



UNIVERSIDADE DA CORUÑA

FACULTADE DE INFORMÁTICA

TRABALLO FIN DE GRAO

GRAO EN ENXEÑARÍA INFORMÁTICA

Mención en Enxeñaría do Software

**Aplicación web para a análise do  
comportamento humano en zoas transitadas**

**Autor:** Xoan Iago Suárez Canosa

**Director:** Cancela Barizo, Brais

**Director:** González Penedo, Manuel Francisco

**Director:** Novo Buján, Jorge

**Director:** Ortega Hortas, Marcos

A Coruña, 4 de setembro do 2015



*A Meus pais Elvira e Manolo, que grazas ao seu  
esforzo e cariño permitíronme chegar ata aquí*



# Agradecementos

En primeiro lugar grazas a Brais Cancela por toda a axuda e o apoio que me brindou ao longo destes meses, xa que sen a súa achega este proxecto non tería nin sequera comezado. Grazas tamén a tódolos compañeiros da facultade que coñecín ao longo deste incríbles anos, deles aprendín a meirande parte do que sei e ademais pasamos momentos inesquecibles. Por último agradecer á miña familia e ao resto dos meus amigos por soportar día a día cada un dos meus defectos e permitirme gozar das vosas virtudes.



# Resumo

No mundo da seguridade, un dos maiores retos que se propoñen hoxe en día é a detección de condutas sospeitosas. Esta situación volvese cada vez máis complexa ao aumentar o número de cámaras a vixiar, polo que é imprescindible dispoñer dunha ferramenta que facilite esta tarefa.

Este proxecto consiste nunha aplicación web que emprega funcionalidades relativas á análise do comportamento en zoas transitadas, incluíndo visualización de vídeo e distintas capas que mostran información de alto nivel sobre o comportamento detectado.

En concreto esta ferramenta encargase de detectar a todos aqueles obxectos ou persoas que aparecen nunha secuencia de vídeo, aplicando un algoritmo para medir o “graou de anormalidade” da súa conduta en base aos movementos que realizan.

## Palabras clave

Análise Comportamento, Comportamento Humano, Aplicación Web, Django, Python, Javascript.



# Índice xeral

<b>1. Introdución e Obxectivos</b>	<b>1</b>
<b>2. Fundamentos Teóricos e Conceptos Previos</b>	<b>3</b>
2.1. Arquitectura web . . . . .	3
2.2. Análise do comportamento . . . . .	4
2.3. Programación Web . . . . .	5
2.3.1. Desenvolvemento Áxil . . . . .	6
2.3.2. AJAX . . . . .	8
2.3.3. Outras cuestiós da web . . . . .	8
2.4. O Vídeo . . . . .	8
2.4.1. Codec's . . . . .	8
2.4.2. Formato de Vídeo . . . . .	9
2.4.3. Streaming de Vídeo . . . . .	10
<b>3. Análise de antecedentes e alternativas</b>	<b>13</b>
3.1. Software de carácter privativo . . . . .	13
3.2. Software libre . . . . .	14
<b>4. Metodoloxía seguida no desenvolvemento de Proxecto</b>	<b>17</b>
4.1. As metodoloxías áxiles . . . . .	17
4.2. Persoas . . . . .	18
4.2.1. ProductOwner . . . . .	18
4.2.2. ScrumMaster e Development Team . . . . .	18

4.3.	Reunións . . . . .	19
4.3.1.	Sprint Planning Meeting . . . . .	19
4.3.2.	Daily Scrum . . . . .	19
4.3.3.	Sprint Review . . . . .	19
4.3.4.	Sprint Retrospective . . . . .	19
4.4.	Control de Versións con GitHub . . . . .	19
4.5.	Integración Continua con Travis CI . . . . .	20
4.6.	Control da cobertura con Coveralls . . . . .	20
4.7.	Xestión de Incidencias e Control de Proxecto con YouTrack . . . . .	22
<b>5.</b>	<b>Tecnoloxías Empregadas</b>	<b>25</b>
5.1.	Estudo comparativo das tecnoloxías web . . . . .	25
5.1.1.	Back-End . . . . .	26
5.1.2.	Front-End . . . . .	27
5.2.	Navegador Ciente . . . . .	29
5.2.1.	HTML5 . . . . .	29
5.2.2.	CSS3 . . . . .	31
5.2.3.	Javascript . . . . .	32
5.3.	Sistema de Recoñecemento . . . . .	35
5.3.1.	OpenCV . . . . .	36
5.3.2.	Extensible Markup Language (XML) . . . . .	36
5.4.	Servidor Web . . . . .	37
5.4.1.	FFmpeg . . . . .	37
<b>6.</b>	<b>Desenvolvemento</b>	<b>39</b>
6.1.	Funcionalidades desexadas . . . . .	39
6.2.	Primeiro Sprint: Cargar e Visualizar vídeos . . . . .	40
6.2.1.	Carga de Vídeo . . . . .	41
6.2.2.	Reproducción de Vídeo . . . . .	42
6.3.	Segundo Sprint: Subida e conversión de vídeos . . . . .	45
6.3.1.	Lista de vídeos e Imaxe de Portada . . . . .	46

6.3.2. Sistema de Notificacións . . . . .	47
6.4. Terceiro Sprint: Xerar e mostrar deteccións básicas . . . . .	50
6.4.1. Análise de Vídeo . . . . .	51
6.4.2. Mostrar Deteccións . . . . .	55
6.5. Cuarto Sprint: Arranxar bug's e mellorar a estratexia de probas . . . . .	57
6.5.1. Re-analizar vídeo . . . . .	59
6.6. Quinto Sprint: Traxectorias . . . . .	60
6.6.1. Táboa de Deteccións . . . . .	62
6.6.2. Lista de Deteccións actuais . . . . .	64
6.7. Sexto Sprint: Detección do comportamento anómalo . . . . .	66
6.7.1. Popups de deteccións sospeitosas . . . . .	70
6.8. Sétimo Sprint: Memoria e posta en producción . . . . .	73
6.8.1. Dominio . . . . .	74
6.8.2. Hosting . . . . .	74
6.8.3. Configuración do Servidor . . . . .	76
6.8.4. Axustes de Produción . . . . .	76
<b>7. Validación</b> . . . . .	<b>79</b>
7.1. Probas Unitarias . . . . .	79
7.1.1. Política de acceso ás páxinas web . . . . .	79
7.1.2. Probas da Capa Web con Javascript . . . . .	80
7.2. Probas de Integración . . . . .	81
7.2.1. Probas Funcionais Selenium . . . . .	81
7.3. Probas de Sistema . . . . .	82
7.4. Probas de Aceptación . . . . .	82
7.5. Calidade . . . . .	82
<b>8. Planificación e Avaliación de Custos</b> . . . . .	<b>85</b>
<b>9. Resultados e Conclusións</b> . . . . .	<b>93</b>

<b>10. Liñas Futuras</b>	<b>95</b>
10.1. Incidencias rexistradas como abertas ou re-abertas en YouTrack . . . . .	95
10.2. Outras funcionalidades de interese . . . . .	97
10.2.1. Vídeo en Directo . . . . .	97
10.2.2. Panel de Control . . . . .	97
<b>A. Apéndice</b>	<b>99</b>
A.1. Notas acerca da Terminoloxía . . . . .	99

# Índice de figuras

2.1.	Diagrama simplificado da Arquitectura do Sistema . . . . .	4
2.2.	Clásica arquitectura dunha aplicación web empresarial . . . . .	6
2.3.	Diagrama sobre os conceptos de programación . . . . .	7
2.4.	Diagrama conexión RTPS . . . . .	11
4.1.	Imaxe de parte do ficheiro <code>.travis.yml</code> . . . . .	21
4.2.	Panel Áxil da ferramenta de xestión de incidencias YouTrack . . . . .	23
5.1.	Diagrama da Arquitectura do Sistema . . . . .	30
6.1.	Formulario para a creación de vídeos . . . . .	41
6.2.	Diagrama de secuencia do proceso tras o envío do formulario de creación de vídeos (1) . . . . .	43
6.3.	Diagrama de secuencia do proceso tras o envío do formulario de creación de vídeos (2) . . . . .	44
6.4.	Etiqueta vídeo de html5, coas súas fontes e coas capas <code>&lt; canvas &gt;</code> asociadas . . . . .	45
6.5.	Captura de pantalla da páxina web SuccessfulUpload . . . . .	47
6.6.	Capturas de pantalla do sistema de notificacións . . . . .	48
6.7.	Diagrama de clases do sistema de subida . . . . .	49
6.8.	Diagrama de secuencia da análise de vídeo na capa web . . . . .	50
6.9.	Interfaze de liña de comandos do sistema de recoñecemento . . . . .	52
6.10.	Diagrama de clases do paquete XmlRecognition . . . . .	54
6.11.	Diagrama de clases simplificado do patrón observador na capa web . . . . .	56

6.12. Xeración do atributo fps no lado servidor . . . . .	58
6.13. Diagrama completo da clase Detection . . . . .	59
6.14. Capturas de pantalla que amosan os distintos estados do botón de re-analise . . . . .	60
6.15. Diagrama do cálculo da traxectoria . . . . .	61
6.16. Captura que amosa o funcionamento das capas de traxectorias e deteccións	62
6.17. Diagrama que mostra o aclarado de cores . . . . .	63
6.18. Captura de pantalla que amosa a táboa de obxectos detectados . . . . .	63
6.19. Diagrama de clases que amosa detalladamente a relación entre as clases Detection e VideoDetections . . . . .	65
6.20. Captura de pantalla que amosa a lista de obxectos detectados . . . . .	66
6.21. Captura de pantalla que amosa o popover da lista de deteccións actuais	67
6.22. Ratio de anormalidade gardado nun documento XML . . . . .	68
6.23. Captura de pantalla do componente slider da biblioteca jQuery UI . . .	69
6.24. Captura de pantalla que amosa o comportamento das deteccións sospeitosas . . . . .	70
6.25. Diagrama de clases simplificado para a ventá emerxente de deteccións sospeitosas . . . . .	72
6.26. Captura de pantalla que amosa a ventá de deteccións sospeitosas . . . . .	73
6.27. Diagrama que explica os problemas resoltos para implementar o efecto zoom . . . . .	74
6.28. Panel de control da ferramenta 1and1 . . . . .	75
8.1. Gráfica do fluxo de tarefas acumulado ao longo do proxecto . . . . .	88
8.2. Gráfica da evolución do proxecto . . . . .	89
8.3. Táboa de horas adicadas ao proxecto por versión (1) . . . . .	90
8.4. Táboa de horas adicadas ao proxecto por versión (2) . . . . .	91

# Índice de Táboas

6.1. Táboa de formatos de vídeo soportados polos distintos navegadores . . . 46



## Capítulo 1

# Introdución e Obxectivos

O seguimento de obxectos é o proceso de estimar no tempo a localización de un ou máis obxectos en movemento empregando as imaxes captadas por una cámara. A crecente mellora na potencia de cálculo dos procesadores actuais, xunto coa dixitalización dos sensores de imaxe propiciou dende comezos de século a aparición de novos algoritmos de análise que aportan cuantiosas melloras a este campo.

Neste aspecto, o Grupo de Visión Artificial e Recoñecemento de Patróns (VARPA) da UDC leva anos investigando para aportar á comunidade científica os seus propios algoritmos e desenvolver novas aplicación que empreguen estes algoritmos para detección de persoas, vehículos, ou calquera outro obxecto susceptible de seres estudiado. En concreto, o grupo posúe ferramentas que permiten o seguimento en zoas transitadas nas que poden aparecer multitude de obxectos a seguir simultaneamente.

Co fin de achegar estes métodos de análise á súa aplicación final, proponse dende o laboratorio a construción dunha web, que sexa capaz de reproducir vídeos, e sobre eles mostrar distintas capas con información de alto nivel, como pode ser a resultante de detectar obxectos, medir o seu grao de anormalidade, a súa velocidade, etc.

Seleccionase unha arquitectura web xa que a diferencia das arquitecturas de escritorio, proporciona ás persoas que acceden á web independencia do Sistema Operativo empregado e dispoñibilidade dende calquera lugar con acceso a rede, evitando así as dificultades asociadas coa instalación ou actualización da aplicación.



## Capítulo 2

# Fundamentos Teóricos e Conceptos Previos

O desenvolvemento dunha aplicación web non é algo trivial e máis se temos en conta todas as peculiaridades que este proxecto contén. Para a súa comprensión é preciso coñecer unha serie de conceptos teóricos que se expoñen a continuación:

### 2.1. Arquitectura web

No proxecto séguese unha Arquitectura Web baseada no modelo cliente-servidor, que consiste nun lado servidor que distribúe os recursos como poden ser o contido multimedia (vídeos, imaxes, etc) ou as páxinas web ao outro lado, o cliente, que tipicamente corre nun navegador web interpretando as páxinas html e o código javascript asociado a estas.

Neste caso a parte servidor estará dividida en dúas componentes claramente diferenciadas, o sistema para a análise do comportamento e a aplicación web que permitirá o acceso a este, os fundamentos de ámbalas dúas partes pódense ver no diagrama 2.1 e os fundamentos nos que están baseadas explícanse a continuación.

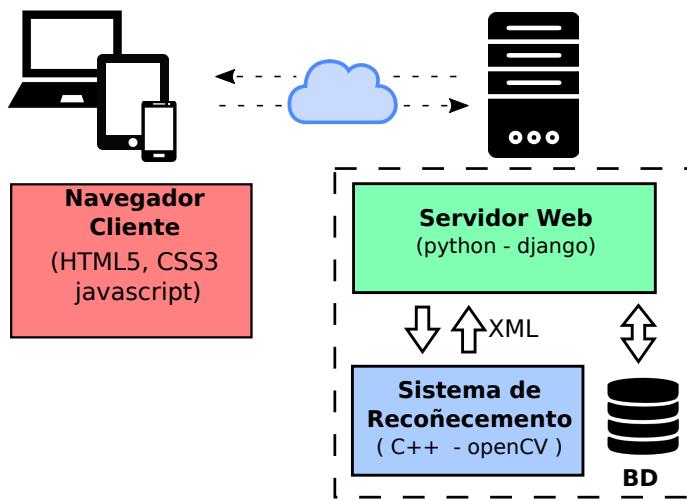


Figura 2.1: Diagrama simplificado da Arquitectura do Sistema

## 2.2. Análise do comportamento

É un dos campos de investigación mas activos hoxe en día. A idea principal na que se centran estes sistemas como o que nos ocupa é a de detectar calquera acción levada a cabo polos obxectos involucrados nunha escena de vídeo. Un obxecto é calquera cousa que debe ser seguida, polo que dependendo do tipo de problema estes obxectos poden ser dunha natureza ou doutra. O tipo de accións a detectar tamén depende da tipoloxía do sistema, xa poden ser comportamentos individuais( camiñar, correr, loitar...) ou grupais (reunirse, abandonar un grupo de persoas...).

Tanto neste proxecto como nos sistemas para a análise do comportamento en xeral, pódense discernir tres tarefas importantes que colaboran entre si [1]:

- **Detección de Obxectos:** Partindo dunha secuencia de vídeo como entrada obtéñense os distintos obxectos que aparecen en cada fotograma da escena. Para este fin empréganse técnicas de visión por computador.
- **Seguimento de Obxectos:** A partires da información obtida na detección, asignanselle identificadores a cada obxecto detectado no vídeo, agrupando se procede distintos obxectos baixo o mesmo identificador en caso de considerarse que

estes obxectos forman parte de un grupo ou unha mesma detección.

- **Análise do comportamento de Alto Nivel:** Unha vez obtida a información dos dous pasos anteriores pódese catalogar o comportamento de cada detección empregando técnicas de recoñecemento de patróns.

Os resultados más destacables destas técnicas cos que a aplicación terá que traballar serán:

- A lista de obxectos detectados para cada un dos fotogramas e a súa posición neles
- A traxectoria de cada un dos obxectos detectados
- O grao de anormalidade da traxectoria seguida por un obxecto en cada un dos fotogramas

Estas tres tarefas agrupanse baixo un mesmo sistema ao que chamaremos Sistema de Recoñecemento e que xunto co servidor web, apoiado nos fundamentos que veremos a continuación, forman o lado servidor deste proxecto.

