase para desenrolar a conexión entre os dispositivos.

### 2.2.1 Fase 1: Dispositivo principal

As tarefas a realizar son as seguintes:

- Conexión dos leds e probas de funcionamento.
- Funcións de control dos leds, secuencias e cores.
- Servidor de peticións de ordes.
- Conexión da cámara e probas de funcionamento.
- Transmisión de vídeo.
- Autoarranque e apagado.
- Alimentación enerxética e batería.
- Deseño e construción de carcasa e ancoraxes.

### 2.2.2 Fase 2: Aplicación do dispositivo móbil

Realizaranse a seguintes tarefas:

- Introducción de ordes.
- Deseño de interfaces.
- Transmisión de ordes.
- Xestión do estado.
- Probas con sensores.
- Funcións de automatización.
- Recepción de vídeo.
- Ancoraxe do dispositivo a bicicleta.

Compoñente	Cantidade	Custo por unidade	Subtotal
Raspberry Pi Zero W	2	11.00€	22.00€
Raspberry Pi Zero	2	5.00€	5.00€
Pi Camera Module V1	1	5.00€	5.00€
Pi Camera Module V2	2	20.00€	40.00€
Conxunto de lentes	1	2.00€	2.00€
Tira 8 leds w2812b	4	1.00€	4.00€
Anel 8 leds w2812b	2	1.00€	2.00€
Adafruit Powerboost 1000C	1	27.00€	27.00€
Batería Lipo 1600mA	1	7.00€	7.00€
Batería 18650 3400mA	2	2.00€	4.00€
Batería USB 5000mA	1	10.00€	10.00€
Material impresión 3D PLA	<1Kg	15.00€	15.00€
Cables			5.00€
Total			148.00€

Táboa 2.1: Custos monetarios do proxecto

Tarefa	Tempo
Documentación	80h
Desenvolvemento do dispositivo	100h
Desenvolvemento do software do dispositivo	100h
Desenvolvemento da aplicación	200h
Deseño das pezas 3D	70h
Probas e avaliación experimental	32h
Total	512h

Táboa 2.2: Tempo empregado no proxecto

## 2.3 Custos do proxecto

O software utilizado neste proxecto conta con licenzas de software libre polo que os custos restrinxíranse aos recursos humanos e recursos hardware. Nos recursos hardware da táboa 2.1 inclúense as pezas utilizadas finalmente para o dispositivo así como as que se utilizaron para realización de probas.

As estes custos engádense as ferramentas utilizadas incluíndo entre outros ordenador, impresora 3D, multímetro, soldador, dispositivo android e bicicleta.

Na táboa 2.2 móstranse os recursos humanos utilizados medidos en horas de traballo.

# Deseño e implementación do dispositivo

---

Neste capítulo afondarase no desenvolvemento do dispositivo comezando polo aspecto físico, opcións dispoñibles e alternativas, seguido pola construción do dispositivo e a implementación do software tendo en conta os problemas xurdidos e as solucións aplicadas.

## 3.1 Esquema xeral do dispositivo HW

O "dispositivo HW" debería ter un nome (estar bautizado) para que fose máis sinxelo falar del ...

As esixencias principais destes dispositivo son a existencia de conexións para as luces leds, conexións para unha cámara e capacidade de procesamento e comunicación o que engadiremos unha quinta esixencia, xa que o obxectivo do proxecto e conseguir unha solución de hardware e software libre que posibilite que o usuario final poida adquirir os compoñentes e montalos de forma sinxela, é preciso que os compoñentes sexan fáciles de adquirir e traballar con eles.

A continuación analizaranse as diferentes opción de hardware dispoñibles, as súas características e seus pros e contras para o proxecto proposto.

### 3.1.1 Placas de desenvolvemento

Faltan referencias !! para que o lector poida consultar toda a información. Aquí só debes poñer a relevante para o proxecto (evita poñer palla)

Sendo o desenvolvemento dunha placa dedicada a este propósito a solución idílica, o custo de este proceso xunto coa dificultade final da construción, sempre que non se optase por unha produción en serie, son as principais contras desta opción. Poren optárase por elixir unha placa cos requisitos requiridos entre as posibles opcións dispoñibles no mercado. Centrarémonos nos modelos co menor tamaño posible.

As placas contempladas son as seguintes:

- Arduino

Arduino é unha compañía dedicada o deseño e produción de placas de desenvolvemento con software e hardware, o que permite que terceiras compañías produzan as súas placas permitindo prezos finais de produto moi baixos.

Con un ide propio cunha linguaxe de programación baseada en C++, cunha ampla compatibilidade con diverso hardware, a diversidade de opcións e a súa facilidade de uso converte as súas placas nas mais populares do mercado.

Polo seu tamaño e forma as placas Arduino consideradas son as seguintes:

- Arduino Micro e Arduino Nano

O primeiro esta baseado no microcontrolador de 8bits ATmega32U4 cunha frecuencia de 16MHz, 32KB de memoria flash e 2.5KB de SRAM. Conta con 20 pins de entradas/saídas dixitais con 7 canles PWM e 12 pins de entradas analóxicas.

O segundo baséase no microcontrolador de 8bits ATmega328 traballando a unha frecuencia de 16MHz, conta con 32KB de memoria flash e 2KB de SRAM. Conta con 22 pins de entradas/saídas dixitais dos cales 6 son PWM tamén conta con 8 pins de entradas analóxicas.

Estas dúas versión o contar con gran cantidade de pins tanto dixitais como analóxicos e un consumo enerxético e moi baixo xunto cun baixo prezo inferior os 5 euros nas versións de fabricantes de terceiros fainos perfectos para o propósito de control de leds, pero a súa baixa potencia computacional dificultaría o procesamento de vídeo. Tampouco conta co hardware necesario para as comunicación sen fíos.

- Arduino MKR ZERO e Arduino MKR1000

Ambos baseado no procesador ARM M0+ de 32 bit de baixo consumo cunha frecuencia de funcionamento de 48MHz, 256KB de memoria flash e 32kB de SRAM, contan con 7 entradas analóxicas e 1 saída analóxica e 12 pins poden funcionar como PWM. Inclúe conexións SPI UART e I2C. Tamén inclúen unha conexión para alimentalos directamente cunha batería de 3.7v. A diferenza entre ambos e que o primeiro conta con 22 pins de entrada e saída dixital mentres que o segundo conta con 8 pero inclúe un chip Wi-Fi.

Ambos teñen as capacidades de procesamento necesarias para a xestión das luces, do vídeo e as conexións. O único punto negativo é que as cámaras compatibles a nivel de conexión e librerías con estas placas non dispoñen de moita calidade de vídeo.

Estas placas poden obterse por entre 20 e 50 euros



- Arduino MKR VIDOR 4000

Esta placa de desenvolvemento a parte do procesador ARM M0+ inclúe un chip *FPGA* que permite a súa configuración como diferentes hardware permitido que a placa poda dispoñer de diferentes compoñentes configurables como podería ser múltiples USB ou chips aceleradores de vídeo. Aparte conta con conexión micro HDMI mini PCI Express e un conector de cámara *MIPI* no que se poderían conectar diversas cámara con calidade mais que suficiente para este proxecto.

O seu prezo é superior os 60 euros.

- Raspberry Pi

As Raspberry Pi son unha serie de placas de desenvolvemento cun prezo moi axustado e unha potencia suficiente para para pode executar un sistema operativo completo. Grazas a súa popularidade dispón dun amplo soporte e compatibilidade con diversos software e outras plataformas hardware que a fan perfecta para diversos proxectos, como robótica, *Iot* ou centros multimedia. A versión dispoñible con menor tamaño e a Raspberry Pi Zero

- Raspberry Pi Zero e Raspberry PI Zero W

Esta placa conta cun microprocesador baseado na arquitectura ARM de 32bits que funciona a 1GHz, acompaña de un procesador de vídeo e unha memoria ram de 512MB. No apartado de conexións conta con un micro USB de carga e outro de datos, unha saída de vídeo HDMI e outra analóxica, unha ranura para unha tarxeta micro sd e un conector de cámara CSI. Tamen conta con 20 pins de conexión que a dotan de entradas e saídas dixitais, dúas canles *PWM*, conexións *SPI I2C* e *UART* xunto a conexións de 5v, 3.3v e terra. A versión Zero W tamén dispón dun chip Wi-Fi e Bluetooth. A súa potencia e capacidade de conexión a fan máis que capaz de para este proxecto, e o seu prezo, 5 e 10 euros respectivamente, é unha das súas principais vantaxes.

- ESP8266 e ESP32

A principal característica destas placas é que implementan chips Wi-Fi e Wi-Fi máis Bluetooth respectivamente,contan cun procesador *RISC* de un ou dous núcleos con velocidades dispoñibles entre os 80MHz e 240MHz e memorias ram de entre 32KiB e 520KiB.