## 2.3. Programación Web

A programación web de aplicacíons de carácter empresarial require do coñecemento da rede, ademais do de unha serie de ferramentas e estratexias para chegar a un deseño sostible e de calidade.

A arquitectura clásica das aplicacíons web pódese ver no gráfico 2.2, ela contén unha parte cliente que se executa no navegador do usuario, e unha parte servidor que á súa vez acostuma a dividirse nunha BD (Base de Datos) que almacena a información precisa, unha capa modelo que reflexa o modelo de negocio da nosa aplicación tipicamente nalgúnha linguaxe de programación e por último unha capa de IU Web (Interface de Usuario Web) que se encarga de transformar os datos da capa modelo a un formato web comprensible polo cliente e viceversa.

As técnicas e estratexias más importantes á hora de construír unha aplicación web relátanse nos puntos subseguistes:

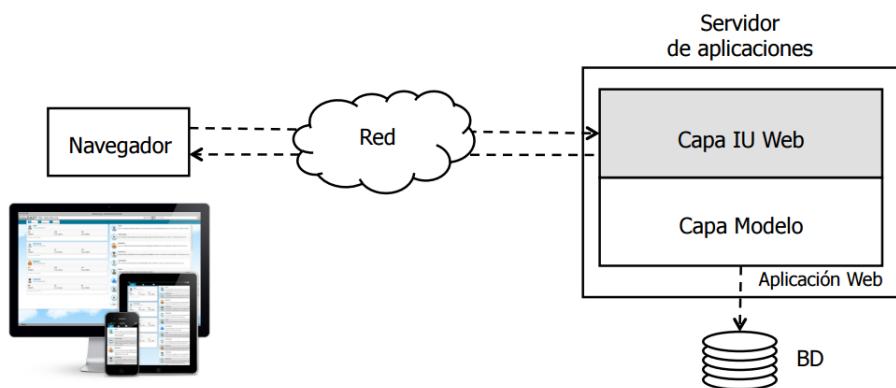


Figura 2.2: Clásica arquitectura dunha aplicación web empresarial

### 2.3.1. Desenvolvemento Áxil

Para reducir custos e poder proporcionar solucións rápidas é preciso que as aplicacións web's de carácter empresarial se leven a cabo en pouco tempo e con bos principios de enxeñaría, a isto contribúen en gran medida as estratexias de programación que se explican a continuación e que se ven no diagrama 2.3.

#### Soporte para transaccións

Unha transacción nun Sistema Xestor de Base de Datos (SGBD) é un conxunto de ordes que se executan formando unha unidade de traballo, de forma invisible e atómica. As transaccións cobran gran importancia nas aplicacións web debido á inestabilidade da rede e á concorrencia dos distintos clientes conectados, polo que é axuda a un desenvolvemento moi áxil que a tecnoloxía traia a súa xestión integrada.

#### Object-Relational Mapping (ORM)

Os mapeado obxecto-relación é unha das técnicas de programación que máis velocidade imprimen na construción de webs, xa que converte os datos dunha linguaxe Orientada a Obxectos (OO) a datos de un sistema relacional no que son persistidos e viceversa, aforrando ao programador o traballo de ter que programar o código pa-

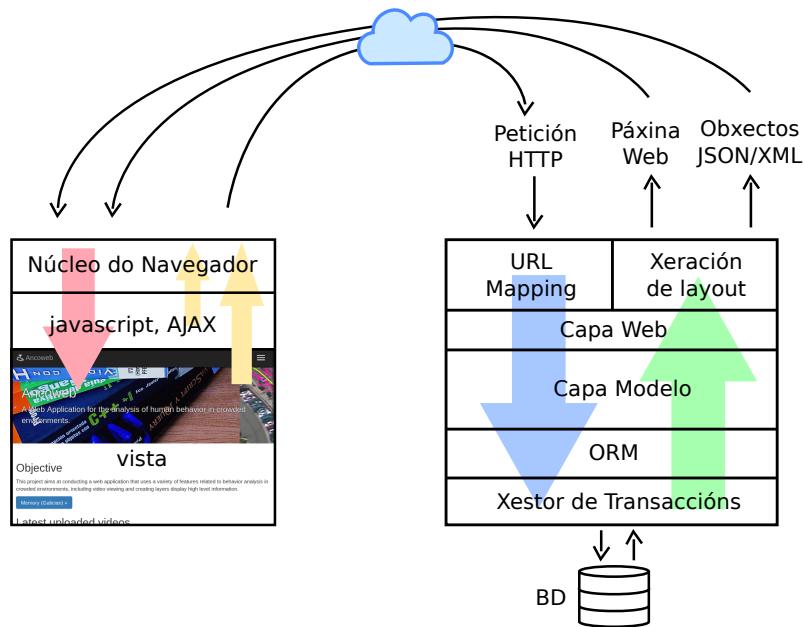


Figura 2.3: Diagrama sobre os conceptos de programación

ra esta tarefa. É deseable pois que a tecnoloxía a empregar dispoña dun mapeador obxecto-relacional ben integrado, algúns exemplos disto poden ser: a combinación Java+Maven+Hibernate, o EF(Entity Framework) de Microsoft ou os Models de Django.

## Xestión de Layout

Tamén resulta moi práctico dispoñer dunha linguaxe de plantilla que permitan xesar contido html ben estruturado dinamicamente. Algúns exemplos son o Sistema de Templates de Django, o Sistema JSP de Spring, a libraría Thymeleaf ou os compoñentes de ASP.NET. Todos eles axudan a xesar contido HTML de xeito sinxelo e scalable que logo será enviado ao cliente. De tódolos xeitos esta parte cliente as veces precisa comunicarse co servidor sen que sexa preciso unha recarga da páxina e para elo empregase AJAX.

### 2.3.2. AJAX

AJAX ou Asynchronous JavaScript And XML é unha tecnoloxía da web empregada para crear aplicacións interactivas, estas aplicacións executanse no navegador dos usuarios mentres manteñen unha comunicación asíncrona co lado servidor en segundo plano. Normalmente empregase javascript coma linguaxe para a realización das chamadas asíncronas en combinación con algunha linguaxe para a definición de obxectos como XML ou JSON, para as que ademais os propios navegadores adoitan a facilitar ferramentas de parsing.

### 2.3.3. Outras cuestiós da web

A maiores existe toda unha gama de outras funcionalidades que cobran importancia cando deseñamos e construímos unha web como o manexo de errores nos formularios, internacionalización (i18n), visualización de grande cantidades de datos (en listas ou táboas), seguridade...

## 2.4. O Vídeo

O vídeo permite gravar, procesar, almacenar e transmitir información en forma de imaxe en movemento, esta imaxe en movemento soe estar composta por unha serie de imaxes estáticas chamadas fotogramas, que se manexan a unha velocidade alta causando o efecto de que o que se está a ver está en movemento.

Non obstante o vídeo en formato electrónico non almacena necesariamente todas as imaxes de forma individual xa que isto suporía moita información redundante. En lugar disto empréganse técnicas de codificación-decodificación (codec's) que comprimen e descomprimen os datos para facilitar o seu manexo.

### 2.4.1. Codec's

Os codec's teñen como función principal a de transformar unha sinal de vídeo para que poida ser visto. A maioría dos codec's provocan con cada transformación unha perda de información para conseguir un tamaño final o máis pequeno posible, estes codec's

chámanse lossless (con perdida) e a pesares de que perden calidade soe compensar pola cantidade de espazo que aforran ao comprimilo vídeo.

A parte de este fenómeno da compresión tamén cobra moita importancia outros aspectos relacionados co vídeo como a reprodución e sincronización de son, os subtítulos do vídeo, as imaxes representativas deste... Todo isto depende do formato no que se almacene o vídeo.

#### 2.4.2. Formato de Vídeo

O formato dun vídeo determina como se almacenan os distintos tipos de información involucrada como as imaxes que poden estar codificadas en varios codec, o son, os subtítulos... este formato correspondece cunha extensión específica do arquivo que o contén, como por exemplo:

- **AVI (Audio Video Interleaved):** Sendo un dos formatos más famosos pode conter un vídeo dunha calidade excelente pero soe requerir dunha gran capacidade. Os codec's que se soen empregar neste formato pola súa capacidade de compresión e calidade aceptable son DivX e XviD, inda que tamén se permiten outros como DV(Digital Video), CinePak...
- **MKV (Matroska):** É un formato de código aberto que basea o seu nome nas clásicas bonecas Matroskas. Ten capacidade para conter tanto vídeo, son e subtítulos en diferentes idiomas, empregándose como códec de video normalmente algunha implementación de H.264, como por exemplo x264. Mientras que para o son é habitual empregar o codec de audio Vorbis.
- **WebM (Google, 2010):** Un dos formatos más recentes é o WebM (WebMovie), un proxecto lixeiramente baseado en Matroska adquirido e liberalizado por Google en 2010 co obxectivo de empregalo con HTML5 como estándar libre. O formato ten un excelente rendemento e xunto ao codec VP9, motivo polo cal forma parte dos recomendados pola W3C.
- **Formato OGG (Xiph.Org, 1993):** O formato contedor OGG é un formato libre deseñado para incluír vídeo, son, subtítulos e metadatos. O vídeo en este

formato soe estar codificado co codec Theora, que se basea nunha versión liberada de VP3. Tamén se emprega para este tipo de empaquetado a extensión .OGV, mais o que marca o estandar é a extensión .OGG.

- **MP4 - MPEG (Moving Pictures Expert Group):**

O Moving Picture Experts Group (MPEG) é un grupo de expertos da ISO (Organización Internacional de Normalización) e da Comisión Electrotécnica Internacional (IEC) para crear estándares en canto ao mundo do audio e o vídeo.

Froito do traballo deste grupo naceron os formatos MPEG-1 (calidade CD), MPEG-2 (calidade DVD), MPEG-3 (orientado ao audio MP3) e MPEG-4 que é o que máis nos interesa xa que vai enfocado á compresión de vídeo e son na web, podendo incluír tamén subtítulos ou imaxes de referencia. O ficheiros de este último formato teñen extensión .mp4.

#### **2.4.3. Streaming de Vídeo**

Dado que este proxecto está centrado no tratamiento de vídeo, é de especial importancia ver de que xeitos podemos distribuílo e reproducilo a través da rede. A estes efectos existen dúas grandes alternativas que varían en canto ao seu grao de escalabilidade, dificultade de implementación, e calidad final do servizo:

##### **Pseudo Streaming ou Descarga Progresiva**

Consiste na descarga do vídeo por fragmentos, típicamente emplegando o protocolo HTTP. Neste formato, o reprodutor vai acumulando fragmentos de vídeo ata obter os precisos como para comenzar a reproducción, mais se o ancho de banda fose insuficiente, o vídeo remataría por pararse. Este sistema é o empregado por servizos como YouTube, Vimeo, DailyMotion...

Será a opción empregada por motivos de simplicidade, mais compre explicar tamén o verdadeiro Streaming, xa que é a diferenza do pseudo-streaming pode ser empregado para a emisión de contido en directo como o dunha cámara de seguridade.

## Streaming

O verdadeiro streaming ( do inglés True Streaming) consiste na emisión en directo do contido multimedia a través da rede, que o reprodutor reproduce no momento que recibe. Este outro xeito de distribuír vídeo, apoiase en axustar a calidade do vídeo ao ancho de banda do que dispón o cliente, evitando así interrupcóns na reproducción.

O protocolo máis destacable á hora de empregar este tipo de streaming é RTSP (Real-Time Streaming Protocol) que operando a nivel de aplicación permite controlar un ou varios fluxos sincronizados de contido multimedia como se pode ver na figura 2.4.

Por unha parte RTSP soe empregar o Real-Time Transport Protocol (RTP) sobre UDP(User Datagram Protocol) para o transporte de contido multimedia, maximizando así o emprego da rede pero sen garantir un mínimo na calidade do servizo.

E por outra parte RTSP emprega o Real-time Control Protocol (RTCP) sobre TCP(Transmission Control Protocol) para a transmisión periódica de paquetes de control da sesión, o diagnóstico de fallos e o control de la calidade da transmisión.

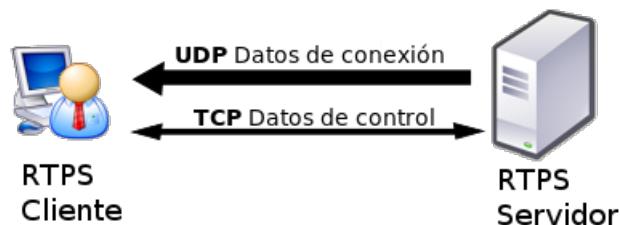


Figura 2.4: Diagrama conexión RTPS

RTSP asemellase a HTTP no formato das peticións/repostas e na sintaxe, pero dispoñendo dun estado que permite tanto a clientes como a servidores fazer peticións.

Tamén existen outros protocolos propietarios como MMS (Microsoft Media Server) ou RTMP (Real-Time Messaging Protocol) e RTMFP (Real-Time Media Flow Protocol) de Adobe.



## Capítulo 3

# Análise de antecedentes e alternativas

En canto aos produtos existentes ate o momento que traballan no ámbito da detección de obxectos e a análise do comportamento, hai que diferenciar aqueles produtos da empresa privada que prometen grandes resultados pero dos cales debido á falta de medios non se pode comprobar o seu correcto funcionamento, daqueles que pertencen ao mundo do software libre ou que están apoiados en documentos científicos e que si teñen unha reputación contrastada.

### 3.1. Software de carácter privativo

Do primeiro grupo pódese salientar toda unha serie de ferramentas que permiten o tratamiento de distintas señais de vídeo procedentes de varias cámaras sobre as que se executa unha análise que dependendo do producto ofrecenos uns datos ou outros. Algúns exemplos de estes productos son:

- **CyeWeb:**[2] É unha aplicación de escritorio para monitorización de cámaras de vixilancia que segundo detallan na súa páxina web permite detección de movemento, conteo de obxectos, detección de aparcamentos en zoas non permitidas, detección de obxectos conflitivos...

- **Cerebrus Intelligent Video Analytics:**[3] Outra aplicación de escritorio que tamén obtén en directo o vídeo de varias cámaras de vixilancia e mostra sobre este vídeo unha análise de alto nivel capaz de detectar intrusos, monitorizar de vehículos, detectar obxectos abandonados, contar persoas, etc.

En software privativo pero máis enfocados cara o tema da análise de comportamento e menos focalizados como programas de video-vixilancia temos outra serie de produtos como poden ser:

- **Nolus:**[4] Dende esta empresa aseguran que o seu software é capaz de recoñecer distintos patróns de comportamento en seres humanos e en animais.
- **WINanalyze:**[5] WINanalyze é unha aplicación de escritorio para a análise de vídeo procedente tanto de cámaras en vivo como de ficheiros, que permite analizar os movementos realizados por un obxecto do que se seguen distintos puntos de interese. Está baseado nunha serie de artigos científicos de comezos dos anos noventa sobre o seguimento de pixeles e puntos de interese que se poden consultar na web da empresa á que pertence[6].
- **Huygens Software - ObjectTracker:**[7] Huygens Software Suite é un conxunto de paquetes para o procesamento de imaxes. Esta suite centrada na análise de partículas inclúe funcionalidades como análise interactivo, visualización de volumes en 2D e 3D, funcionalidades para tratamiento de imaxes procedentes de microscopio ou no tracking de obxectos mediante fluxo óptico: estudo das posicións, traxectorias, velocidades ...

### 3.2. Software libre

En este apartado ao igual que no anterior analizamos dous tipos de software, que non están intrinsecamente ligados entre si, senón que parece seguir liñas un pouco independentes. Por un lado están os programas de vídeo-vixilancia que implementan algúns tipo de análise do comportamento e por outro lado as aplicacións para investigación científica que permiten detectar comportamento e estudalo mais en detalle.

Do primeiro grupo podemos destacar:

- **iSpy:**[8] iSpy é un dos software's libres para vídeo-vixilancia más completos do mercado. Emprega unha arquitectura moi semellante á que se precisa neste traballo fin de grao, cun servidor que recolle os sinais de vídeo para logo ofrecelas mediante un acceso web dende calquera navegador. Ao ser un proxecto de código libre sempre é posíbel descargalo para realizar sobre el as modificacións relativas á análise do comportamento, mais isto non é preciso posto que dispón da posibilidade de acoplar plugins que modifican ou amplían o seu comportamento. Son de especial interese os plugins dispoñibles para realizar tarefas de visión por computador[9], como poden ser detección e seguimento de actividades, recoñecemento de matriculas, conteo de persoas, detección de caras ou análise do comportamento.
- **ZoneMinder:** [10] ZoneMinder é xunto con iSpy un dos software's de vídeo-vixilancia libres más completo e potente proporcionando toda unha serie de funcionalidades para amosar vídeo de cámaras IP, de circuito de televisión ou USB. Segue ao igual que iSpy unha arquitectura cliente-servidor, e inda que non ten un abanico de plugins tan rico como o de iSpy dispón de Detección de Movemento integrado[11].

En outra liña diferente, tamén son de interese proxectos de carácter máis investigador que permiten análise do comportamento como son:

- **SwisTrack:**[12] SwisTrack é unha ferramenta increíblemente versátil para o seguimiento de obxectos, animais, humanos... empregando como fonte de datos unha cámara ou ben un arquivo de vídeo. Distribúese como unha aplicación de escritorio escrita en C++, e que emprega as funcionalidades de OpenCV. É de destacar unha opción que permite acceder aos datos da súa análise vía web. O programa dispón dunha interface TCP(Transmission Control Protocol), que amosa en formato text-based NMEA 0183 protocol o resultado dos compoñentes de saída que escriban nela.

- **Community Core Vision:**[13] É un proxecto moi similar a SwisTrack, con algo menos de transcendencia pero con soporte multiplataforma, tamén creado como interface de escritorio permite escoller entre 27 opcións de análise diferentes para o tratamento do vídeo.
- **Seguimento de Insectos ou partículas:** Neste campo tamén son de especial utilidade os programas de tracking, de feito existen dous destacables sistemas para a análise do comportamento dos insectos: BIO-TRACKING [14] e Ctrax[15], ambos permiten unha detallada análise para determinar cales son os comportamentos destes animais en función das traxectorias que seguen no seu camiño ou as zoas que máis frecuentan dentro dun determinado entorno.

## **Capítulo 4**

# **Metodoloxía seguida no desenvolvemento de Proxecto**

Os diferentes obxectivos do proxecto abordáronse seguindo a Metodoloxía SCRUM, adaptada a un proxecto de un único Developer.

Esta metodoloxía áxil tamén chamada melé pola súa inspiración no Rugby, permite un desenvolvemento rápido en situacíons de requisitos inestables. Apoiase no seu carácter iterativo e incremental, dividindo o traballo a realizar en períodos de aproximadamente un mes chamados Sprint's.

Para a realización deste traballo de fin de grao foi preciso adaptala, pois está pensada en principio para organizar equipos de entre 3 a 9 persoas (Team) mentres que neste caso só se contará cunha única persoa para desenvolver todo o proxecto. Por outro lado, o marco de traballo planifica reunións diarias (Daily Scrum), ao supoñer que todos os membros do equipo traballan unha xornada laboral enteira entre cada unha destas reunións, o cal tampouco se dá no caso deste proxecto, xa que a dedicación será de determinadas horas nos momentos dispoñibles.

### **4.1. As metodoloxías áxiles**

Escolleuse unha metodoloxía áxil como SCRUM para este proxecto xa que as metodoloxías áxiles en xeral son unha forma excepcional de minimizar os riscos asociados

grazas ao seu carácter iterativo (abordase por iteracións curtas de tempo) e incremental(as funcionalidades do proxecto crecen en cada iteración). Este carácter obriga a levar a cabo as fases de planificación, análise de requisitos, deseño, codificación, revisión e documentación en cada Sprint, o cal se se compara coas metodoloxías clásicas fai que tras un Sprint se poida axustar o seguinte en función de todo o aprendido, sen a necesidade dunha longa experiencia para poder planificar con exactitude.

Outra faceta importante das metodoloxías áxiles é a falta de documentación, isto está motivado polos sucesivos cambios que se producen nestes proxectos por mor do seu carácter adaptativo, e que serían moito más custosos se ademais de modificar o propio software fose preciso modificar longas listas de documentación asociada. Por este motivo, neste sistema: a análise de requisitos plasmarase como unha lista de tarefas que pasan ao Sprint Backlog, os diagramas de deseño só se elaborarán para as partes más críticas do sistema, no código evitaranse comentarios innecesarios e o plan de probas realizarase empregando unha folla excel en vez de un extenso documento de texto.

## 4.2. Persoas

Os tres papeis que se definen nesta metodoloxía [16] foron adaptados do seguinte xeito:

### 4.2.1. ProductOwner

O papel de ProductOwner, que define os requisitos da aplicación estivo representado polo director de proxecto Brais Cancela, que participou na creación do Anteproxecto. En certos momentos o señor Cancela tamén desempeñou a función de membro do equipo, posto que é foi autor do algoritmo de análise de vídeo.

### 4.2.2. ScrumMaster e Development Team

Ambos papeis leváronse a cabo polo autor, xa que carece de sentido definir ambas figuras nun equipo de unha única persoa. De este xeito á par que se desenvolvía o proxecto, íase asegurando o cumprimento das regras de SCRUM.

## 4.3. Reunións

As reunións pola súa parte modifícanse do seguinte xeito:

### 4.3.1. Sprint Planning Meeting

Esta reunión mantén o mesmo formato que no SCRUM orixinal, xuntando ao autor co ProductOwner e concretando as tarefas do Product Backlog que se realizarán no seguinte Sprint, pasando por tanto a formar parte do Sprint Backlog.

### 4.3.2. Daily Scrum

Dado que o equipo de Desenvolvemento e o ScrumMaster están conformados pola mesma persoa e que o número de horas diarias adicadas é moito menor que o de unha xornada laboral, considerouse oportuno substituír esta reunión diaria por unha reunión dúas veces á semana (Martes e Xoves pola tarde normalmente). Na que se mostrase ao ProductOwner o avance do proxecto.

### 4.3.3. Sprint Review

Esta reunión fusionase co Sprint Planning Meeting, xa que ao mesmo tempo valorase o traballo realizado no Sprint que remata e, en base a el, planificase a videira iteración.

### 4.3.4. Sprint Retrospective

Pola súa parte, esta reunión toma un carácter unipersonal, pasando a ser unha valoración do propio autor sobre as persoas, relacións, procesos e ferramentas implicadas no último Sprint. Nela avalíase os elementos con éxito e os suxeitos a melloras, creando un plan para implementar estas melloras na videira iteración.

## 4.4. Control de Versións con GitHub

GitHub é un sistema de control de versións que permite a xestión dos distintos cambios efectuados sobre un producto software ou sobre a súa configuración. Facilitando a administración das distintas versións do produto.

GitHub tamén é unha plataforma de desenvolvemento colaborativo que emprega o sistema de control de versións Git. Esolleuse empregar este sistema pola súa potencia e simplicidade, xa que proporciona libre acceso aos titores para comprobar o avance do proxecto, e a súa vez asegura que o código este sempre a bo recaudo.

A páxina do proxecto é: <https://github.com/iago-suarez/ancoweb-TFG>

#### 4.5. Integración Continua con Travis CI

A Integración Contínua (CI do inglés Continuous Integration) é un modelo informático que consiste en facer integracións automáticas dun proxecto o máis a miúdo posible para así poder detectar os posibles erros o antes posible, minimizando as súas posibles consecuencias. Outro factor importante é o feito de garantir que a versión subida ao repositorio segue a funcionar con independencia do entorno de desenvolvemento. Enténdense como pertencentes á integración continua a compilación e a execución das probas de todo un proxecto.

Travis CI é unha plataforma de integración continua para proxectos aloxados en GitHub, que detecta automaticamente cando se produce un cambio no repositorio, e executa unha serie de pasos definidos no ficheiro .travis.yml (figura 4.1), que contén as accións a realizar antes, durante e tras a as probas.

Esolleuse Travis CI, pola súa integración con GitHub, pola súa potencia (permite executar praticamente todo o que se pode executar nunha máquina local) e pola súa sinxela integración con outras ferramentas como Coveralls.

Pódese ver a páxina do proxecto en Travis CI en:

<https://travis-ci.org/iago-suarez/ancoweb-TFG>

#### 4.6. Control da cobertura con Coveralls

Coveralls é un servizo online que axuda a facer seguimento e control da cobertura de código para proxectos open-source aloxados en GitHub. É tremendalemente sinxelo de empregar e ademais está moi ben integrado con Travis CI, de xeito que cunha única liña de configuración pódese facer que o servidor de Travis CI envíe os resultados da

```
- cd
- curl -o ~/ffmpeg.tar.xz http://johnvansickle.com/ffmpeg/re
- tar xf ~/ffmpeg.tar.xz
- rm -rf ~/ffmpeg-*~static/{manpages,presets,readme.txt}
- mkdir ~bin
- ls -la
- cp ~/ffmpeg-*~static/* ~bin
- chmod 700 ~bin/{ffmpeg,ffprobe,ffmpeg-10bit,qt-faststart}
- cd && rm -rf ffmpeg{.tar.gz,-*~static}
- PATH=$PATH:~/bin/
- cd $PROY_DIR

# Config the xvfb server to selenium tests
- "export DISPLAY=:99.0"
- "sh -e /etc/init.d/xvfb start"
- "/sbin/start-stop-daemon --start --quiet --pidfile /tmp/cu
screen 0 1280x1024x16"

install:
- pip install -r requirements.txt
- pip install coveralls
before_script:
- cd src/RecognitionSystem
- make CONF=Debug
- cd ..
# command to run tests
script:
- python manage.py migrate
- python manage.py test -v 2
- coverage run --source='.' manage.py test
after_success: coveralls
```

Figura 4.1: Imaxe de parte do ficheiro .travis.yml

cobertura a Coveralls cando as probas se pasen correctamente.

Neste proxecto empregarase nun comezo para facer seguimento dos test realizados sobre a parte servidor inda que é posible que máis adiante se empregue tamén para probar o código que corre na parte cliente.

Pódese consultar a páxina de Coveralls para o proxecto en:

<https://coveralls.io/github/iago-suarez/ancoweb-TFG>

## 4.7. Xestión de Incidencias e Control de Proxecto con YouTrack

Co fin de levar a cabo un control das tarefas do Product Backlog e Sprint Backlog realizadas e pendentes empregarase YouTrack como Sistema de Xestión de Incidencias (Issue Tracking System). Un sistema de xestión de incidencias serve para xestionar as distintas incidencias (Tarefas pendentes, Bug's, Problemas de usabilidade ou rendemento...) que poidan ter lugar nun entorno como o proxecto web que nos ocupa. A estes efectos YouTrack mostrase como un completo sistema de incidencias que permite a estimación e xestión de tempos, o emprego de comentarios para cada incidencia, buscas avanzadas, filtrado de incidencias, etc.

Nótese que tamén se tiveron en conta outros sistemas de xestión de Incidencias como Redmine ou o sistema de xestión de incidencias integrado de GitHub, pero finalmente escolleuse YouTrack pola súa coherencia coa filosofía ágil que se pode ver nos seus Paneis Áxiles, neles pódense xestionar todas as incidencias dun Sprint mediante o modelo Kanban, simplemente desprazando unha tarefa da columna de Tarefas Abertas á de Tarefas en Curso ou de esta última á de Tarefas Solucionadas. A vista que YouTrack ofrece pódese ver na captura da figura 4.2.

Pódense consultar as incidencias do proxecto en YouTrack accedendo a:

[http://iago-suarez.myjetbrains.com/youtrack/issues?q=\\*](http://iago-suarez.myjetbrains.com/youtrack/issues?q=*>)

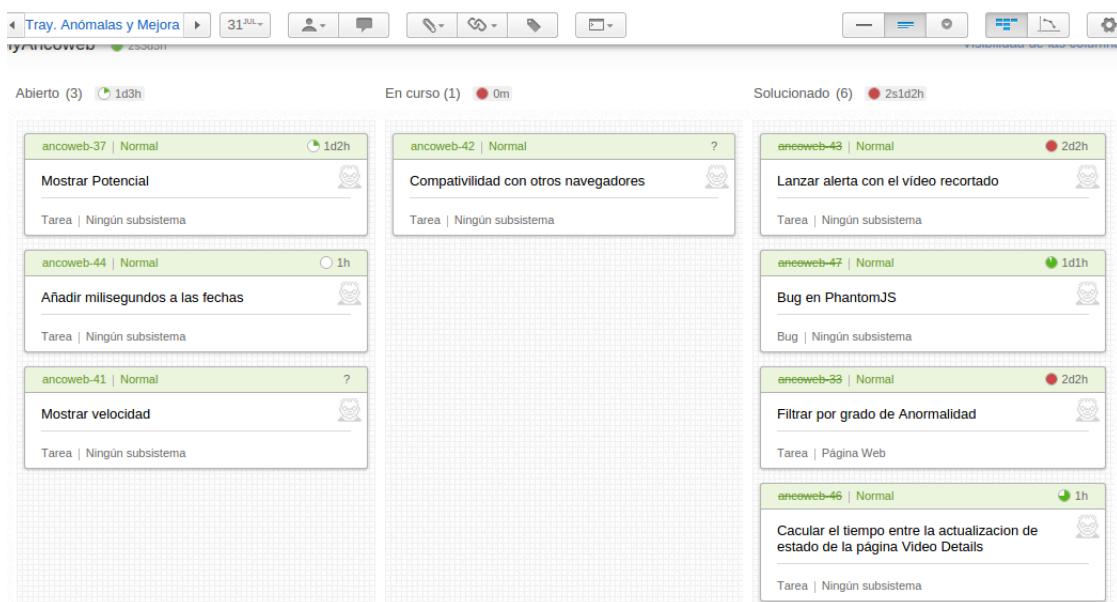


Figura 4.2: Panel Áxil da ferramenta de xestión de incidencias YouTrack



# Capítulo 5

## Tecnoloxías Empregadas

O feito de traballar na web, e moito máis o de facelo no ámbito da análise de vídeo, requieren que este sexa un proxecto cuns altos niveis de integración no que toman parte toda unha serie de librerías e ferramentas software que axudan a alcanzar os fins desexado.

Nesta primeiras sección procederase a explicar que tecnoloxías se tiveron en conta á hora de escoller o xeito de implementar este proxecto e con que criterios, para a continuación proceder cunha explicación detallada de cada unha de elas.

### 5.1. Estudo comparativo das tecnoloxías web

Para a elaboración de este traballo de fin de grao, é preciso seleccionar unha serie de tecnoloxías tanto para o lado servidor coma para o lado cliente, vamos a empregar os seguintes criterios para poder comparalas entre si e escoller entre elas a que mellor se adecúan ás necesidades do proxecto.

- **Plataforma e Portabilidade**

Segundo as especificacións iniciais do proxecto, e de cara a facilitar a implementación deste, as ferramentas empregadas deben de executarse baixo os Sistemas Operativos baseados en Linux.

- **Compatibilidade co algoritmo de Procesado de Vídeo**

Dado que o algoritmo que procesa o vídeo foi parcialmente implementado en

C++, a tecnoloxía que se empregue para o desenvolvemento da parte web debe dispoñer da máxima compatibilidade con esta linguaxe de programación.

#### ■ Desenvolvemento Áxil

A tecnoloxía que se seleccione ten que minimizar o custe en tempo e esforzo da implementación, por este motivo valorarase positivamente que dispoña de IDE's axeitados, facilidades de acceso a BD(Base de Datos),se é posible ORM(Object-Relational Mapping) integrado, recarga en quente... En xeral todo aquello que permita axilidade e flexibilidade.