Os seus múltiples portos e interfaces, *SPI*, *I2C*, *UART*, *PWM* entre outros, o seu baixo consumo e a súa compatibilidade co entorno de programación de arduino fainos ideais para pequenos proxectos de IoT, robótica ou domótica. Segundo as súas características poden obterse dende o prezo de un euro.

O igual que pasaba coas placas Arduino os ESP son ideais para a parte do manexo das luces pero non para a xestión do vídeo.

- Outras placas baseadas en procesadores ARM

NO mercado existen múltiples placas de desenvolvemento baseadas en procesadores ARM, non obstante o prezo e o soporte da Raspberry Pi faina a mellor opción para a maioría de proxectos mais xenéricos.

- Outras placas baseadas en *FPGA*

Os chip *FPGA* permiten un nivel de personalización hardware moi elevado, pero tamén contan cun alto prezo, e na maioría dos casos cun *toolchain* privativo que e necesario pagar para poder desenvolver en eles. O contrapunto a súa versatilidade é un maior custo de desenvolvemento en comparación con solucións de programación de alto nivel.

Tendo en conta o tamaño, a potencia, as conexións dispoñibles e o prezo, decidiuse optar pola Raspberry Pi Zero e Zero W como a placa encargada do control das luces e do vídeo.

### 3.1.2 Luces led

As dúas principais vantaxes das luces led fronte a outras formas de iluminación son o se baixo consumo e o seu pequeno tamaño. Estas calidades fainas ideais para un dispositivo portátil coma o que pretendemos construír. Os requirimentos principais son poder controlar a intensidade dos leds e dispoñer de polo menos un color vermello para indicar a posición e a freada, e un color amarelo ou ámbar para os intermitentes.

Coa intención de miniaturizar se optara por utilizar leds *RGB* que permiten xerar diversas combinacións de cores e así poder utilizar os mesmos leds para as diferentes funcións. Xa que a Raspberry Pi non conta con saídas analóxicas, utilizaranse as canles *PWM* que permite enviar sinais moduladas en pulsos. A modulación *PWM* permite acender e apagar os leds múltiples veces a unha alta frecuencia a unha velocidade, tan rápidas que o ollo humano percibe como diferentes intensidades lumínicas en función da anchura dos pulsos. En leds *RGB* compatibles o *PWM* tamén se pode utilizar para codificar a cor elixida, e en series de leds conectados e direccionables se pode elixir que led a iluminar e a súa cor e intensidade individualmente, permitindo así controlar un alto numero de leds cunha soa saída *PWM*.

Existen diferentes tipos de leds *RGB* direccionables no mercado, elixiremos o tipo de led en función do seu tipo de conexión, a dispoñibilidade de librerías de software compatibles e coa limitación de que deberán operar a 5v que a voltaxe constante que necesita a Raspberry Pi para funcionar.

Os tipos de led direccionables analizados son:

- WS2812B e WS2813

Estes leds inclúen un circuíto integrado en cada led que o conectalos en serie permite o control dunha secuencia teoricamente infinita de leds. Cada led conta con tres entradas e tres saídas: voltaxe, terra e datos. A información a pasar os leds se formara cun fluxo de datos a almacenar nun *buffer* en memoria, ocupando a información de cada un 3 bytes, e se pasarán o primeiro led que lerá os primeiros 24 bits coa información da intensidade de cada cor, vermella, verde e azul, e pasará o resto de datos o seguinte led.

Cada led tarda 30 microsegundos en recibir os datos e 50 microsegundos e actualizar a súa cor, o atraso de transmisión entre leds e de 0.5 microsegundos. O consumo máximo de cada led e de 60 mA a 5V.

Os WS2813 engaden unha segunda liña de datos para que se un led deixa de funcionar os seguintes poidan seguir recibindo a información.

- SK6812

Estes leds comparten a maioría de características dos WS2812B coa diferenza de que aumenta a súa taxa de refresco a 1.2KHz con respecto os 400Hz dos WS2812B. Tamén engaden unha cuarta cor branca en cada led.

O a nivel de software son compatibles con WS2812B pero as súas diferenzas non permiten a súa interconexión física.

- APA102 e APA102C

Estes leds contan cunha interface SPI que conta cunha sinal de datos e outra de reloxo. Isto e para solucionar o problema de sincronización que se poden producir cando os led son manexados dende placas con capacidade de multitarefa sen un *kernel* específico para entrada e saída a tempo real. Tamén aumenta súa taxa de refresco ata os 19.2kHz.

Sendo os leds APA102 superiores en características os outros dous, contan coa desvantaxe de que requiren máis cables de conexión. As súas vantaxes a nivel de velocidade e sincronismo son esenciais para a xeración de imaxes ou vídeo pero non para simples animacións como as que utilizaremos neste proxecto. Porén optaremos por utilizar os leds WS2812B e WS2813 ou SK6812.

Este tipo de leds están dispoñibles en diferentes combinacións: leds individuais, tiras flexibles, tiras ríxidas, aneis e matrices. Faremos probas con tiras e aneis de diferentes tamaños.

### 3.1.3 Cámara

No caso da cámara plantexanse dúas opcións utilizar unha cámara usb ou unha das cámaras deseñadas para funcionar coa Raspberry Pi que utilizan a súa conexión CSI. Optaremos pola

segunda opción xa que unha cámara usb implica un tamaño demasiado grande para o proxecto e ademais non soen estar indicadas para a iluminación de exteriores.

No mercado existen diversos módulos de cámara para a Raspberry Pi pero a maioría están baseados nos Raspberry Pi Camera Module V1 e Raspberry Pi Camera Module V2 a principal diferenza entre ambos e que o primeiro conta con 5 megapixels mentres o segundo conta con 8 megapixels e unha notable mellora na calidade de imaxe.

As principais diferenzas nos módulos dispoñibles no mercado son:

- Tamaño

Existen versións específicas para a Raspberry Pi Zero máis pequenas pero solo do Camera Module V1, as versións normais xa contan cun tamaño moi axustado.

- Presenza de filtro infravermello

As cámaras soen contar cun filtro de luz infravermella para evitar o *aliasing* que se produce nas cámaras xa que as pantallas que utilizamos non están destinadas para emitir infravermellos e os nosos ollos non son capaces de percibilos. As cámaras que non contan con este filtro dan como resultado imaxes máis luminosas e cunha tonalidade violácea. Estas cámaras son útiles para entornas exteriores no solpor ou para visión nocturna se se conta con fontes de luz infravermellas.

- Tipo de lente

A uso dunha cámara cunha lente curva permite ampliar o campo de visión da cámara, se a lente é demasiado curva se producirá unha distorsión da imaxe nos bordes.

Xa que estas cámaras son todas compatíbeis a nivel de software probaremos diversos tipos con varios tipos de lente xa sexan incorporados ou engadindo unha lente externa.

### 3.1.4 Alimentación e baterías

A Raspberry Pi Zero necesita unha fonte de alimentación que provea de 5V constantes. O seu consumo enerxético varía segundo a carga computacional, o uso do Wi-Fi e o uso da cámara podendo ascender a entornon 300mA. Os leds tamén funcionarán a 5V cun consumo máximo de 60mA por led, cando emiten luz branca a máxima intensidade.

Plantexamos dúas solucións posibles:

- Bateria externa USB

Utilizar unha batería usb externa permite dispoñer de altas capacidades que prolongarían o tempo de uso pero implican o uso dun dispositivo a maiores. Outra de desvantaxes e que cando se esgote a batería a corrente interrompese de golpe sen que a Raspberry

poida realizar un apagado normal, como consecuencia tras varios apagados podería danarse o sistema de ficheiros se se estaba a escribir nel no momento do apagado.

- Batería, circuío de alimentación e circuío de acendido e apagado.

A maioría de baterías usadas en electrónica son baterías de *ions de litio* principalmente debido as sua alta capacidade enerxética e lonxevidade. A *voltaxe nominal* destas baterías é de 3.7V sendo 4.2V a voltaxe coa carga o máximo e por debaixo de 3V deixan de proporcionar suficiente intensidade eléctrica para a maioría de aplicacións. Este tipo de baterías son as que atoparemos nos teléfonos móbiles, ordenadores portátiles e incluso en vehículos eléctricos. Poden colocarse en serie cando é necesario unha maior voltaxe ou en paralelo cando o que se necesita é unha maior capacidade.

Para poder utilizar estas baterías é necesario un circuío de carga, un de protección, e un de conversión de voltaxe. Na maioría dos casos as baterías de consumo utilizan o estándar USB, que funciona a 5V, tanto para cargarse como para proporcionar enerxía. Polo que será necesario un chip de carga que acepte unha toma de 5V e que cargue a batería ata 4.2V. As baterías de litio poden ser perigosas danándose e chegando incluso a estoupar se se descargan demasiado ou se se sobrecargan polo que é necesario un circuío de protección que evite a sobrecarga e a sobrecarga. Para proporcionar unha saída estable de 5V tamén é necesario un conversor de voltaxe. É habitual atopar o circuíos de carga máis protección xuntos no mesmo chip aínda que tamén se atopan circuíos que integran as tres funcionalidades.