Posto que empregamos unha arquitectura baseada no modelo cliente-servidor, temos que determinar por un lado en que tecnoloxías vamos a construír o Servidor ou Back-End, e por outra parte o Cliente ou Front-End que se executará nun navegador.

##### 5.1.1. Back-End

En canto ás tecnoloxías para a elaboración do Back-End, hai que ter en conta que procesará os datos proporcionados polo Front-End, atendendo as súas peticóns e xestionando o modelo de datos e os procesos implicados na aplicación. A maiores neste caso en concreto, o Back-End será o encargado de interactuar directamente co sistema de Análise de Vídeo.

As tecnoloxías estudiadas para esta parte do sistema son as seguintes:

##### Java

Java é unha das linguaxes máis empregadas actualmente. Ademais existen diversos frameworks web como Tapestry ou SpringMVC e en canto as BD Hibernate, que facilitan o seu uso, mais é preciso integralos xa que non forman parte da plataforma en si. As súas posibilidades de integración con C++ son altas grazas á interface JNI (Java Native Interface), pero a súa configuración pódese volver tediosa. Un dos proxectos estudiados para ver o seu funcionamento é Red5 [17].

## C#

Esta linguaxe en combinación con .NET resulta unha combinación bastante áxil de cara á programación web, integrando na propia plataforma un deseñador Web e un ORM moi intuitivo. O gran problema polo que se descartou este entorno foi polo seu baixo grao de compatibilidade cos sistemas operativos Linux.

## C - C++

C++ presenta como era de esperar a maior compatibilidade co algoritmo implementado, non obstante, inda que existen algunas utilidades que facilitan o desenvolvemento web con esta linguaxe como Wt (Web Toolkit)[18], o grado de axilidade está moi por baixo do que facilitan o resto das combinacións. Tamén pode resultar de interese o coñecido proxecto Icecast[19], que fai streaming de vídeo sobre unha interface web.

## Python + Django

Python presentase como a mellor opción para desenvolver o lado servidor, por unha parte dispón do módulo Subprocess[20] que permite executar calquera comando pola terminal maximizando así a modularidade e a integración co algoritmo en C++. Por outra parte Django[21] contén un potente ORM e un sistema de “Templates” que simplifica a parte web. Para concluír cabe destacar o feito de que python sexa unha linguaxe interpretada, xa que isto evita o paso previo de compilación.

### 5.1.2. Front-End

Unha vez seleccionada a tecnoloxía do lado servidor, é hora de ver que opcións existen para o Front-End, a parte do Sistema encargada da interacción co usuario.

Por unha parte están as tecnoloxías que pola súa transcendencia e nivel de aceptación cosideranse xa imprescindibles no desenvolvemento web, estamos a falar de HTML e CSS linguaxes de facto para definir o contido e o aspecto visual respectivamente dunha páxina web. De estas linguaxes seleccionaremos as súas versións máis recentes, que a día de hoxe son HTML5 e CSS3.

Sen embargo, noutros campos coma son a reprodución de vídeo e o control de elementos dinámicos non existe unha tecnoloxía que abarque a case completitude da rede, é por elo que neste capítulo estudaremos aqueles xeitos que permitan a reprodución de vídeo e a maiores o debuxado de figuras en movemento sobre este vídeo.

En canto á reprodución de vídeo e o control deste destacan principalmente dúas alternativas:

### Vídeo en Flash

Vídeo Flash é a tecnoloxía de reprodución de vídeo máis empregada e madura en internet dende hai anos. Inicialmente creada por Macromedia e mercada por Adobe en 2005, permite crear elaboradas animacións vectoriais, que logo poden proxectarse sobre un vídeo, mentres que tamén manexa os eventos de reprodución de vídeo como o play ou o stop.

Ten certos problemas en tanto ao Posicionamento Web(SEO), reprodución en dispositivos móbiles, accesibilidade... pero o meirande de todos eles é que mentres que outros dos exemplos estudiados son 100 % libres, Flash é un programa propietario para o que é preciso adquirir unha licencia.

### Vídeo HTML5 + Javascript

Esta é outra das combinacións máis empregadas actualmente, xa que segue o estándar do W3C (World Wide Web Consortium)[22] no que se define como se han de mostrar e obter os vídeos dunha páxina codificada coa linguaxe HTML5., e destaca por seres extremadamente sinxelo en comparación con Flash ou outras tecnoloxías.

Se o comparamos con Flash podemos ver a seguintes **vantaxes**:

- Resulta moito máis sinxelo de codificar grazas a que é o navegador o que se encarga da reprodución do vídeo mentres que o programador só define o xeito de obtelo.
- Non precisa da instalación de ningún Plugin que poida dar problemas por exemplo en dispositivos móbiles.

- Mientras que HTML5 e Javascript son libres, Flash é unha tecnoloxía propiedade de Adobe.
- HTML5 + CSS3 dispón de máis facilidades se buscamos un deseño “responsive”.

Será por tanto a alternativa escollida para a construcción deste proxecto, e en cuanto ao debuxo de figuras sobre o vídeo escoilleremos o elemento `< canvas >` tamén de HTML5. Todas estas tecnoloxías e moitas máis explícanse en detalle no seguinte capítulo.

A continuación explicase detidamente cal é a función de cada unha das tecnoloxías que forman parte deste proxecto co fin de comprender o explicado nos capítulos seguintes. O diagrama 5.1 pode servir como orientación á hora de comprender que función desempeña cada unha de elas.

Comezamos polas tecnoloxías empregadas no lado cliente:

## 5.2. Navegador Ciente

O navegador do cliente empregará unha serie de tecnoloxías para mostrar os datos obtidos do servidor, as más destacables son:

### 5.2.1. HTML5

Á hora de representar o contido dunha web precisase unha linguaxe que defina este contido, estruturándoo e facendo que sexa mais flexible e comprensible para que o navegador poida representalo en pantalla, esta linguaxe é HTML.

HTML (HyperText Markup Language) é a linguaxe de marcas empregada en internet para a elaboración de páxinas web, define unha estrutura básica e un código para a definición do contido da páxina como poden ser texto, imaxes, vídeos... Existen diversas versións deste estándar, mais para este proxecto empregarase a súa última versión HTML5, que inclúe toda unha serie de novos elementos.

Un dos componentes que máis empagaremos para a construción desta web será o elemento `< video >` de HTML5 que achega unha serie de Métodos, Eventos e Propiedades [23] que poden empregarse dende o código Javascript.

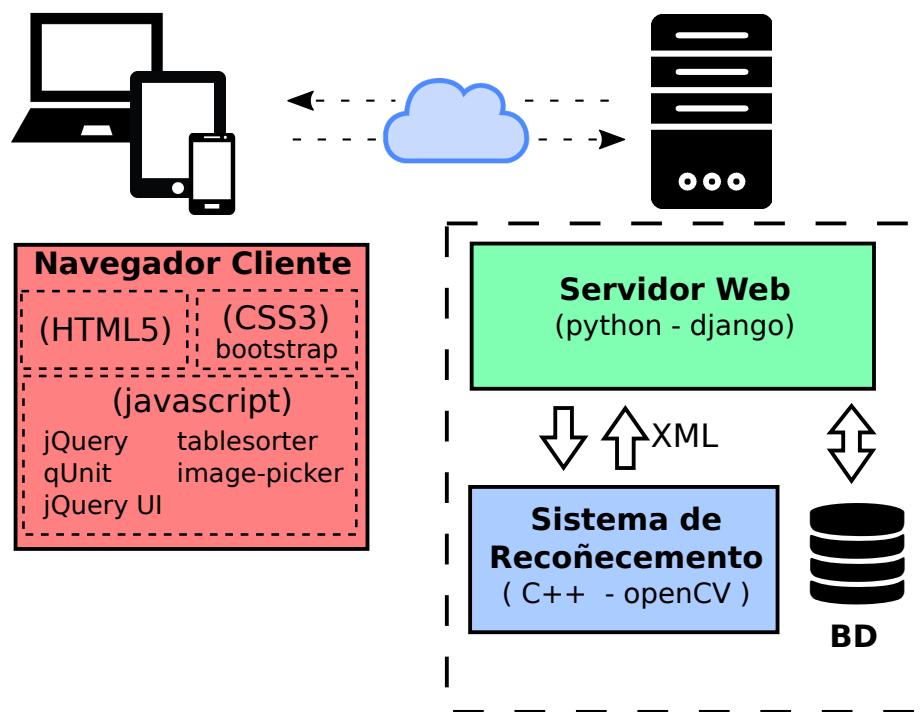


Figura 5.1: Diagrama da Arquitectura do Sistema

A maiores do propio contido da páxina, HTML permite incluír referencias a outros ficheiros que están tamén asociados á páxina como poden ser ficheiros javascript ou css que se mostran de seguido.

### 5.2.2. CSS3

HTML permite unha perfecta estruturación dos elementos que componen unha páxina web, pero en caso de que queiramos personalizar o estilo da páxina web, ou adaptala a diferentes dispositivos precisamos CSS. CSS(Cascading Style Sheets) é unha linguaxe para definir e crear a presentación dun documento HTML.

Para elo defínense unha ou máis follas de estilos, que definen para cada elemento seleccionado nelas unha serie de características de estilo. No caso da aplicación a desenvolver a política de follas de estilo será a de crear unha folla de estilo xeral para conter as características de estilo comúns a todo o proxecto e a maiores as que sexan precisas para páxinas ou elementos concretos, todo elo traballando coa versión 3 desta linguaxe.

### Twitter Bootstrap

CSS3 permite facer todos os axustes precisos en canto a estilo web, pero o problema é que elaborar un bo estilo leva unha cantidade de tempo cuantiosa e por ese motivo se precisa algunha libraría que marque unha pauta de estilo uniforme e que minimice o traballo a realizar, bootstrap mostrase como unha das librarías mais empregadas a estos efectos e por iso foi seleccionada para este proxecto.

Twitter bootstrap é un framework ou conxunto de ferramentas de código aberto para deseñar web's. Contén todo un conxunto de plantillas como tipografía, botóns, formularios, táboas, barras de navegación... Estas plantillas normalmente consisten simplemente nun estilo CSS, pero ás veces tamén hai un procesado en javascript como o da función popover que permite xerar unha ventá flotante para amosar os datos dos obxectos detectados na aplicación.

### 5.2.3. Javascript

A pesar de todas as cousas que permiten facer HTML e CSS moitas veces é preciso un tratamento dinámico da información para desprazar un elemento na páxina, atender un evento clic, lazar unha ventá... e neste caso é preciso o emprego dunha linguaxe de scripting como javascript.

Javascript é a linguaxe de programación interpretada que se executa nos navegadores cumplindo co estándar ECMAScript, permite executar ordes podendo interactuar coa páxina web a través da arbore DOM (Document Object Model) e co servidor mediante o sistema de chamadas asíncronas. Cada script en javascript está asociado a unha páxina web, de forma que todo o que executemos terá efecto sobre a páxina actual e nunca poderemos prolongar a execución dun código .js máis aló da vida desta páxina.

A pesar de executarse nun navegador javascript é unha linguaxe cunha potencia considerable, esta potencia ven da súa orientación a obxectos que tamén permite programación imperativa con un tipado débil e dinámico.

Javascript pode incluirse directamente no documento HTML, mais isto é problemático porque mestura dúas linguaxes e non permite a reutilización do código javascript en distintas páxinas, polo que na practica sempre se traballará con código contido en ficheiros .js que logo serán importados dende as páxinas web que o precisen. Mediante referencias tamén se pode engadir código de librarías, que pode achegarse de dous xeitos distintos, ou ben cun enlace á páxina onde están publicadas, ou ben descargando estas librarías ao servidor da aplicación e ofrecéndoas dende aí. O enfoque seguido neste proxecto é o segundo, pois de este xeito a aplicación está auto-contida, podendo traballar sen conexión con internet, e mantendo sempre a integridade a pesar de que o provedor da biblioteca decida deixar de ofrecela.

Mediante Javascript tamén pode manexarse o elemento `<canvas>` de HTML5, que xera un mapa de bits para construír gráficos, manipular imaxes e crear dinamicamente animacións nunha páxina web. A única dúbida que soe xurdir sobre esta tecnoloxía

está en canto ao seu rendemento e alcance, pero exemplos como os que amosa Kevin Roast na súa páxina web[24] despexan toda dúbida posíbel.

Por desgraza, a execución en navegador ten tamén os seus inconvintes como a execución multi-threading empregando **Web Workers**, estes elementos pensados para permitir a execución paralela de código javascript deixan polo de agora moito que deixar xa que cada thread ten as súas propias variables estancas, impedindo polo tanto o acceso á arbore DOM dende threads paralelos co problema engadido de que algúns navegadores como Opera en vez de facer un multi-threading sobre threads do propio sistema tan só simulan este fenómeno nun único thread. O paralelismo cobrará especial importancia cando executemos os algoritmos que mostran os datos da análise en XML, pois ao executarse todo no mesmo fio debemos prestar especial atención a non bloquear a interface de usuario con tarefas moi prolongadas.

## jQuery

Javascript tamén é especialmente potente á hora de parsear documentos, xa sexa un documento HTML para acceder á arbore DOM ou ben un XML como o que xera o sistema de análise. Por desgraza, o manexo de excepcións, e a selección de elementos dentro de un documento son tarefas que se farán de forma moi habitual neste traballo e programar estas tarefas con javascript resulta tremendalemente laborioso, así que co fin de simplificar esta tarefa empregarase a biblioteca jQuery.

jQuery é unha biblioteca javascript pensada para simplificar os aspectos más complexos desta linguaxe como poden ser a manipulación de documentos HTML/XML ou as chamadas asíncronas mediante AJAX. Está escrita en javascript e é 100 % libre, tal vez por isto sexa a biblioteca javascript máis empregada.

O sistema de selección de jQuery é o mesmo que se emprega en CSS, permitindo seleccionar de xeito sinxelo un conxunto de elementos dentro dun documento, a maiores tamén dispón de un sinxelo acceso e modificación tantos destes elementos como dos seus atributos.

## **Qunit**

Outra faceta tamén moi laboriosa en javascript é a de probar o código escrito e xa que a meirande parte desta aplicación estará escrita en javascript faise imprescindible verificar que o código cumple coa súa función. Por este motivo escolleuse outra biblioteca tamén deseñada polo equipo de jQuery para simplificar os tests en javascript, esta biblioteca chamase Qunit.

QUnit é un framework de probas unitarias para código javascript doado de emplegar e bastante poderoso. Empregase tanto en jQuery, jQuery UI e os proxectos de jQuery Mobile sendo capaz de probar calquera código javascript xenérico.

Para facer probas emprega un conxunto de sentencias assert como todas as bibliotecas que realizan tests de unidade. No caso da nosa aplicación empregarase para probar todo aquel código independente da arbore DOM da páxina. É importante destacar que co fin de maximizar a facilidade de proba do código na capa javascript seguíronse os consellos amosados na páxina de Qunit[25] e unha interpretación flexible do MVC(Modelo Vista Controlador) onde a vista está conformada polo código HTML+CSS, o controlador polo manexadores dos ficheiros video-player.js, video-controls.js e por último as clases do modelo en javascript nos ficheiros Detection.js, DetectionObserver.js, Video-Detections.js e en menor medida suspicious-popup.js.

## **Image-picker**

Durante a construción da aplicación, nunha das páxinas xurdiu a necesidade de seccionar entre unha lista de imaxes, pero HTML5 é unha linguaxe que por unha parte permite amosar imaxes e por outro lado á hora de enviar información ao lado servidor permite enviar datos moi simples mediante un formulario. Para dotar a HTML5 de esta característica empregouse a libraría baseada en jQuery Image-Picker[26] que permite xerar un formulario cun campo de tipo “select” baseado en imaxes en vez de nunha entrada despregable, todo elo de forma extremadamente sinxela.

### jQuery UI

En outro momento da construcción viuse a necesidade de engadir unha barra selec-tora para seleccionar un determinado valor na interface gráfica, e para elo optouse polo plugin slider[27] da biblioteca jQuery UI. Esta biblioteca de compoñentes para jQuery engádelle un conxunto de plugins, widgets e efectos visuais. Pódese descargar dende a súa páxina oficial o núcleo da biblioteca e os compoñentes nos que se estea interesado.

### tablesorter

Como se verá mais adiante, os requisitos da aplicación farán preciso que nunha das páxinas webs se amose unha táboa e que esta táboa se poida ordenar segundo o con-tido de distintas columnas. A estes efectos incorporase a derradeira libraría javascript chamada tablesorter[28]. Tablesorter é un plugin baseado en jQuery para transformar unha táboa HTML estándar coas etiquetas *<thead>* e *<tbody>* nunha táboa que se pode ordenar polo contido das distintas columnas sen ter que recarga-la páxina. Neste caso será de moita utilidade á hora de amosar os resultados da análise, pois así pode-ranxe ordenar as deteccións segundo aparecen no vídeo, segundo o tempo que pasan en pantalla...

Todas estas tecnoloxías de capa web axudarannos a mostrar con máis facilidade a análise que o sistema de recoñecemento faga do vídeo, e en canto as ferramentas empregadas para esta análise a ferramenta fundamental que se emplegará é OpenCV.

## 5.3. Sistema de Recoñecemento

O sistema de recoñecemento debe ser capaz de ler un vídeo e en base a el recoñecer os obxectos de ese vídeo á vez que analiza o seu comportamento. Isto pódese implementar de distintos xeitos, mais neste proxecto en concreto fíxose empregando a linguaxe de programación C++ que xa se citou na sección 5.1, en combinación coa librería OpenCV e a linguaxe de marcas XML.

### 5.3.1. OpenCV

OpenCV é unha biblioteca libre de visión artificial escrita en código C/C++ optimizado. Dende a súa aparición publicada por Intel en Xaneiro de 1999, empregouse en infinidade de proxectos, tanto para detección de movemento como para aplicativos de control de procesos que requieren recoñecemento de obxectos.

OpenCV é multiplataforma, existindo versión para GNU/Linux, Mac OS X e Windows. Contén máis de 500 funcións que abarcan unha ampla gama de áreas como o proceso de visión, recoñecemento de obxectos (tamén recoñecemento facial), calibrado de cámaras, realidade aumentada e visión robótica.

Por todas estas características OpenCV é unha das bibliotecas más empregadas hoxe en día, e é por elo tamén que se escolleu para a implementación do sistema de recoñecemento que a aplicación web empregará para a análise do vídeo.

Non obstante, xa que se desexa que a aplicación web sexa o máis versátil posible, contemplase a posibilidade de que poida empregar para a análise sistemas desenvoltos noutras tecnoloxías como pode ser Matlab. E para dotala deste grao de versatilidade decídese definir unha interface de liña de comando a través da cal se chamará ao sistema, e un formato de ficheiro XML (Extensible Markup Language) no que este sistema de recoñecemento deben escribir os datos da súa análise.

### 5.3.2. Extensible Markup Language (XML)

Como os datos que devolve o sistema de recoñecemento teñen que ser gardados nun ficheiro de xeito que calquera sistema poida entendelos ou traballar con eles, precisase un formato para ese ficheiro de saída que sexa capaz de almacenar datos estruturados de forma sinxela e comprensible, este formato é XML.

XML é unha linguaxe de marcas desenvolvida polo W3C (World Wide Web Consortium) e empregado para almacenar datos de forma clara e flexible. Permite definir a gramática de linguaxes específicas para estruturar así grandes documentos.

Os documentos XML seguen una estrutura xerárquica baseada en etiquetas(tag's) e atributos, que se poden definir nunha Definición de Tipo de Documento ou DTD. [29]

Cando un documento en formato XML segue as directrices definidas no ficheiro

DTD asociado, dise que este ficheiro esta ben formado(well formed en inglés), e para validar iso empregarase na elaboración do traballo algúns avaliador de XML en liña como por exemplo o da W3Chools.[30]

XML é especialmente útil para comunicar varias aplicaciones que traballan en tecnologías diferentes grazas á súa simplicidade que permite integrar os datos de xeito moi sinxelo. Precisamente por iso, servirá como nexo de unión entre o sistema de análise e a aplicación web.

Para a modificación da súa apariencia pódese emplegar *CSS3* (Follas de Estilos en Cascada), e para modificar o seu comportamento, o elemento *<video>* de HTML5 achega unha serie de Métodos, Eventos e Propiedades[23] que poden emplegarse dende código *Javascript*.

## 5.4. Servidor Web

Como se explicou anteriormente na sección 5.1 Django é un framework que traballa sobre python para a creación de páxinas web, as súas funcionalidades emplegadas explicaranse pouco a pouco no capítulo 6, polo que aquí explicaranse outros compoñentes do servidor emplegados que compre coñecer antes de proseguir, en concreto ffmpeg.

### 5.4.1. FFmpeg

Ao tratarse a nosa aplicación de unha web centrada principalmente no traballo con vídeos, é preciso dispor de unha ferramenta capaz de proporcionar datos sobre ese vídeo como poden ser a súa duración, o seu número de fotogramas por segundo... e tamén capaz de transformar vídeo de un formato a outro. O software empleado para esta labor no proxecto é FFmpeg.

FFmpeg é unha colección de software libre que pode gravar, converter (transcodificar) e facer streaming de audio e vídeo, nel está incluído libavcodec unha biblioteca de codecs que contén aqueles máis emplegados. Pódese dicir ademais de el, que a pesar do seu grandioso número de opcións é un programa bastante sinxelo de emplegar.



# Capítulo 6

## Desenvolvimento

### 6.1. Funcionalidades desexadas

Antes de comenzar o ciclo de sprint's tivo lugar unha pequena fase de análise no que se requiriron as seguintes funcionalidades do proxecto a realizar:

- **Control de Usuarios:** É preciso un mínimo control de usuarios para controlar quen sube vídeos á plataforma.
- **Carga de Vídeo:** Dado que a aplicación traballará con vídeos subidos polos usuarios, o primeiro paso é lograr a subida exitosa de vídeos á plataforma por parte de usuarios autenticados.
- **Reproducción de Vídeo:** Desexase que a aplicación permita a reproducción dos vídeos contidos, mediante técnicas de streaming ou pseudo-streaming.
- **Lista de vídeos e Imaxe de Portada:** É preciso amosar de algúin modo os vídeos subidos á plataforma, a poder ser cunha imaxe representativa que axude a identificalos e con algúin mecanismo de busca que permita filtralos en caso de que haxa un número elevado de eles.
- **Análise de Vídeo:** A aplicación web debe facer uso de dúas librerías de computación visual proporcionadas polo laboratorio VARPA para analizar os vídeos subidos e extraer de eles os obxectos detectados e unha análise do seu comportamento.

mento, como é lóxico só os usuarios autenticados porán facer isto xa que só eles poden subir vídeos á aplicación.

- **Mostrar Deteccións:** Devénse amosar aqueles obxectos detectados mediante unha capa de información extra sobre o vídeo subido.
- **Traxectorias:** Tamén se han de indicar con unha capa de información sobre o vídeo as traxectorias seguidas polos obxectos detectados.
- **Listar Deteccións:** É importante listar os obxectos detectados en todo o vídeo coa súa información asociada, así como listar tamén aqueles que están en escena nun momento determinado.
- **Detección do comportamento anómalo:** Desexase que empregando as funcionalidades da biblioteca proporcionada se analice o comportamento dos obxectos involucrados nunha escena para así poder filtralos segundo o estranho que resulte este comportamento.
- **Popups de deteccións sospeitosas:** Outra funcionalidade requirida é que cuando un obxecto teña un comportamento sospeitoso respecto ao do resto de obxectos, se abra unha nova ventá que o siga para poder ver máis de preto o que está a facer.

O desenvolvemento da aplicación tivo lugar seguindo a filosofía de SCRUM, polo que o enfoque más correcto neste caso para ver como se cumpriron todas as funcionalidades anteriores é o de explicar o traballo acometido en cada un dos sprint's.

## 6.2. Primeiro Sprint: Cargar e Visualizar vídeos

Neste primeiro Sprint a parte de un cuantioso número de horas á formación no framework de Django e no traballo con GitHub, creouse o esquema básico sobre o que se partirá para a creación de toda a web. Este esquema está baseado no plantilla Edge[31], que proporciona un control mínimo de usuarios, unha liña de estilo, integración con bootstrap e control de administrador, inda que posteriormente a maioría destes aspectos tiveron que ser modificados ou ampliados.

A partir desta plantilla que xa proporciona o control de usuarios, as dúas primeiras funcionalidades a implementar foron a Carga dos vídeos e a súa reproducción.

### 6.2.1. Carga de Vídeo

Dado que a aplicación traballará con vídeos subidos polos usuarios, o primeiro paso é lograr a subida exitosa de vídeos á plataforma. Para este fin empregarase un formulario HTML que viaxa sobre unha chamada POST de HTTP (figura 6.1). Cando o navegador faga esta chamada incluíndo o vídeo como parte do formulario, este vídeo comenzará a subirse ao servidor en pequenos anaquiños (data chunk).

## Upload New Video

**Title:**

**Video:**

**Description:**  

Este é un vídeo no que se ven varios coches circulando pola rua

Figura 6.1: Formulario para a creación de vídeos

É de especial importancia que no caso de que o vídeo teña un peso considerable e precise duns segundos para subirse á plataforma, o usuario poida coñecer de forma gráfica o avance deste proceso.

Con este fin, crease un sistema de notificación de progreso baseado no django-

progressbarupload [32], este sistema apoiase nunha componente fundamental chamada VideoUploadHandler, que é unha extensión da interface de Django TemporaryFileUploadHandler [33], e que basicamente manexa a subida dun ficheiro de tamaño considerable.

Esta componente compónse dunha función de inicio (`handle_raw_input`) que crea unha entrada na Cache de Django, almacenando como chave un número aleatorio e a IP do cliente que está a subir o vídeo, e como valor o tamaño do ficheiro e o porcentaxe de este que xa está subido. Esta entrada será actualizada cada vez que o servidor reciba un novo anaco de vídeo (mediante a outra función do componente VideoUploadHandler, `receive_data_chunk`).

Por outra parte, para que visualmente o cliente poida ver o avance da subida a través unha barra de progreso, crease unha función asíncrona en javascript (Tecnoloxía AJAX), que periodicamente consulta ao servidor para obter o valor da cache que indica a porcentaxe de subida do vídeo, e unha vez obtido, actualiza a barra de progreso para mostraloo. Todo isto ten lugar no navegador mentres este inda está a subir o arquivo de vídeo como indica o diagrama 6.2.

Unha vez que a subida se completa, o POST é manexado pola vista UploadView, que se todos os datos do formulario son correctos, encargase de crear un modelo VideoModel. Como parte desta creación o vídeo pasa do directorio temporal no que foi almacenado (baixo linux por defecto é `/tmp`) a un directorio calculado pola función `get_valid_filename`. Esta función pásaselle ao modelo como parte do seu campo "video" do tipo VideoField que ademais comproba que o tamaño de vídeo non sexa excesivo. Todo este proceso pódese ver de xeito máis claro no diagrama 6.3.

Unha vez subido o vídeo satisfactoriamente agora debese reproducir ese vídeo subido nunha páxina adicada.

### 6.2.2. Reprodución de Vídeo

Desexase que a aplicación permita a reprodución dos vídeos contidos, mediante técnicas de streaming ou pseudo-streaming, neste caso empregarase o pseudo-streaming polo sinxela que resulta esta implementación empregando as capacidades da etiqueta

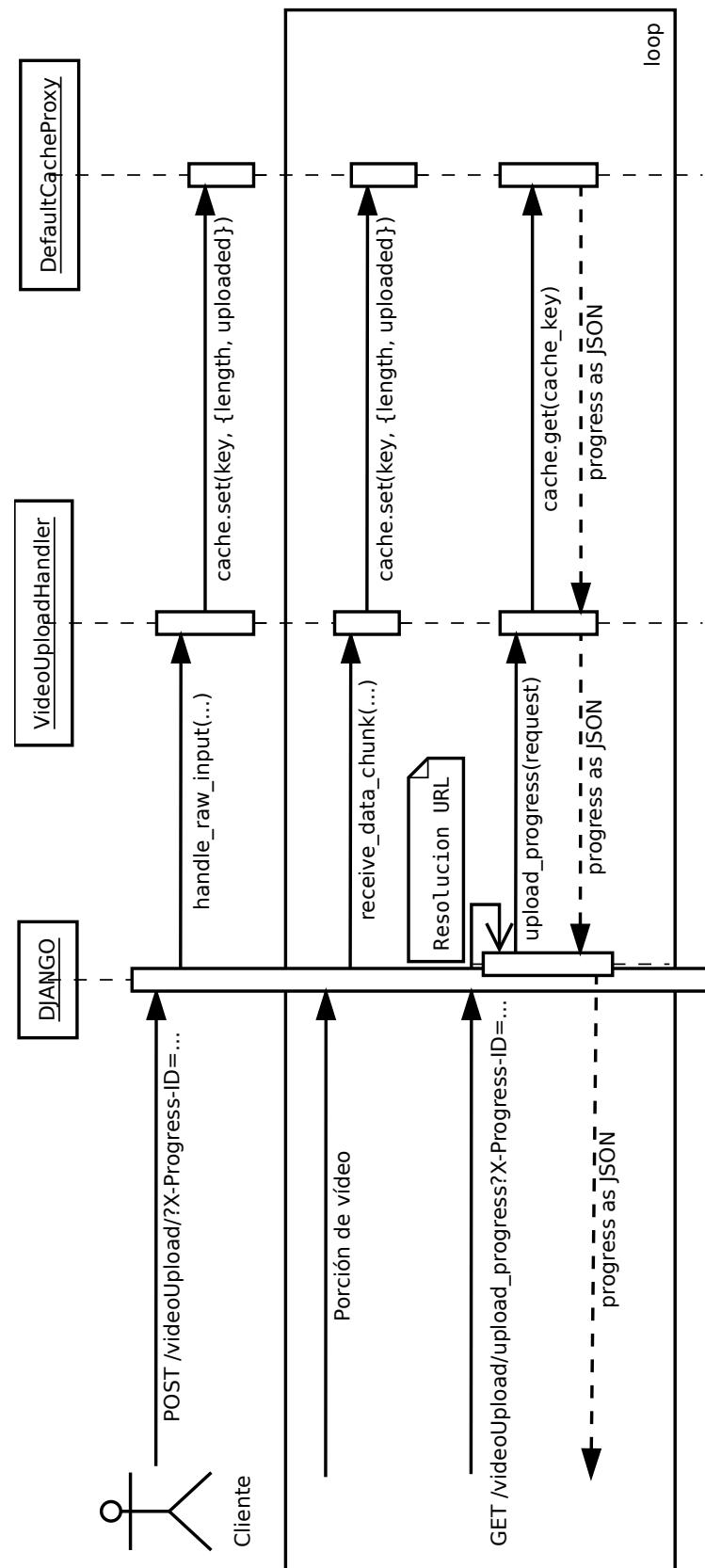


Figura 6.2: Diagrama de secuencia do proceso tras o envío do formulario de creación de vídeos (1)