### 3.1.5 Software a utilizar

A Raspberry Pi conta cunha ampla gama de sistemas operativos, algúns con propósitos concretos como reprodución de multimedia, servidores locais ou nodos de rede. Tamén conta con versións das distribucións Linux máis popular como Ubuntu, Arch ou Kali entre outros. Raspbian é unha distribución baseada en Debian, é máis antiga e máis optimizada para a Raspberry Pi e a que dispón de máis soporte polo que será a elixida para o proxecto.

Para o control dos leds existen varias librarías dispoñibles para varias linguaxes de programación. As máis utilizadas son a de *Adafruit*, que só está dispoñible en Python, e a de Jeremy Garff que será a que utilizaremos, xa que conta con unha documentación detallada e esta dispoñible en varias linguaxes de programación, entre outras C, Python e Java.

Para a captura e transmisión do vídeo contamos con varias alternativas. A librería *picamera* para Python permite configurar calquera parámetro e o *streaming* do vídeo na rede. Por outra parte o software para captura de vídeo *raspivideo*, escrito en Python, tamén permite moitos parámetros de configuración e a sua saída de vídeo pode enviarse a rede utilizando algún programa para redireccionar o *bitstream* dos datos como pode ser o software *socat*.

A comunicación entre o dispositivo de control e o de luces e captura de vídeo realizarase a través dunha conexión IP. Para manexar as peticións podemos empregar unha das clases servidor Python, unha librería de terceiros ou implementar o servidor a nivel de *sockets*.

Por motivos de compatibilidade entre todo o software necesitado para o control de leds, vídeo, e conexións, decidiremos integralo todo nunha aplicación Python que se encargará das tres tarefas.

### 3.1.6 Caixa e ancoraxes a bicicleta

**incluir esquemas e figuras para ilustrar conceptos**

O lugar a colocar o dispositivo de iluminación e captura será na barra da sela da bicicleta, esta posición é a ideal tanto para capturar o vídeo como para que as luces sexan vistas polo tráfico que circula detrás do vehículo.

A Raspberry Pi conta cunha caixa oficial na que se pode instalar xunto coa cámara V2. Esta é a que utilizaremos no caso de alimentala cunha batería externa. Para suxeitar a placa á barra deseñaremos un soporte e o imprimiremos cunha impresora 3D. Para a versión con batería interna deseñaremos unha caixa protectora que albergue tódolos compoñentes e que se poida suxeitar á barra.

Os prototipos imprimíranse en *PLA* un material biodegradable e de fácil impresión, no é o mellor material para resistir a auga ou a humidade pero é ideal para o prototipado. Poderá utilizarse *ABS* para imprimir unha versión final, un material moito mais resistente as condicións atmosféricas.

## 3.2 Leds

Crearanse dous prototipos con dúas configuracións de leds diferentes, ambas contarán cun anel de 8 leds *RGB* direccionables WS2812B, que conta cun tamaño perfecto para colocar arredor da cámara, a segunda opción contara ademais con dúas tiras de 8 leds cada unha que se utilizaran como indicadores de xiro para aumentar a visibilidade.

### 3.2.1 Conexión coa Raspberry Pi

Estes leds contan con catro puntos de conexión entrada de voltaxe, terra, entrada de datos e saída de datos.

A voltaxe necesaria para alimentalos é de 5V, aínda que na maioría dos casos o fabricante indica un soporte a voltaxes de entrada de entre 4V e 7V. A Raspberry Pi conta con pins de saída a 5V conectada directamente a entrada, sen contar coa limitación dun fusible como noutras versións da placa, polo que pode alimentar os leds directamente pero xa que cada led pode chegar a consumir ata 60mA e facer pasar polo power rail unha corrente excesiva podería

probocar danos ou unha aumento da temperatura da placa implicando menor velocidade e maior consumo. Tendo en conta que o fabricante non recomenda que a placa consuma mais de 1A será convinte alimentar os leds directamente dende a fonte de alimentación, especialmente na versión na que utilizaremos 24 leds.

Para a conexión de datos teremos que usar unha das saídas da placa conectadas a un das dúas canles *PWM* da que dispón. Estas saídas lóxicas contan cunha voltaxe de 3.3V, pero as tiras leds requiren que a voltaxe na entrada de datos, para ser interpretada como un valor lóxico HIGH, sexa alomenos un 70% da voltaxe de alimentación, neste caso 3.5V. Para solucionalo contamos con dúas opcións: Reducir a voltaxe de entrada dos leds, o que implicaría unha menor luminosidade ou aumentar a voltaxe do valor lóxico que é a solución pola que optaremos utilizando un conversor lóxico de nivel. Na práctica comprobaremos que os leds utilizados seguen interpretando como valor lóxico positivo os 3.3V polo que a utilización ou non do conversor de nivel a valoraremos máis adiante en función do espazo dispoñible.

Os pins da placa dispoñibles con conexión *PWM* serán o GPIO18 e GPIO12 para a canle PWM0 e GPIO13 para a canle PWM1. A librería que utilizaremos tamén permite controlar o led mediante a conexión SPI e a PCM, utilizaremos a *PWM* por que é a única que nos permite controlar dúas tiras led independentes simultaneamente, coa contraindicación de que o utilizar o *PWM* a Raspberry non poderá xenerar audio analoxico.

Utilizaremos o GPIO18 para controlar o anel led e o GPIO13 no caso que utilizemos os intermitentes que irán conectados un o outro como se indica na figura 3.1.

### 3.2.2 Software de Control

Para manexar os leds utilizaremos a librería `rpi_ws281x` de Jeremy Garff na súa versión para Python. O seu funcionamento é moi simple, primeiro teremos que configurar os parámetros da tira led, coma o número de leds, o pin a que esta conectado, o tipo de tira ou a canle *PWM* entre outros. No programa principal deberemos inicializar os leds con estes parámetros e executala coa función `begin`. Cada vez que queramos que os leds cambien o seu estado chamaremos a función `show`.

Escribiranse funcións encargadas dos padróns de iluminación. Estes padróns serán os seguintes:

- Luz vermella fixa, é a encargada de indicar a posición da bicicleta.
- Luz vermella intermitente, a súa función é a mesma que a anterior, pero o padrón de palpadeo aumentará a visibilidade. Crearanse distintos padróns combinando distintas frecuencias en intensidades luminicas.

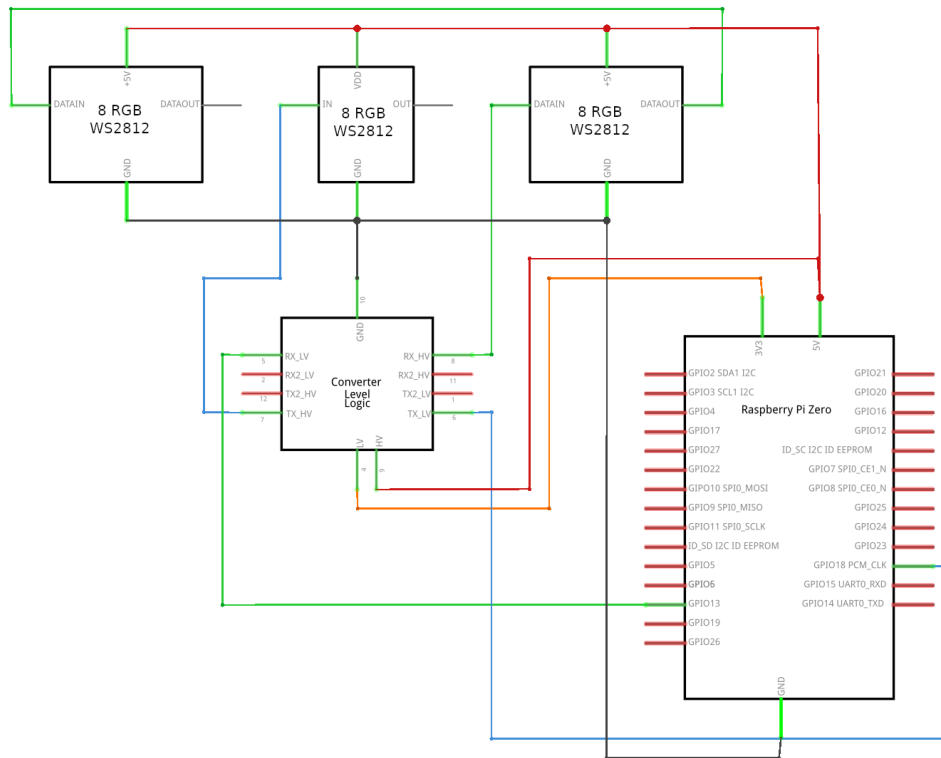


Figura 3.1: Diagrama de conexión dos leds.

- Luz vermella incremental, é a encargade indicar a freada, a súa intensidade aumentará ata o valor máximo para emular as luces de freada dos coches.
- Luz amarela de xiro a esquerda ou dereita, indica o xiro iluminando progresivamente os leds do anel dende os situados no centro ata os do extremo esquerdo ou dereito, de dispoñer das tiras extra de leds de xiro estas iluminaranse a continuación. Unha vez iluminados todos os leds estes se apagarán e o padrón repetirase de novo.

Tamén se escribirá unha función para controlar a intensidade dos leds en función dun valor numérico recibido, 0 será mínimo e 100 a máxima intensidade.

### 3.3 Camara

A camara a utilizar é a Raspberry Pi Camera, porbarase a version 1 e a versión 2, ambas conectanase a raspberry Pi Zero co mesmo cable, o conector da placa é delicado polo que deberase conectar con coidado. Para habilitala executaráse o comando raspiconfig na terminal e no apartado de interfaces activarase a opción camara.



A versión 1 da cámara conta cunha resolución de 5 megapíxeis mentres que a versión 2. As seguintes imaxes ilustran a diferenza de calidade de imaxe entre as dúas versións.

### 3.3.1 Lentes

Para poder capturar o completo da estrada a cámara necesitará de algun tipo de lente que permita un maior campo de visión. Existen versións da cámara que xa inclúen unha lente, pero tamén poderemos atopar lentes externas coma as destinadas para os dispositivos móbiles, que contan co tamaño necesario para a cámara da Raspberry. Nas seguintes fotos ilustramos as diferenzas segundo o tipo de lente.

### 3.3.2 Captura de vídeo

Raspivideo é o software que utilizaremos para capturar as imaxes, o programa executase dende terminal proporcionandolle diferentes parámetros. No noso caso os parámetros a utilizar serán:

- -t Tempo de captura de vídeo, no noso caso será 0 indicando que a captura será continua.
- -w e -h Son os parámetros de anchura e altura de píxeles, probaremos diferentes resolucións para conseguir a máxima calidade posible sempre que o tamaño da imaxe non repercuta na latencia de transmisión.
- -fps Frames per second, é o número de imaxes a capturar cada segundo, variaremos con este valor para minimizar a latencia.
- -b Bitrate, o número de bits por segundo, buscaremos o valor máis alto posible sen que produza retardos na transmisión.
- -n Con este parámetro desabilitaremos a previsualización do vídeo.
- -pf Parámetro para elixir o perfil do codificador de vídeo H264, as opcións disponibles son, baseline, main and high. Utilizaremos a opción baseline xa que é a que menor coste computacional ten.
- -o Con este parámetro indicamos a saída de vídeo, como por exemplo a un arquivo, no noso caso utilizaremos a saída estándar que indicaremos con "-", e que redirecionaremos máis tarde.

### 3.4 Transmisión de video

Para transmitir o video ao dispositivo móbil a traves da rede preséntanse varias posibilidades. Para comparalas transmitiremos video dende a raspberry pi cunha mesma resolución, 720p, e recibiremos e reproduciremos o nun pc mediante vlc.

- A primeira opción a analizar é o software de video vlc, unha completa ferramenta de reprodución que tamén permite a transmisión e a recepción de video na rede mediante diferetes protocolos. Faremos unha proba utilizando o vlc na raspberry pi para transmitir o video da camara e recibilo nun pc con vlc. Como resultado obtemos unha transmisión cunha latencia superior a un segundo, que imposibilita o seu uso para controlar o trafico en tempo real.
- A segunda opción que probaremos consistirá en capturar o video coa ferramenta de captura de video da raspberry pi, raspivid, e redireccionar a sua saída a rede utilizando netcat unha utilidade para transmitir e recibir na rede mediante tcp ou udp. Transmitiremos mediante udp para conseguir unha menor latencia a custo de perder algun fotograma. A recepción de video a realizaremos nun pc mediante vlc coma no caso anterior. A latencia obtida neste caso e mellor que no anterior.
- Buscando reducir ainda mais a latencia probaremos a utilizar o software socat, que funciona de forma similar a netcat e conta tamen con moitas opcións de configuración. O procedemento será igual que no caso anterior, faremos a captura con raspivid e redireccionaremos o video a un porto nunha dirección ip mediante udp neste caso utilizando socat. Como resultado obtemos unha latencia ainda menor que con netcat polo que utilizaremos este software para a transmisión de video.

### 3.5 Recepción de ordes

Para recibir os comandos enviados dende o dispositivo mobil e executalos mediante python implementaremos un servidor tamén python para poder integrar a recepción e a execución de ordes no mesmo programa.

A primeira opción será utilizar peticións http e manexalas mediante a clase de python BaseHTTPRequestHandler, o problema é que esta clase bloquea o programa mentres se executan as ordes correspondentes a petición recibida. Para solucionar este problema poderíase implementar un servidor multithread ou utilizar unha librería para python que implemente o servidor multithread de forma transparente. Plantexase utilizar as librerías multithread Tornado, Twisted ou lighttpd pero debido os recursos limitados da raspberry pi zero o uso dunha destas librerías podería implicar maiores latencias e consumo enerxético.

Finalmente optase por mañexar a conexión directamente mediante sockets non bloqueantes é así poder utilizar un solo thread. Para elo introduciremos as peticións de conexión nunha lista, unha vez aceptada introducirase nunha segunda lista as mensaxes recibidas e se executara a orde correspondente para cada mensaxe.

As mensaxes seran as seguintes:

- **r** para o xiro a dereita
- **l** para o xiro a esquerda
- **n** para a luz de noite
- **b** para a luz de freo
- **k** para o padrón de palpadeo
- **o** para apagar as luces
- **v** para iniciar a captura e transmisión de video
- **vs** para deter a captura e a transmisión de video
- **c** para comprobar que a conexión segue aberta
- **valor numérico** para establecer a intensidade das luces

### 3.6 Conexión co servidor

Os dispositivos conectaranse mediante unha rede local, xa sexa a traves dun cable usb ou dunha rede wi-fi, en ambos os casos o dispositivo android será o encargado de aloxar a red xa sexa compartindo por usb ou creando un punto de acceso wi-fi. cc

Para permitir que o dispositivo móbil se conecte o servidor sen ter que coñecer a dirección ip de este procederase da seguinte maneira. Dende o servidor creárase un thread no que se abra un socket encargado de enviar unha mensaxe a dirección de broadcast para que poda ser recibido por todos os dispositivos da rede. O dispositivo móbil

### 3.7 Autoarranque do servidor

O servidor deberá arrancar automaticamente o encender o dispositivo, e arrancar de novo se por algun motivo detense a súa execución. Para elo crearemos un servizo en systemd que se encargará de arrancar o programa e reinicialo se é necesario. A estrutura do servizo e simple, indica a localización do programa python a executar, a orde de reiniciar sempre, os

logs a utilizar para rexistrar as execucións e os fallos e o usuario e grupo a no que se executará. Ubicaremos o ficheiro do servizo na ruta `/etc/systemd/system/` e unha vez ali o habilitaremos coa orde `sudo systemctl enable bikeview.service` agora o servizo executarase cando se arranque o dispositivo e reiniciarse se se para.

### 3.8 Alimentación e enerxía

O consumo de amperios da raspberry pi zero sen carga de traballo é duns 120 mA, grabando video a 1080p o consumo é de 230mA, nos grabaremos video a 720p polo que o consumo sera algo menor pero engadirase o consumo do chip wifi funcionando. Os leds ws2812 teñen un cosumo máximo de 60 mA cada un 20 mA como máximo por cada un dos tres leds RGB a máxima intesidade, o noso máximo consumo realizarse coa luz vermella acesa de forma continua, xa que no resto de modos os patrdrons de palpadeo reducen o cosumo. Contamos con 24 destes leds polo que o consumo máximo será de 20mA por 24 leds, un total de 480mA que sumados o consumo da raspberry pi nos da un consumo máximo teorico de 710mA na versión sen leds intermitentes o consumo seria de 160mA máis 230mA, en total 390mA.

- Unha primeira versión máis sinxela contará solo cunha batería usb para alimentar a raspberry. Os requisitos de esta batería seran a amperaxe e a capacidade.

Partirase do valor do cosumo maximo aproximado de 400mA para calcular o tempo de funcionamento. Con esta amperaxe aos 5V que funcionan a raspberry e os leds a pontencia utilizada sería de 2W. Neste suposto unha batería de 5000 mAh cunha voltaxe nominal de 3.7V pode proporcionar 18.5Wh polo que duraría ata 9 horas e 15 min, no caso dunha batería de 1000 mAh o tempo mínimo teórico de funcionamento sería de algo menos de duas horas. Comprobaremos se estes supostos se cumpren facendo medicións do tempo de funcionamento.

- Realizaremos unha segunda versión máis avanzada que apagará o dispositivo cando a batería baixe de certo umbral de voltaxe para evitar que o dispositivo se desconecte e contará tamén cun pulsador para poder encendela e apagala.