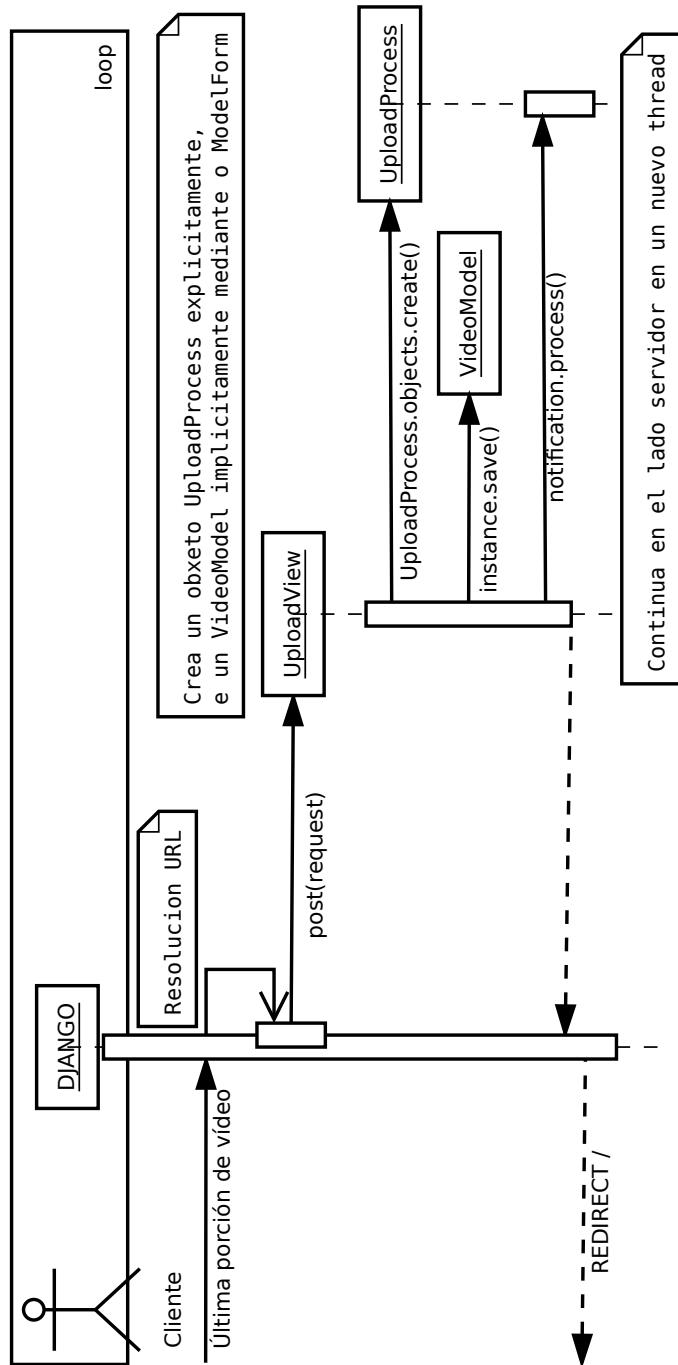


Figura 6.3: Diagrama de secuencia do proceso tras o envío do formulario de creación de vídeos (2)

<video> de HTML5 en conxunto con un servidor HTTP como Apache ou o servidor para desenvolvemento de Django.

```

► <div class="col-sm-4 col-md-3">..</div>
▼ <div class="col-sm-8 col-md-6">
  ▼ <video id="video-player" controls>
    <source src="/media/videos/2/v_7760.mp4" type="video/webm" fps="10">
    <source src="/media/videos/2/v_7760.webm" type="video/webm" fps="10">
    "Your browser does not support the video tag."
  </video>
  <canvas id="training-canvas" class="drawing-layer" width="462" height="347" style="top: 7729px; left: 1008.5px; padding-left: 0px; padding-top: 0px; display: none;">
  <canvas id="objects-canvas" class="drawing-layer" width="462" height="347" style="top: 7729px; left: 1008.5px; padding-left: 0px; padding-top: 0px; display: none;">
  <canvas id="trajectories-canvas" class="drawing-layer" width="462" height="347" style="top: 7729px; left: 1008.5px; padding-left: 0px; padding-top: 0px; display: none;">
  <span id="xml_detected_objs" hidden> /media/xml/2/x199.xml </span>
</div>
► <div class="col-sm-12 col-md-3">..</div>
```

Figura 6.4: Etiqueta vídeo de html5, coas súas fontes e coas capas <canvas> asociadas

na figura 6.4 podemos ver o resultado en HTML5, vense claramente a etiqueta <video> coas súas fontes de datos <source>, cabe destacar que aquí engadiuse o atributo fps (Fotogramas Por Segundo do inglés Frames Per Second) que non pertence ao estándar definido pola W3C[34] mais no caso da nosa aplicación é fundamental para poder coñecer a velocidade á que o navegador vai amosar os fotogramas do vídeo.

Outra cuestión a aclarar é o motivo polo cal non se subministra a fonte de vídeo en formato .ogv que a W3C recomenda. A resposta é que o codec theora que ffmpeg inclue soporta decodificación pero NON codificación, polo cal é posible pasar de vídeos en .ogv a outros formatos pero non de outros formatos a .ogv imposibilitando pois que se poida ofrecer o vídeo neste formato mentres o codec de ffmpeg non o permita. Non obstante, isto non supón un problema, xa que como se pode ver na táboa 6.1 todos os navegadores permiten a reproducción baseándose nestes dous formatos.

### 6.3. Segundo Sprint: Subida e conversión de vídeos

Nesta segunda iteración, deséxanse acometer díus tarefas principais que teñen que ver coa usabilidade da aplicación. Por unha parte crear un listado de vídeos subidos á aplicación para poder acceder a eles de xeito ordenado, e por outra parte interesa

Navegador	MP4	WebM	Ogg
Internet Explorer	SI	NON	NON
Chrome	SI	SI	SI
Firefox	SI dende Firefox 21 (win) dende Firefox 30 (linux)	SI	SI
Safari	SI	NON	NON
Opera	SI dende Opera 25	SI	SI

Táboa 6.1: Táboa de formatos de vídeo soportados polos distintos navegadores

crear os mecanismos precisos para que se poidan levar a cabo os procesos de conversión do vídeos a distintos formatos, á análise do vídeo (que se implementará en futuras iteracións) e á extracción de imaxes para empregalas como imaxes representativas do vídeo, dunha forma fluída e estética.

### 6.3.1. Lista de vídeos e Imaxe de Portada

Co fin de que os usuarios accedan aos distintos vídeos subidos, deséñase unha páxina web que será a principal do módulo video\_manager, na cal un usuario pode visualizar unha **lista paxinada dos vídeos** dispoñibles. Esta lista estará ordenada começando polo vídeo máis recente e rematando polo máis antigo, tamén se contempla a posibilidade de poder filtralos por exemplo polo nome.

Para a elaboración de esta páxina empréganse o elementos dos que dispón Django como son a clase ListView co seu atributo paginate\_by e o filtrado de resultado co método QuerySet.filter(...), mentres que para o paso das palabras chave polas que buscar un vídeo empregase un parámetro de URL chamado “name”.

Outra funcionalidade interesante de cara a amosar unha lista de vídeos é a de poder mostrar unha imaxe representativa de cada un deles. Con este fin crease a vista SuccessfulUpload (imaxe 6.5), á que se redirecciona unha vez o vídeo é subido correctamente

para seleccionar a súa **imaxé de portada**.



Figura 6.5: Captura de pantalla da páxina web SuccessfulUpload

Con este fin extraíense mediante ffmpeg unha serie de fotogramas do vídeo, que se gardan nun directorio temporal para que unha vez o vídeo estea subido e analizado, as imaxes se integren como parte dun formulario na páxina SuccessfulUpload. Mediante este formulario, xerado co plugin image-picker[26], o usuario poderá escoller o fotograma que lle pareza máis representativo do vídeo e unha vez que pulse no botón de "Submit" este fotograma gardarase como parte do Modelo de Django VideoModel, quedando pois accesible para que a lista de vídeos poida amosalo.

### 6.3.2. Sistema de Notificacións

Cando deseñamos unha aplicación web é de capital importancia que o usuario este informado de que está a acontecer na aplicación para que non sinta que está perdido, ou que a aplicación non responde. Tendo isto en conta, e sabendo que tanto o proceso de análise(que inda non está implementado) coma o de conversión do vídeo a outros formatos poden requirir dun tempo prolongado, albíscase un problema: Como manter ao usuario informado destes longos procesos e evitar a sensación de bloqueo?

A solución deseñada é un sistema de notificacións que permite ao usuario rexistrado

navegar libremente pola aplicación mentres o vídeo se está a analizar, mostrando en todo momento unha barra de progreso para o proceso que se está a seguir nestes intres como se pode observar na imaxe 6.6.

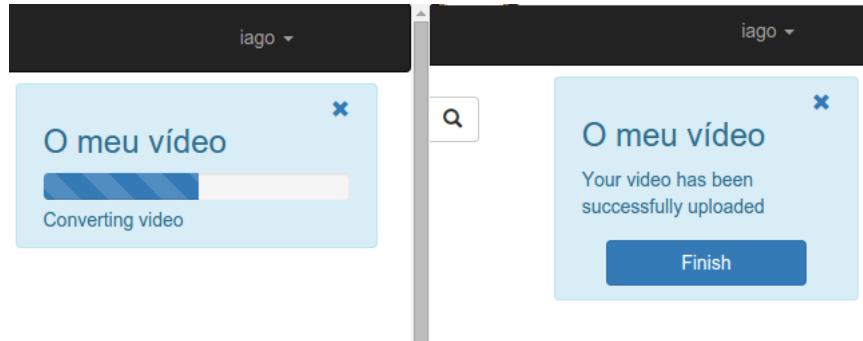


Figura 6.6: Capturas de pantalla do sistema de notificacións

Para albergar tanto o sistema de notificacións como os procesos que engloba, creouse o módulo de Django `video_upload` composto polas clases que se poden observar no diagrama <sup>1</sup> 6.7.

`UploadProcess` representa o proceso de subida, análise, conversión e extracción de imaxes a partir de un vídeo. Mientras que `AnalysisProcess` representa o proceso que se segue no caso de que un vídeo xa subido á plataforma sexa analizado de novo. Os distintos estados nos que pode estar un proceso modelanse mediante a clase abstracta `ProcessState`, que nas súas implementacións define tanto o traballo a realizar neste estado coma a mensaxe que se amosará ao usuario cando este se execute.

Dado que é o `ProcessState` que executará a tarefa, tamén será o encargado de actualizar a través do método `set_progress(self, progress)` o progreso do proceso asociado (`UploadProcess` ou `AnalysisProcess`).

É importante destacar que as figuras etiquetadas co estereotipo `<< Model >>` son

---

<sup>1</sup>Como se pode observar no diagrama 6.7, todos los métodos e atributos son públicos, isto es debido a que en Python no existen los miembros privados. No obstante aqueles atributos o métodos que no se aconseja emplear desde clases externas comienzan por `_`

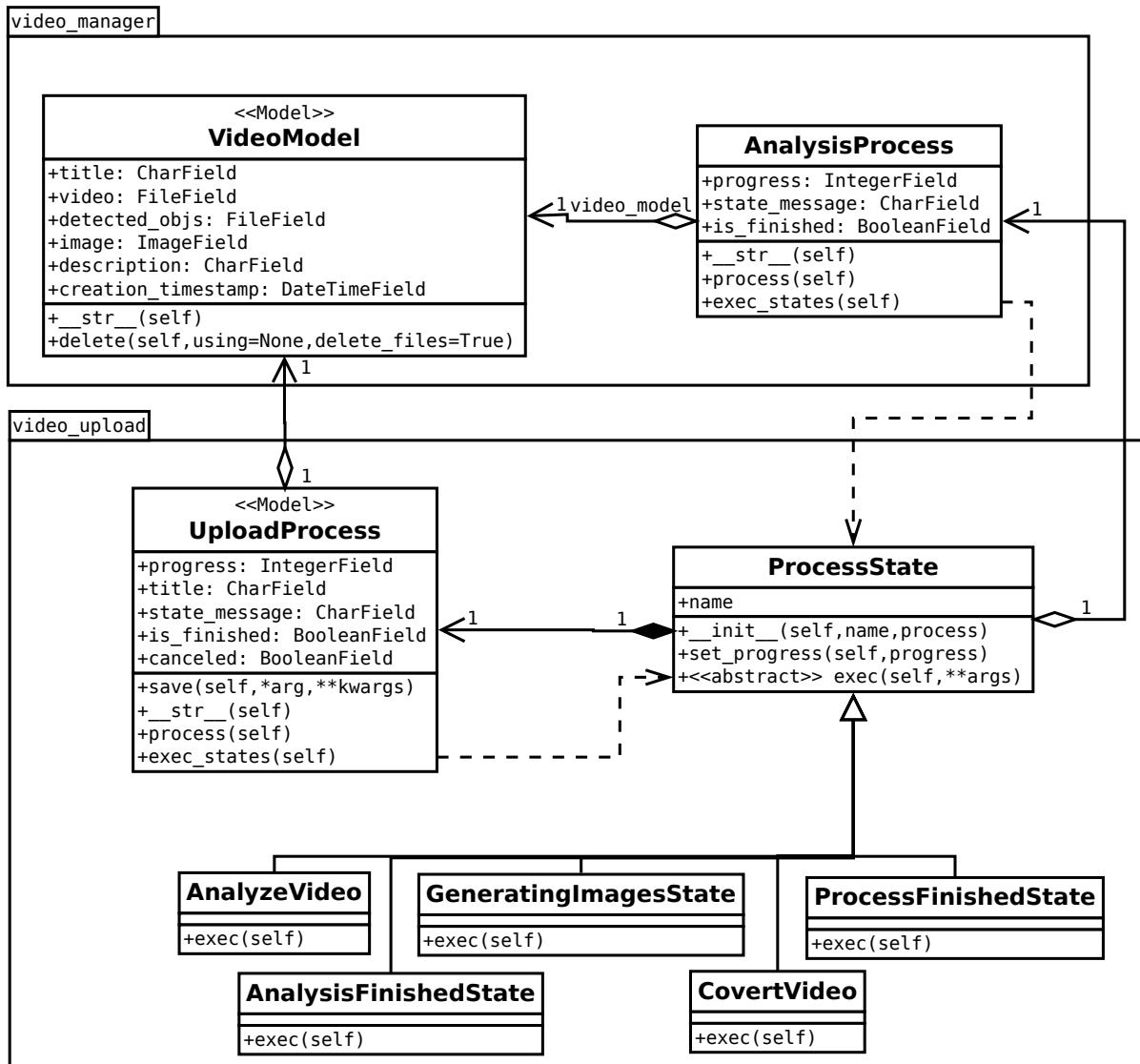


Figura 6.7: Diagrama de clases do sistema de subida

modelos de BD manexados por Django. Nótese tamén que pese a que UploadProcess e AnalysisProcess comparten a meirande parte do seu código, non foron refactorizados nunha clase abstracta, dadas as complicacións de base de datos que isto carrexa. En lugar diso empregase o tipado dinámico de Python para pasar os obxectos tanto de UploadProcess como de AnalysisProcess á clase ProcessState, que ao invocar só os métodos comúns non é capaz de percibir a diferenza entre ambas. O seu funcionamento mostrase no gráfico 6.8.

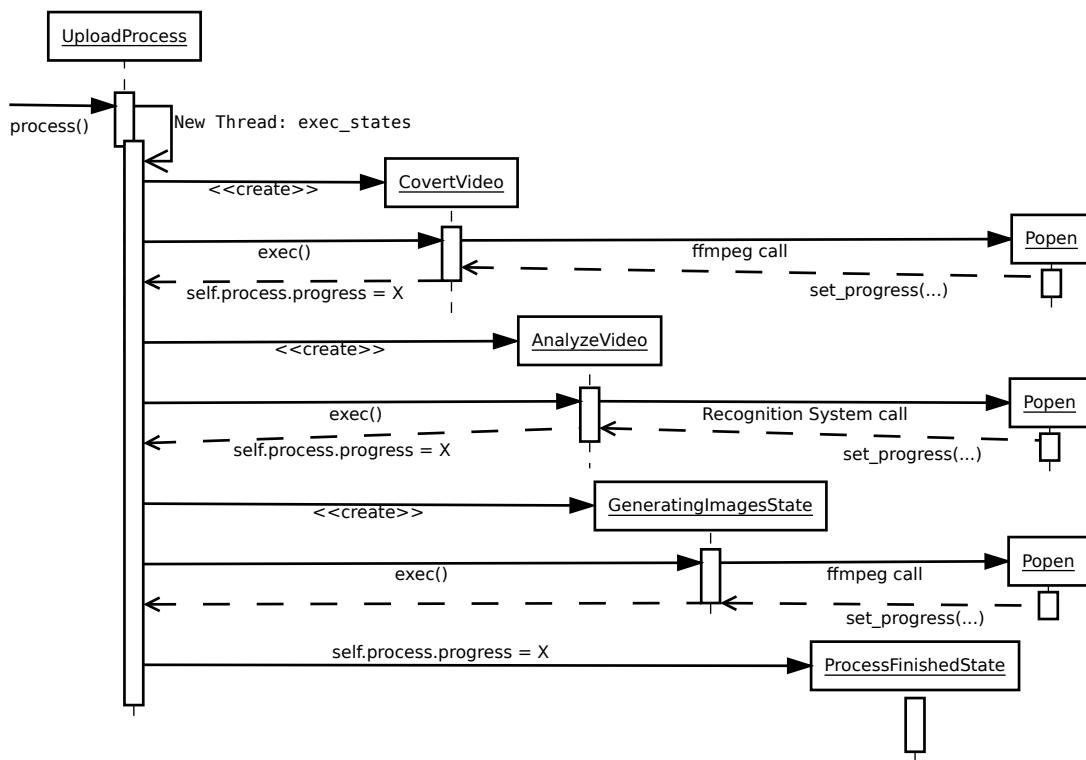


Figura 6.8: Diagrama de secuencia da análise de vídeo na capa web

#### 6.4. Terceiro Sprint: Xerar e mostrar deteccións básicas

En esta iteración tratouse tanto a análise do vídeo creando o sistema de recoñecemento, como o xeito de amosar os resultados destes na capa web, todo elo deseñado de xeito flexible e modulado como se explica a continuación.