Para isto utilizaremos o chip de carga Adafruit Powerboost 1000 que conta cunhas características moi interesantes a maiores do abitual protección de sobrecarga e sobredescarga conta cunha led e un pin que se activaran cando a voltaxe da batería baixe dos 3.2v, unha voltaxe operacional de 5.2v para evitar perdidas de voltaxe en cables e conectores, un pin abilitador que permite conectar ou desconectar a batería, e proporciona 1 amperio de intensidade sen baixar a voltaxe dos 5v.

Para realizar o circuito basearemonos no proxecto lipopi de Daniel Bull que utiliza o adafruit powerboost, nas suas duas versións a de 500 mA e 1 A para programar o apagado

automatico da raspberry pi cando a batería baixe de 3.2v e un pulsador para o encendido, tamén conta con dúas versións mais unha que tamén permite o apagado, e outra que monitoriza a voltaxe da batería. Realizarase a versión con pulsador para encendido e apagado.

O funcionamento é o seguinte, o presionar o pulsador conectase o positivo da batería co pin abilitador, acendendo o adafruit powerboost e por conseguinte acendendo a raspberry pi. O acender a raspberry pi un pin conectado o pin abilitador acenderase para seguir mantendo un valor positivo. Para evitar que o voltaxe no pin abilitador caia entre que pulsamos o pulsador e a voltaxe do pin da raspberry se encargue de manter o valor positivo situaremos un circuíto RC formado por un condensador cunha resistencia en paralelo entre o pulsador e o pin abilitador. O condensador cargarase cando o pulsador cerre o circuíto e descargarase a continuación mantendo a voltaxe o tempo suficiente para que a raspberry arranque e acenda o pin. Utilizarase un condensador de  $100\mu\text{F}$  xunto cunha resistencia  $100\text{k}\Omega$  que proporcionan un tempo suficiente de 10 segundos. O pin da raspberry que utilizaremos para este propósito pode ser o 14 correspondente a conexión uart, que se acenderá coa raspberry e se desconectará cando se apague, engadiremos unha resistencia de  $10\text{k}\Omega$  para protexer este pin. Tamén poderíamos utilizar calquer outro pin de propósito xeral indicando no arquivo de configuración config.txt na partición boot da raspberry, que o pin arranque cun valor positivo e cun valor negativo cando o dispositivo se apague.

Para o apagado utilizarase un segundo pin conectado o pulsador, cando este se pulse, estando a raspberry acesa, se conectará a voltaxe da batería, cando este valor positivo chegue o pin un script python encargarse de apagar o dispositivo. Engadiranse un divisor de voltaxe para reducir a voltaxe da batería que cando está completamente cargada a súa voltaxe de 4.2v e superior o máximo valor lóxico tolerado pola raspberry pi de 3.3v. Utilizaremos unha resistencia de  $33\text{k}\Omega$  conectada entre a o pin e a batería e unha resistencia de  $100\text{k}\Omega$  entre o pin e terra. Para evitar que o pin de encendido dispare o apagado situarase un diodo entre o pin de encendido e apagado evitando que a voltaxe circule nesa dirección.

Finalmente conectarase o pin indicador de batería baixa a outro pin de entrada da raspberry que mediante o script python apagará o dispositivo. O diagrama de funcionamento mostrase na figura 3.2.

Na figura 3.3 podemos ver o diagrama final do dispositivo e na figura 3.4 o esquema completo integrando os leds o circuíto de carga e a cámara.

Para arrancar automaticamente o script Python instálase un novo servizo en *systemd* e de igual maneira que co servidor executase no arranque e cada vez que se pare.

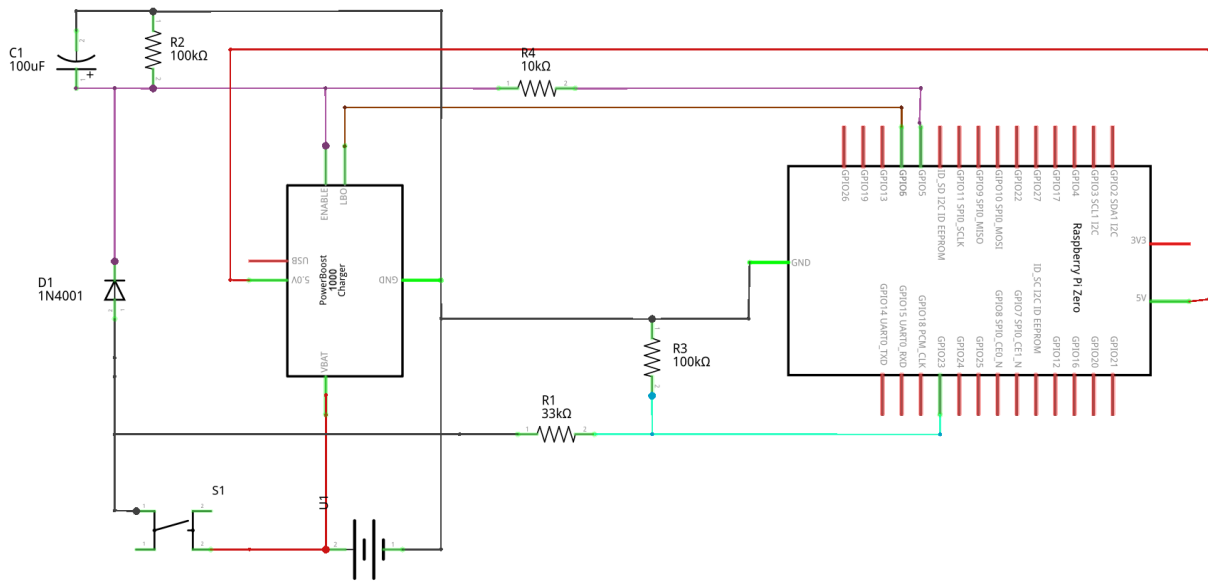


Figura 3.2: Diagrama de funcionamento do circuito de alimentación.

Na implementación física colocaremos o chip de carga xunto cos compoñentes electrónicos nunha placa cun conector para poder conectalo directamente os conectores da Raspberry Pi Zero. Tamén colocarase nesta placa o conversor lóxico de nivel como se ve na figura incluír foto circuito

Na elección da batería teremos en conta a maiores da súa capacidade o seu tamaño e forma, sendo o ideal que sexa similar o da raspberry para poder integrala no dispositivo con facilidade. Por eso elixiremos unha batería de 1600mAh e 5.92Wh cunhas dimensións de 9 x 34 x 50mm que para este suposto cun consumo máximo de 3.55W debería proporcionar un tempo mínimo de funcionamento de 2 horas aproximadamente.

### 3.9 Carcasa e anclaxe

Para protexer o dispositivo e suxietalo baixo a sela da bicicleta plantexaranse dúas opcións.

- A primeira realizarase para a versión do proxecto alimentada cunha batería usb. Cosistirá en utilizar a carcasa oficial da raspberry pi zero que inclúe un oco para a cámara e espazo para a conexión na parte de atrás, a carcasa só permite a uso de cámaras sen lentes polo que incorporárase unha lente externa.

Para suxeitar a carcasa a bicicleta deseñaremos un soporte en 3d co software Blender. Partiremos das medicións da carcasa e deseñárase un soporte que suxeite a carcasa

firmemente e permita atala a barra da sela mediante unha correa.

Unha vez deseñado e tras comprobar que o deseño e imprimible o exportarase no formato STL que abriremos cun software encargado de dividir o deseño en capas e traducilo a ordes de desplazamento interpretables pola impresora, como resultado obterase un arquivo GCODE que pasaremos a impresora. O prototipo imprimírase o prototipo en 3d probarase e aplicaranse correccións no modelo. Para este deseño realizáronse tres iteracións ata conseguir o resultado desexado.

Primeira iteración Segunda iteración Terceira iteración

- A segunda versión terá que albergar a raspberry pi zero xunto coa cámara, o chip de carga e alimentación, o conversor lóxico de voltaxe e a batería. Utilizaremos tamén neste caso o software de edición 3d Blender para deseñar os prototipos.

O deseño contará co anel led situado no exterior o redor da lente da cámara, no interior colocaranse tódalas compoñentes electrónicas e a batería. Na parte superior contará cun oco para o conector micro usb de carga e acceso a tarxeta micro sd da raspberry pi. No exterior contará tamén con dous brazos articulados nos que situaremos as dúas tiras leds para indicar o xiro.

Da mesma forma que no caso anterior o deseño imprimírase modificárase e volverase a imprimir ata que se obteña un resultado aceptable.