#### 6.4.1. Análise de Vídeo

Para a análise de vídeo será preciso definir unha interface de liña de comandos mediante a cal a aplicación web chamará ao Sistema de Recoñecemento, indicándolle aqueles parámetros que sexan precisos 6.9.

#### Interface de Liña de Comandos

Estes parámetros que a aplicación web lle ten que proporcionar ao sistema de recoñecemento son: a ruta onde se atopa o vídeo que debe empregar como entrada para o recoñecemento e a ruta ao ficheiro de saída onde ten que escribir os datos da análise. A maiores, pódesele indicar cada canto fotogramas se desexa que o Subsistema Behaviour System, encargado da análise de alto nivel, entre en funcionamento. Po último a opción `--standar` fai que se mostre o resultado da detección de obxectos por liña de comandos, esta saída é empregada pola capa web para calcular o progreso do proceso de análise.

Estas opcións tamén se poden visualizar mediante o comando `--help`:

```
iago@UbuIago:~/TFG/src/RecognitionSystem$ ./recognitionsystem --help
Usage:   recognitionsystem
option:
-i       --input      <path/to/outputFile.xml> Set the input file.
-o       --output     <path/to/outputFile.xml> Set the output file.
-f [value] --frequency Determines after how many frames the Behaviour
                           System checks the minimal path
--standar           Print the xml into the standar output.
--verbose
```

```

command = settings.RECOGNITIONSYS_BIN + " -i " + video_file + " -o " + xml_file + " --standar" + " -f 5"
video_nfames = utils.VideoUtils.get_number_frames(video_file)

p = Popen(command, shell=True, stdout=PIPE, stderr=STDOUT, universal_newlines=True)
while True:
    line = p.stdout.readline()
    if line == "":
        iago@iago-S551LB:~/PycharmProjects/TFG/src/RecognitionSystem$ ./recognitionsystem --help

Usage: recognitionsystem
option:
-i      --input      <path/to/outputFile.xml> Set the input file.
-o      --output      <path/to/outputFile.xml> Set the output file.
-f [value] --frequency Determines after how many frames the Behaviour
                        System checks the minimal path
--standar          Print the xml into the standar output.
--verbose

```

Figura 6.9: Interfaze de liña de comandos do sistema de recoñecemento

## Ficheiro XML

Como resultado desta chamada, o sistema de recoñecemento debe crear no ficheiro indicado para a saída, un XML co formato que se define no esquema .dtd situado no directorio “src/static/detections\_schema.dtd”. Neste ficheiro podemos ver a definición dos seguintes elementos:

- *< objects >*

Contén para cada un dos fotogramas *< frame >* a lista de obxectos detectados segundo o explicado no apartado 2.2, indicando para cada un deles o seu identificador, a distancia á parte esquerda, e superior da escena (xc, yc) e o alto e ancho do obxecto detectado (h, w).

- *< trajectories >*

Este outro elemento garda para cada un dos obxectos detectados a traxectoria que seguiu ao longo do vídeo, esta traxectoria estará composta de unha serie de puntos *< point >*, para cada un dos cales, a parte das súas coordenadas e o número de frame, indicase un grao de anormalidade entre 1 e 0 que indica como de anómala é a conduta dese obxecto no momento no que se atopa sobre ese punto.

Para xerar este ficheiro XML foi preciso desenvolver unha serie de funcionalidades en C++, que están contidas no paquete XmlRecognition.

### Paquete XmlRecognition

Inicialmente o proxecto conta cun código escrito en C++ e apoiado en OpenCV que está distribuído en dúas librerías EllipseLib e BehaviorLib. A primeira delas encargada da detección de obxectos, e a segunda encargada de analizalo comportamento a alto nivel a partir dos resultados que proporciona a primeira.

Para a construción do Sistema de recoñecemento integraranse estas dúas librerías como módulos, e crearse un terceiro módulo C++ chamado XmlRecognition (figura 6.10). Este novo módulo será o responsable de definir e implementar a interface de liña de comandos, xestionar a comunicación entre as dúas librerías e gardar o resultado da análise en formato XML.

Para elo, crease un ficheiro principal chamado **main.cpp** que describe a interface de liña de comandos independente da librería que se emprega para as deteccións, un ficheiro **XmlUtils** que contén as funcionalidades para a escritura do XML en base a unha detección simplificada: DetectionDto (Data Transfer Object), e por último un ficheiro **RecognitionFacade** que variará en caso de cambiar as librarías, transformando o resultado destas en DetectionDto's para logo poder escribilo coas funcionalidades de XmlUtils.

Para maximizar o rendemento evitase crear clases innecesarias: para o caso da DetectionDto é dabondo cunha estrutura, e no caso de XmlUtils y RecognitionFacade, ao non ter estado chega con ficheiros que definen funcións.

De cara ao público, o sistema de recoñecemento completo non será publicado, xa que as bibliotecas nas que se apoia son propiedade privada do laboratorio. Pero si se publicarán as clases desenvoltas a modo de unha fachada que sempre devolve unha análise baldeira, moi útil para probar o sistema.

Coa elaboración deste paquete queda listo o sistema de recoñecemento, agora solo resta que a capa web sexa capaz de chamar ao sistema e mostrar as análises en XML sobre o vídeo. Será o que abordaremos no seguinte apartado.

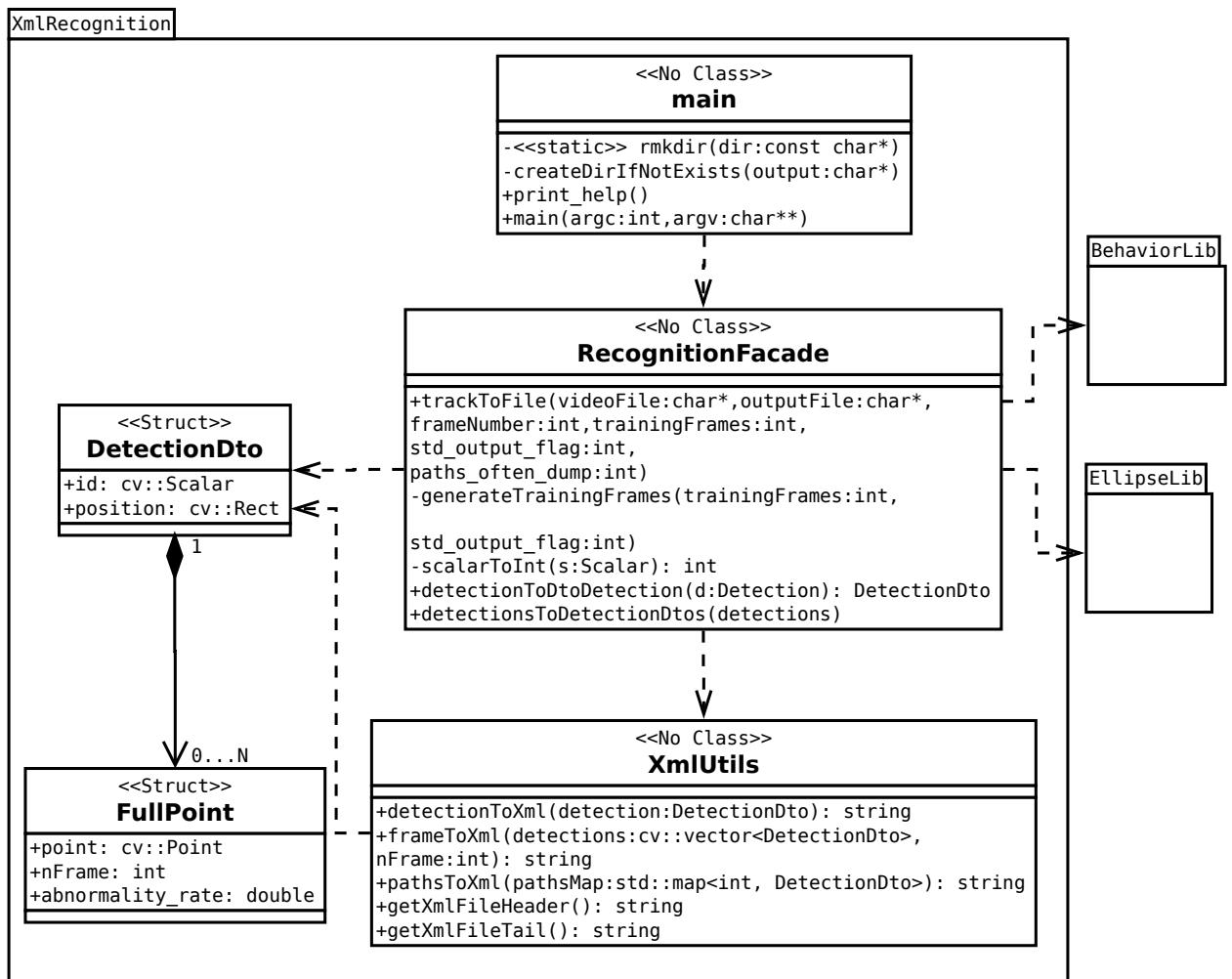


Figura 6.10: Diagrama de clases do paquete XmlRecognition

#### 6.4.2. Mostrar Deteccións

Unha vez que o vídeo xa está subido e correctamente analizado, o que resta é comenzar a construír na capa web as vistas que mostren sobre o elemento `<video>` de HTML5 as distintas capas de análise obtidas a partires do ficheiro XML así como unha serie de paneis que permitan configurar estas vistas.

Todo este traballo realizase na vista DetailsView do módulo video\_manager que finalmente se acaba renderizando como un ficheiro HTML5 con referencias a:

- O vídeo a mostrar
- O ficheiro XML que contén a análise realizada polo sistema de recoñecemento.
- Os ficheiros de estilo.
- Os distintos ficheiros de código javascript involucrados.

Para amosar a análise do ficheiro XML sobre o vídeo a estratexia será a de a de superpoñer distintos elementos `<canvas>` sobre o elemento vídeo `<video>` como se pode ver na figura 6.4. Para axustar todas estas capas etiquetadas **como class=“drawing-layer”** sobre o vídeo crease a función `adjustCanvasExtended` do ficheiro “video-player.js”, esta función chamase cando o vídeo se carga na páxina, en caso de que o tamaño do vídeo cambie ou cando se entra e sae do modo pantalla completa, axustando de novo o tamaño de todas estas capas ao actual tamaño do vídeo.

Unha vez axustados os elementos `<canvas>` nos que se deseja amosar a información, tense que extraer esta do ficheiro XML. O XML cargase mediante AJAX, cunha chamada de jQuery `$.get(...)` ao dirección do servidor que indica a etiqueta oculta con id `xml_detected_objs`, unha vez cargado este arquivo iniciase a carga inicial do sistema.

Esta carga inicial consiste na creación dun obxecto `VideoDetections` creado a partir do XML, e que conterá a lista de deteccións así como unha referencia ao elemento `<video>`. Este elemento `VideoDetections`, é o suxeito de un patrón Observador (figura 6.11) no cal un suxeito ou obxecto central notifica ao seus observadores (`Observers`) os cambios no seu estado. Neste caso, os observadores serán os encargados de actualizar cada un dos elementos da páxina, incluídas as capas `<canvas>` en función dos

cambios no suxeito VideoDetections, estes cambios poden ser por exemplo o avance na reproducción do vídeo, un cambio de preferencias no panel de control... Deste xeito unifícase a xestión das deteccións que só se manexa no elemento VideoDetections e faise moito máis sinxelo engadir ou eliminar algunha capa de información xa que basta con engadir ou eliminar o observador que a manexa.

Nótese que a maiores do propio patrón observador, tamén se engaden os métodos enable e disable á clase DetectionsObserver, isto faise para que no caso de que algúin observador non se atope activo nun momento determinado non reciba as notificacións de actualización. O motivo deste cambio é o incremento de eficiencia que produce, moi importante dado que o proceso de actualización do Suxeito e todo-los seus observadores tense que executar entre 10 e 30 veces por segundo, todo isto pódese ver reflexado no diagrama da figura 6.11.

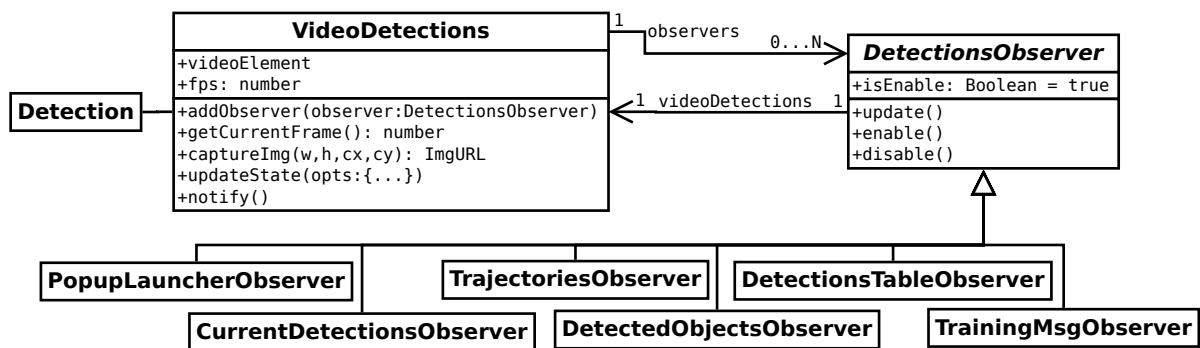


Figura 6.11: Diagrama de clases simplificado do patrón observador na capa web

Agora que se coñece o xeito no que as deteccións pasan de un formato XML a un modelo obxectual no cal poden ser consultadas con maior eficiencia verase como actualizar o estado dos elementos *< canvas >* cada vez que se mostra un novo fotograma ou se modifica algunha opción dos paneis da páxina.

Nun comezo, pensouse en asociar a actualización de estado ao evento timeupdate do elemento *<video>*, que segundo a súa definición lanzase cando a posición do vídeo varía. Por desgraza, e como reflexa a reflexión que podemos atopar no libro [35, Capítulo 6.1] este evento é lento de máis para o seu emprego, polo que o que se fará será crear un bucle que se execute cando o vídeo se estea a reproducir. Para tentar que o bucle

se execute unha vez en cada un dos fotogramas empregase a función `setTimeout` que fará que o código agarde unha cantidade de tempo ( $t_n$ ) antes de volver a executarse. Esta cantidade de tempo calculase en base ás dez execucións anteriores e ao número de fotogramas por segundo como se pode ver na seguinte formula:

$$t_n = \frac{1000}{\text{videoDetections.fps}} - \frac{\sum_{i=0}^{10} \text{lastUpdateTimes}[i]}{10}. \quad (6.1)$$

O número de fotogramas por segundo para esa fonte ao que se accede mediante o obxecto `videoDetections` obtense grazas a que o servidor o inclúe no HTML mediante o atributo `fps` en cada elemento `<source>` do elemento `<video>` como se ve na figura 6.4. Á súa vez o servidor obtén estes datos para as distintas fontes de vídeo dunha chamada ao método `VideoUtils.get_fps` que chama ao `ffmpeg` como se ve no diagrama de secuencia 6.12.