Os deseños son os seguintes:

Primeiras iteracións

Deseño final

### **3.10 Custo do dispositivo**

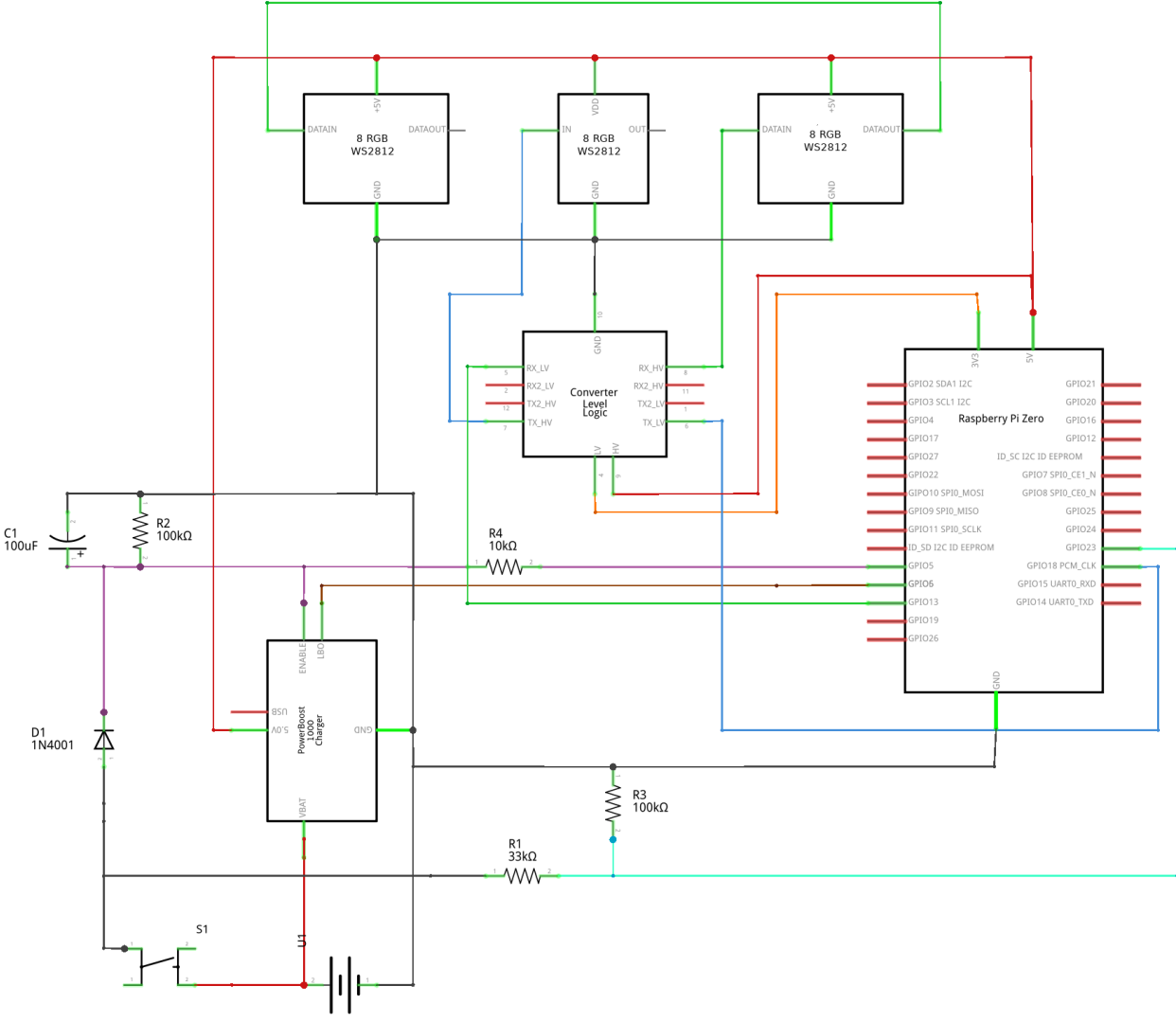


Figura 3.3: Diagrama do dispositivo.



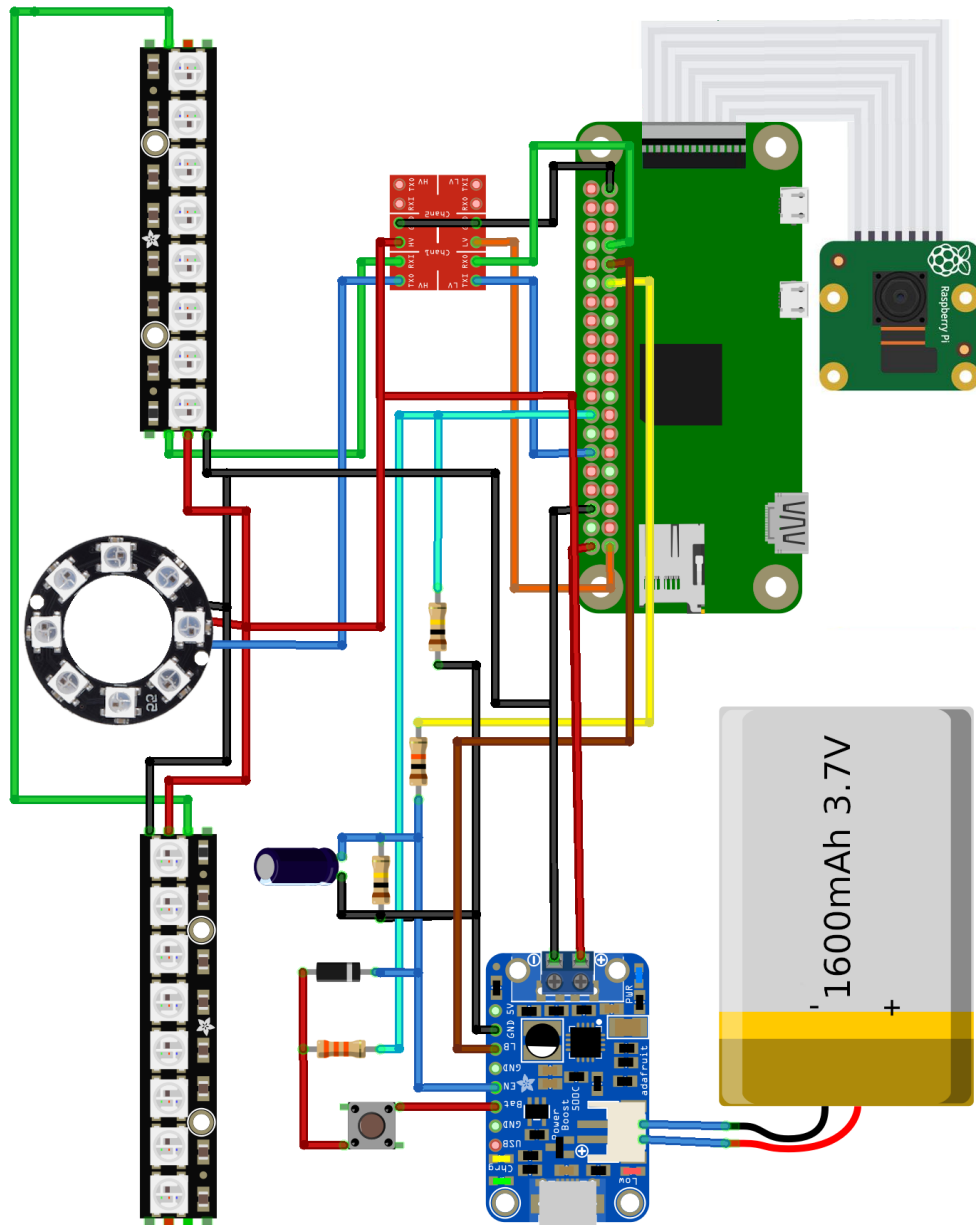


Figura 3.4: Esquema completo do dispositivo.

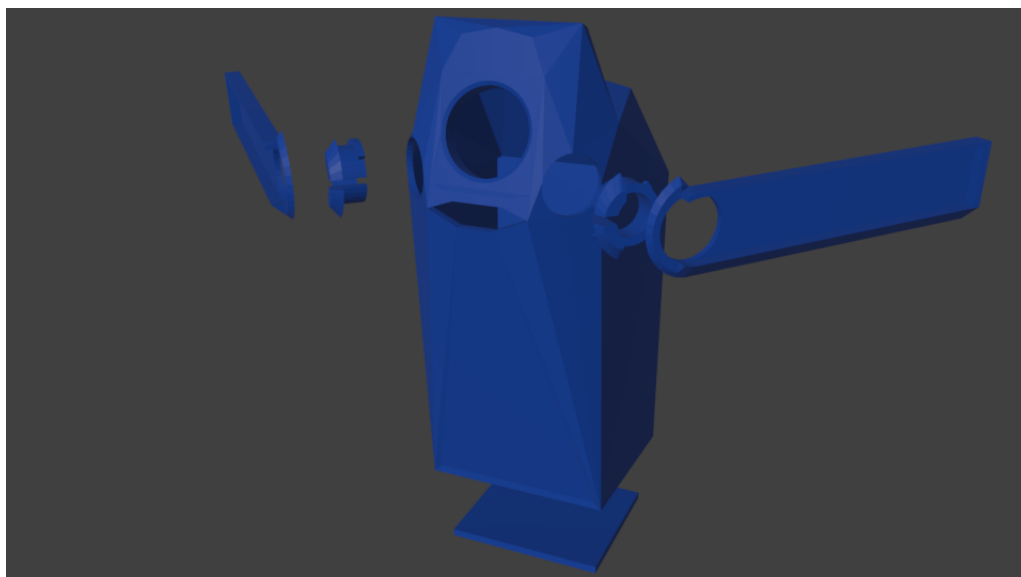


Figura 3.5: Ultimo desenho da carcasa.

# Arquitectura e implementación da aplicación

---

Detalles da implementación do software, problemas xurdidos e solucións aplicadas posibles  
subseccion: servidor python- control das luces, control da cámara, comunicación, sockets e  
broadcasting aplicación android- layouts e botóns, comunicacións ordes e recepción de vídeo,  
xestión de varios dispositivos

Estou algo preocupado, porque agora mesmo o SW é moi sinxelo (literalmente 4 páxinas ...)

Son moitas as posibilidades de implementación deste dispositivo, os requisitos principais son:

- Dispoñer dunha pantalla para visualizar o vídeo e o estado das luces.
- Contar con algunha interface de entrada de datos como botóns ou pantalla táctil.
- Dispoñer dun hardware para comunicarse co dispositivo principal. Como por exemplo: Wi-Fi, Bluetooth ou USB.
- Contar cunha batería ou fonte de alimentación

Realizar unha implementación física do dispositivo contaría coas vantaxes de poder contar cunha alta personalización dos seus compoñentes e robustez ao contar cun único dispositivo obxectivo. Sen embargo se nos presenta outra opción moito máis atractiva: utilizar un teléfono móbil e crear unha aplicación dende onde poder visualizar o vídeo e controlar as luces, aparte de aforrar a construción do dispositivo. Reduciremos así o número de compoñentes que o usuario ten que levar, xa que é habitual dispoñer dun móbil en todo momento.

Desarrollaremos a aplicación para o sistema operativo Android xa que este é o sistema operativo máis empregado globalmente e nos permitirá chegar a un maior número de usuarios.

Para suxeitar un teléfono móbil ao manillar da bicicleta existen múltiples anclaxes que se adaptan a varios tamaños de teléfono. Tamén existen plantillas configurables segundo o tamaño do teléfono que despois se poden imprimir en 3D.

## 4.1 Esquema xeral da aplicación

Para crear esta aplicación android optárase por utilizar a linguaxe de programación Kotlin e a o entorno de programación Android Studio. Kotlin é unha linguaxe de programación creada por JetBrains cun obxectivo inicial de executarse na máquina virtual de Java, dende 2017 Kotlin é unha linguaxe oficial para desenvolver aplicacións android.

Algunhas das súas vantaxes fronte a Java son:

- Unha maior expresividade: Podes escribir máis con menos código.
- Maior seguridade: En kotlin é obrigatorio especificar a nulabilidade dos obxectos e esta se comproba en tempo de compilación.
- É funcional: Kotlin é unha linguaxe orientada a obxectos pero inclúe conceptos da programación funcional como as expresións lambda.
- Fai uso de extensión de funcións: Permite extender clases con novas funcionalidades se necesidade de ter acceso o código da clase.
- É altamente interoperable: Podese utilizar librerías e clases de java no mesmo proxecto.

incluir referencia a libro kotlin

A aplicación contará con catro compoñentes. O principal será a MainActivity, a encargada de iniciar as compoñentes a visualizar na pantalla, executar as ordes e mostrar a información. O segundo será o layout onde se definiran as compoñentes a visualizar e a súa posición en pantalla. O terceiro a clase request a encargada da conexión co servidor e de transmitirle as ordes. O cuarto elemento é a clase encargada de recibir o vídeo e decodificalo.

## 4.2 Actividade principal

A actividade principal ou MainActivity dunha aplicación android é a primeira pantalla que aparece cando executamos a aplicación e a encargada de controlar o seu funcionamento e a súa interface de usuario. Neste caso a actividade será a encargada de mostrar os botóns e encargarse do seu funcionamento, amosar o estado da conexión, encargarse da xestión dos sensores e reproducir o vídeo. O mesmo tempo encargarase de instanciar e comunicarse coas clases encargadas da conexión co dispositivo e da recepción de vídeo.

### 4.2.1 Layout

Un layout é a definición da estrutura da interface de usuario. Esta actividade contará con dous layouts un para a posición vertical e outro para a posición horizontal da pantalla

#### Layout vertical

Este layout contará cunha superficie reservada para o video na metade superior da pantalla. Na metade inferior situaranse os botóns encargados de executar as ordes. Na parte superior situase a barra de estado que na súa parte dereita contará cun botón para conectar que servirá o tempo de indicador de conexión, a súa esquerda mostrarase unha barra para controlar a intensidade dos leds.

imaxes layout vertical

#### Layout horizontal

Neste layout o video mostrarase como o fondo e os botóns sobre él. Os de xiro a esquerda e xiro dereita situados a ambosdous lados e o resto incluíndo o de control de intensidade lumínica e o de conexión situaranse na parte de inferior da pantalla.

imaxes layout horizontal

### 4.2.2 Ciclo de vida da actividade

En android cada actividade conta cun ciclo de vida, pasa por varios estados dende antes de iniciarse ata despois de finalizar. O estado principal dunha actividade é activa, no momento que a actividade está en primeiro plano e interactuando co usuario.

imaxe ciclo actividade

Para xestionar o que sucede no resto de estados utilízanse unhas funcións de callbacks, no noso caso realizaremos as seguintes accións en cada fase:

#### onCreate

Este método é o que se chama ao executar a actividade. Nel iniciaremos os compoñentes a mostrar en pantalla e definiremos o seu comportamento.

Aquí xestionaremos o estado da conexión, iniciando a clase request se non está en funcionamento e enviando mensaxes o dispositivo para comprobar que segue conectado. Tamén xestionaremos o estado da transmisión de video xa que será necesario iniciar a recepción antes de iniciar a transmisión.

**onResume**

Este método executase despois de onCreate cando a pantalla xa é visible para o usuario. Resxistrarase aquí un sensor listener para obter a información do sensor de luz.

**onPause**

Este método executase cando a aplicación deixa de estar en primer plano, se o usuario a minimiza u outra aplicación executase a actividade pasará a onStop se despois de acceder a aplicacións recentes a aplicación volve a primer plano pásese o método onResume.

No noso caso neste método cancelaremos o rexistro do sensor xa que non se seguirá a utilizar e deteremos a transmisión de vídeo se se esta a executar.

**onDestroy**

Neste método a actividade é detida completamente e devense liberar os recursos. Aquí enviaremos a orde para deter a conexión co dispositivo.

**4.2.3 Botones**

A actividade é a encargada de iniciar os botóns e manexar o seu funcionamento. Estes botóns son os seguintes:

- **Esquerda:** O pulsar este botón enviarase unha orde de acender a luz de xiro a esquerda. Se hai algunha luz acesa se enviarase primeiro unha orde para apagala. O pulsalo por segunda vez se enviara a orde de apagar luz de xiro e no caso de que a luz vermella estivese acesa antes de indicar o xiro esta luz se acenderá de novo. O botón palpebrará en amarelo mentres estea aceso.
- **Dereita:** O seu funcionamento é o mesmo que no botón esquerda.
- **Vermello:** O pulsalo enviarase unha orde para acender a luz de vermella, o botón non funcionara se algunha das luces de xiro está acesa.
- **Noite:** O pulsalo activarase o modo noite, no que o sensor lumínico do móbil encargárase de acender a luz vermella cando a luz ambiente baixe de certo umbral.
- **Freo:** Activa o modo de freada no que acenderase a luz vermella progresivamente cando o acelerómetro do dispositivo móbil detecte unha freada.
- **Brillo:** Este botón despregará unha barra cun indicador que poderemos deslizar para elixir a intensidade das luces

- **Conexión:** Este botón enviará unha orde de conexión, unha vez conectado cambiará a súa aparencia para indicar que existe conexión co dispositivo.

#### 4.2.4 Sensores

Unhas das ventaxas de utilizar un dispositivo mobil é que estes contan con diversos sensores para moitos fins. No noso caso utilizarase o acelerómetro e o xiroscopio para rexistrar cambios na posición e aceleracións no dispositivo e o sensor de luz para medir a intensidade de luz no ambiente. En android accederemos ao sensor instanciando a clase `SensorManager` e definindo unha instancia do sensor da que obteremos os datos mediante a función `onSensorChange`

##### Acelerómetro

##### Sensor de Luz

O sensor de luz é un fotorreceptor que xenera unha sinal eléctrica dependendo da incidencia de fotóns. Para utilizar o sensor rexistramos un sensor listener no método `onResume` e o método de callback `onSensorChanged` executarase cada vez que se detecte un cambio, neste método defínase o comportamento do sensor: Cando o botón de Noite está activado encenderase o botón Vermello se o sensor rexistra un valor inferior a 400 lumens. Este valor corresponde coa intensidade lumínica o comezo do solpor. incluír referenciaaa

#### 4.2.5 Superficie de vídeo

Iniciárase a superficie de vídeo na fase de creación da aplicación, procederase a reflexar a superficie para facer un efecto de espello para facer máis natural a visualización do vídeo. O vídeo asignárase a superficie iniciando a clase `VideoReceiver` mediante as funcións de callback que se executarán cando se produza un cambio na superficie de vídeo.

### 4.3 Comunicación co dispositivo

Crearase unha clase `Request` encargada de todas as comunicacións co dispositivo. Esta clase, unha vez instanciada, encargarase de establecer a conexión co dispositivo, reconectar se se perde a conexión e transmitirle as ordes.

#### 4.3.1 Broadcast

Para coñecer a dirección IP do servidor, a clase `request` conta cunha función que abrirá un socket no porto 5555 para esperar a recepción dun datagrama broadcastado a todas as

direccións. O recibir o datagrama comprobase que contén a mensaxe "BikeView" se é así a función devolverá a dirección ip emisora do paquete.

#### **4.3.2 Conexión**

Unha vez obtida a dirección ip a clase request utilizará unha función para establecer a conexión que devolverá un socket conectado ao socket remoto do servidor.

#### **4.3.3 Transmisión de ordes**

Para transmitir as ordes a clase request contará cunha función que recibe a mensaxe a transmitir e a envía a través do socket. Espera a recibir resposta do servidor e se é positiva devolve o valor booleano verdadeiro, en caso de non recibila devolve o valor falso.

### **4.4 Recepción de video**

Para este apartado creárase a clase VideoReceiver encargándose. O video é transmitido codificado no formato H.264

### **4.5 Anclaxe a bicicleta**

Existen diversas opcións para suxeitar o móbil o guiador dunha bicicleta, no mercado hai soportes para dispositivos concretos e outros que se adaptan ao tamaño e forma de diferentes modelos. No noso caso utilizaremos unha código SCAD que executaremos co software de deseño 3D paramétrico OpenSCAD no que introduciremos as dimensións do dispositivo e como resultado obtemos un arquivo STL co deseño 3D do soporte. Vendo que os soportes impresos non constaban coa resistencia esperada optouse por utilizar un soporte de aluminio nas probas para preservar a integridade do dispositivo móbil.



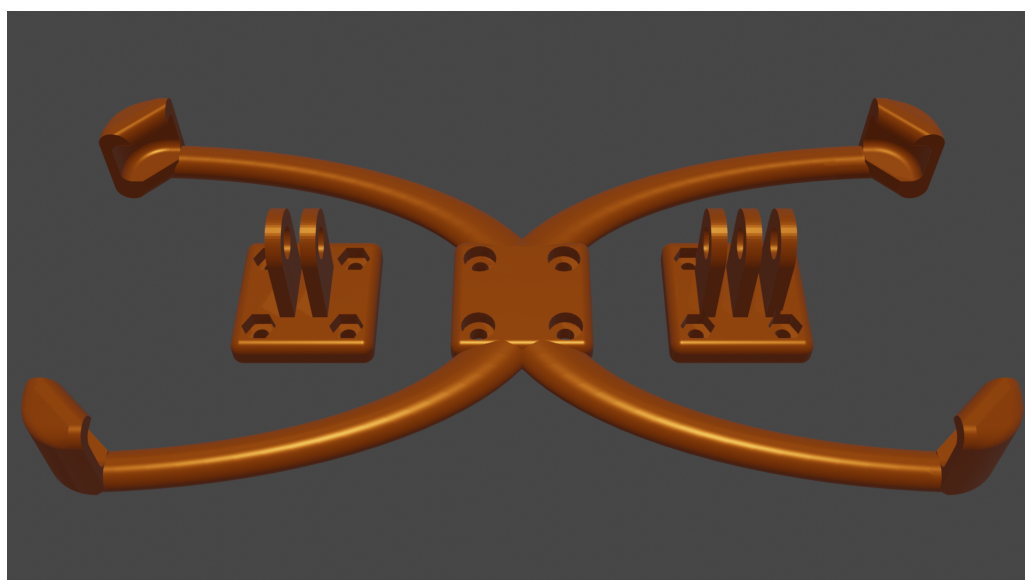


Figura 4.1: Desño 3D do soporte para o dispositivo móbil.

correspóndense

# Evaluación experimental

---

Para comprobar que os resultados obtidos correspóndense co plantexado probaremos o dispositivo e a aplicación. Realizaremos probas en contornos controlados e probas no medio real o que esta dirixido.

## 5.1 Consumo e autonomía

Comenzarase as probas medindo o consumo de amperios do sistema. Para elo colocase un amperímetro usb entre unha fonte de alimentación e a conexión usb coa que alimentase o sistema nestas probas. Repetiremos as probas con diferentes fontes de alimentación e distintos cables para descartar fallos e conseguir unha maior consistencia nos resultados. A continuación realizaranse as mesmas probas no dispositivo medindo o consumo alimentándoo coa batería.

Analizaranse os seguintes supostos para o dispositivo con un anel led e para o que conta a maiores coas dúas tiras led.

- **Sistema en repouso:** O sistema esta aceso pero só se estean a executar as funcións do sistema operativo incluíndo o servidor ssh para o control remoto.
- **Servidor funcionando:** Executamos o servidor.
- **Cliente conectado:** Conectamos o dispositivo móbil o servidor.
- **Vídeo transmitindo:** Transmitimos vídeo en directo o dispositivo móbil.
- **Vídeo parado:** Paramos a transmisión de vídeo.
- **Desconexión do cliente:** Pechamos a aplicación no dispositivo móbil.
- **Luces intermitentes:** Iniciamos as luces intermitentes a máxima intensidade nunha das direccións, consumo varia no proceso, rexistremos o valor máximo.

	24 luces	24 luces e batería	8 luces
Sistema en repouso	165mA	148mA	
Servidor en funcionamento	165mA	148mA	
Cliente conectado	165/168mA	148mA	
Vídeo transmitindo	360mA	342mA	
Vídeo parado	211mA	191mA	
Desconexión do cliente	165mA	148mA	
Luces intermitentes	428mA	414mA	
Luces vermellas	505mA	502mA	
Luces vermellas e vídeo	660mA	645mA	

Táboa 5.1: Comparativa de consumo de amperios

- **Luces vermellas:** Acendemos as luces vermellas a intensidade máxima.
- **Luces vermellas e transmisión de vídeo:** Consumo coas luces vermellas a máxima intensidade e co vídeo transmitindo en directo.

Os resultados obtidos son os seguintes:

As fontes de alimentación utilizadas poden proporcionar un máximo de 3A e 2.1A respectivamente, tamén utilizáronse dous cables un de maior calidade e outro cunha calidade inferior. En ningún dos casos obtivéronse diferencias apreciables sendo a maior diferenca de 4mA. Na versión con batería necesitouse utilizar cableado a maiores para realizar as probas, o que pode que incrementara lixeiramente o consumo.

A seguinte proba a realizar será a de autonomía do dispositivo, para elo buscarase o consumo máximo acendendo as luces vermella a máxima intensidade o tempo que se transmite vídeo.

Comezaranse as probas co dispositivo dotado de batería interna

## 5.2 Vídeo e lentes

## 5.3 Visibilidade

## 5.4 Estabilidade e consumo da aplicación

## Conclusións

---

**D**ERRADEIRO capítulo da memoria, onde se presentará a situación final do traballo, as leccións aprendidas, a relación coas competencias da titulación en xeral e a mención en particular, posibles liñas futuras,...

### 6.1 Traballo futuro

Mellorar interfaces da app, subila a tenda de aplicacións de google Engadir mecanismos de control, control por voz, mando no guiador, pulsadores intermitentes Detector de caídas Soporte a multiples dispositivos, visualizar diferentes camaras, utilidade en outros ambitos, camións etc. Mellorar a estrutura do dispositivo, facelo resistente a auga e golpes, protexer os leds, Estudar a unclution de difusores de luz e ou lentes nos leds para aumentar a distancia de visualización e oou reducir o Consumo Luces frontais e laterais Aumentar a batería e o mecanismo de protección desta para aumentar o tempo de uso



# Apéndices

---

**.1 Contido DVD**

**.2 Requisitos e instalación**





# Glosario de acrónimos

---

**ERLANG/OTP** *Erlang Open Telecom Platform.*

---

## Glosario de termos

---

**Bytecode** Código independente da máquina que xeran compiladores de determinadas linguaxes (Java, Erlang,...) e que é executado polo correspondente intérprete.

---