Coñecido pois o xeito no que se refresca o estado da interface de usuario e o sistema que crea un modelo obxectual para a representación gráfica das deteccións, agora compre ver como é que se remarcan os obxectos detectados no vídeo. Isto faise mediante o elemento `<canvas>` con `id="objects-canvas"` no que o observador da clase `Detecte-dObjectsObserver` encargase de debuxar un recadro da cor axeitada. Para coñecer esta cor que depende das opcións do panel o observador consulta a cada obxecto `Detection` empregando o método `getCurrentColor()` que se pode ver no diagrama 6.13.

Tamén cabe destacar aquí a función do observador `TrainingMsgObserver`, que é o encargado de mostrar a etiqueta de Fotogramas de Adestramento e o recadro de cor amarelo nos primeiros fotogramas do vídeo que corresponden aos empregados polo sistema de recoñecemento para obter o fondo da imaxe e así poder distinguir os obxectos que se moven sobre el.

## 6.5. Cuarto Sprint: Arranxar bug's e mellorar a estratexia de probas

Esta versión estivo adicada a mellorar algúns aspectos transversais da aplicación que se consideraban de importancia como a verificación e securización das páxinas web's

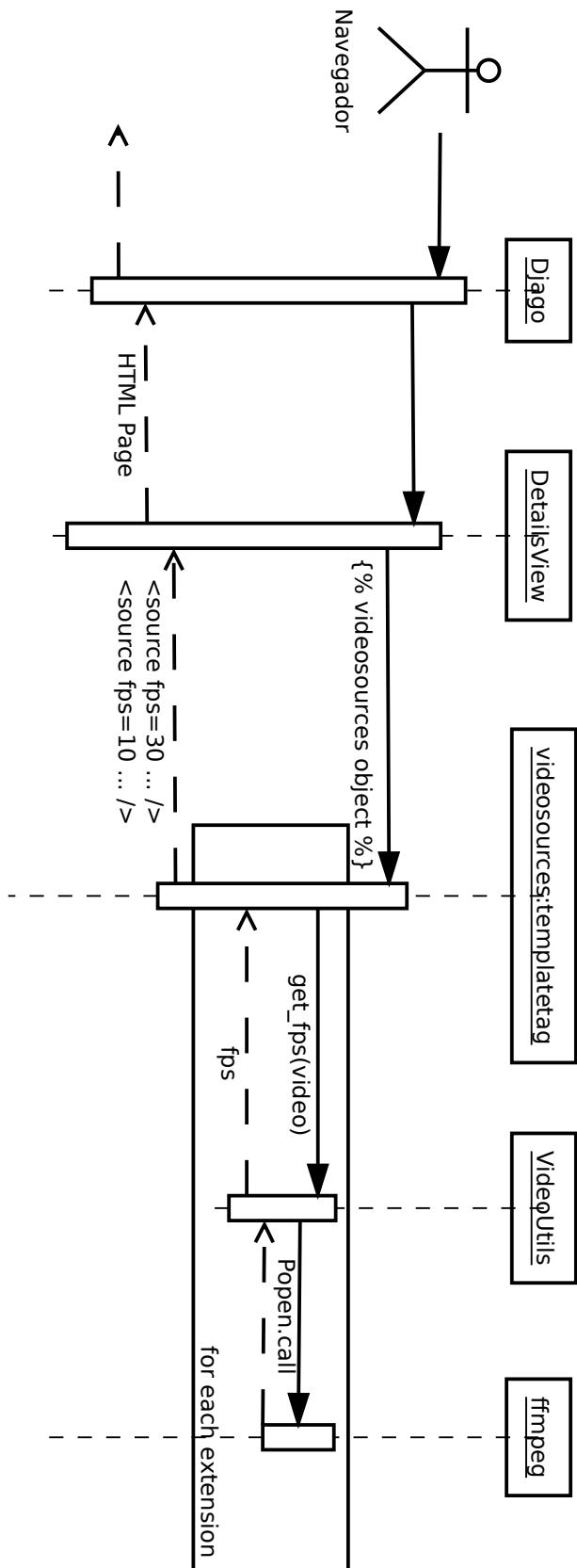


Figura 6.12: Xeración do atributo fps no lado servidor

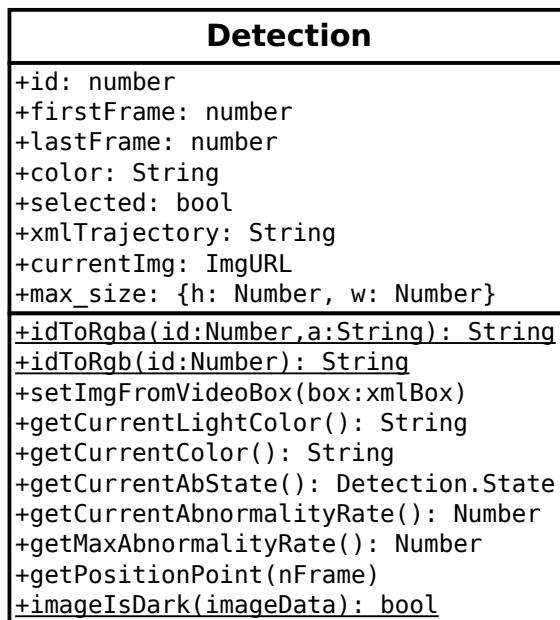


Figura 6.13: Diagrama completo da clase Detection

que se detalla con detalle no capítulo de Validación.

### 6.5.1. Re-analizar vídeo

En este sentido representa un incremento funcional pequeno en comparación co acometido en outras iteracións, a funcionalidade más destacable das elaboradas neste período foi a de Reanalizar o vídeo unha vez subido dende a páxina de reprodución. Isto é moi útil no caso de que queiramos aplicar cambios ao sistema de recoñecemento e ver velozmente como estes cambios afectan ao resultado da análise, xa que non é preciso volver a subir o vídeo á plataforma. O aspecto final desta funcionalidade pódese ver na figura 6.14.

En canto ao seu deseño, empregouse a clase Analysis Process que se pode ver no diagrama 6.7 da sección 6.3.2, e segue para realizar a análise un comportamento completamente análogo ao explicado nesa sección para a análise do vídeo tras a súa subida.

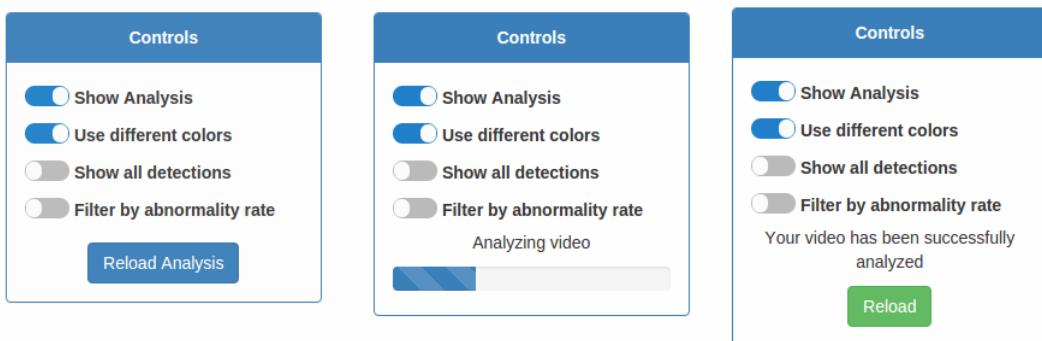


Figura 6.14: Capturas de pantalla que amosan os distintos estados do botón de re-analise

## 6.6. Quinto Sprint: Traxectorias

As traxectorias son outra parte fundamental do sistema, debese mostrar a ruta seguida por cada un dos obxectos detectados dende o momento no que entra en escena ata a súa saída. Para que as traxectorias sexan detectadas no sistema de recoñecemento hai que invocar á biblioteca de análise de alto nivel que require de unha cantidade de tempo considerable. É por elo que na interface de liña de comando precisase un argumento de frecuencia que indique cada cantes fotogramas se realizará a análise de alto nivel(por defecto cada 5 fotogramas).

As traxectorias chegan á capa cliente como parte do XML resultado do sistema de análise, para mostralas empregarase unha nova capa *< canvas >* en conxunto cun novo observador TrajectoriesObserver que itera sobre as deteccións actuais pintando as liñas entre cada un dos puntos da súa ruta, dende o comezo desta ate o punto no que se atopa actualmente o obxecto. Estes puntos son parseados polo obxecto Detection nada máis recibir o documento XML, quedando almacenados como parte do seu campo trajectory.

Xa que a traxectoria forma parte da análise de alto nivel, e esta non é executada en tódolos fotogramas, unha traxectoria estará composta por puntos que indican a posición do obxecto detectado con certa frecuencia (figura 6.15). Para que a liña que indica a traxectoria se vaia pintando ao paso do obxecto detectado cando este avanza entre un

punto desta traxectoria e o seu seguinte, dando así sensación de fluidez, o que se fai é calcular dado ese seguinte punto e o momento actual, cal é o lugar no que debería estar dito obxecto se se movese en liña recta entre ambos puntos, debuxando a liña da traxectoria ate esa posición calculada.

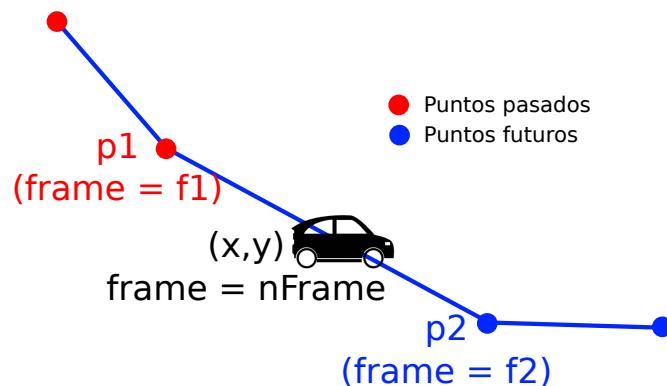


Figura 6.15: Diagrama do cálculo da traxectoria

O cálculo realizase en base ao “ratio” que indica a distancia de entre os dous puntos que o coche leva percorrido.

```
var ratio = Math.abs((nFrame - f1) / (f1 - f2));
var x = p1.x + ratio * (p2.x - p1.x);
var y = p1.y + ratio * (p2.y - p1.y);
```

Os resultados destes cálculos gárdanse para ser empregados en frames posteriores en caso de que sexa preciso.

Na captura da imaxe 6.16 pódense ver tanto os obxectos detectados con un recadro

enriba como a liña indicativa da súa traxectoria dende o momento no que estrou en escena ate o punto no que se atopa actualmente.

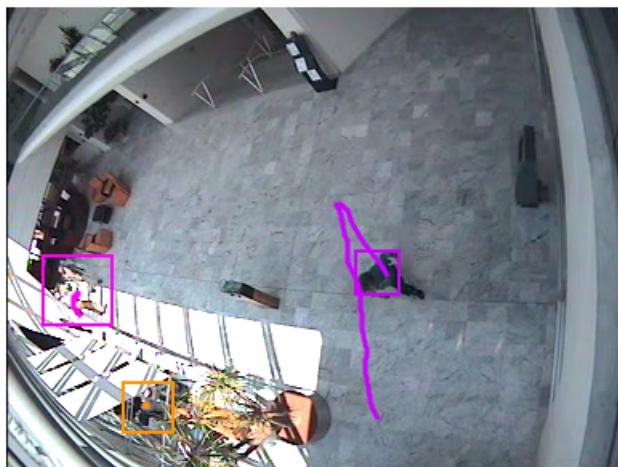


Figura 6.16: Captura que amosa o funcionamento das capas de traxectorias e deteccións

A parte das traxectorias, nesta iteración tamén se deu solución a outras funcionalidades como a táboa de Deteccións e a lista de deteccións actuais.

### 6.6.1. Táboa de Deteccións

Co fin de poder ollar todas as deteccións e podelas ordenar por orde de aparición, tempo en escena, etc, crease unha táboa de obxectos detectados que ademais indica os obxectos que están a aparecer agora mesmo no vídeo, vexase figura 6.18. A estes obxectos asignaselle unha cor en función do seu identificador único, que ven xa composto por tres valores entre 0 e 255, este identificador tratase segundo o model RGB como se mostra na figura 6.17 para obter cores máis brillantes e vistosas.

Para o ordenamento da táboa empregouse a biblioteca javascript tablesorther que permite ordenar ascendente ou descendenteamente a táboa facendo click na cabeceira do elemento polo que se desexa ordenar. Tamén é de destacar que os identificadores situados na primeira columna conteñen un enlace que ao facer click leva o vídeo ao momento no que ese obxecto aparece por primeira vez en escena, función de tremenda utilidade á hora de recoñecelo.

### idToRgb function

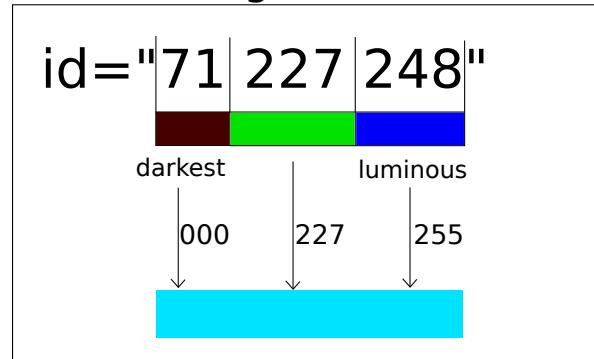


Figura 6.17: Diagrama que mostra o aclarado de cores



Figura 6.18: Captura de pantalla que amosa a táboa de obxectos detectados

Como é lóxico esta táboa actualizarase en cada un dos fotogramas se hai algún novo obxecto detectado ou se algún de eles desaparece, marcando ou desmarcando este como obxectos en escena. Este comportamento implementase mediante dúas listas no obxecto VideoDetections: **detRecentlyDeleted** e **detRecentlySelected**, que son lidas polo observador DetectionsTableObserver, encargado de remarcar todas as Deteccións da lista detRecentlySelected e de desmarcar todas as da lista detRecentlyDeleted, estas relacións entre as clases VideoDetections e Detection pódense observar no diagrama 6.19.

### 6.6.2. Lista de Deteccións actuais

Probablemente a función más necesaria e vistosa é a de amosar os obxectos detectados de forma individualizada mentres estan en pantalla. A estes efectos crease unha lista de deteccións actuais que amosa unha imaxe de tódolos obxectos presentes no vídeo. O resultado pódese observar na figura 6.20.

Esta sección de deteccións actuais ao igual que a táboa de obxectos detectados precisa ser actualizada en cada fotograma, para elo empréganse as listas de detRecentlyDeleted e detRecentlySelected explicadas anteriormente, só que agora é o observador CurrentDetectionsObserver o encargado de engadir ou borrar unha detección desta sección.

Tamén hai que destacar que xa que os observadores CurrentDetectionsObserver e DetectionsTableObserver son os que más tempo de execución consumen, a súa execución é complementaria dependendo do estado do checkbox “Show all detections”. De estar deshabilitado mantén o observador DetectionsTableObserver inactivo mentres que o CurrentDetectionsObserver está activo, e estando habilitado fai o contrario. Isto obriga aos observadores a reimplementar os métodos “enable()” e “disable()”, cargando ou descargando nestes métodos todos os valores actuais das deteccións que non actualizaron mentres estaban deshabilitados.

As imaxes que se amosan como icona de cada un dos obxectos detectados son calculadas en tempo de reproducción do vídeo, isto é posible grazas aos elementos data URI[36] de javascript, que permiten almacenar unha imaxe como se fose unha URL e

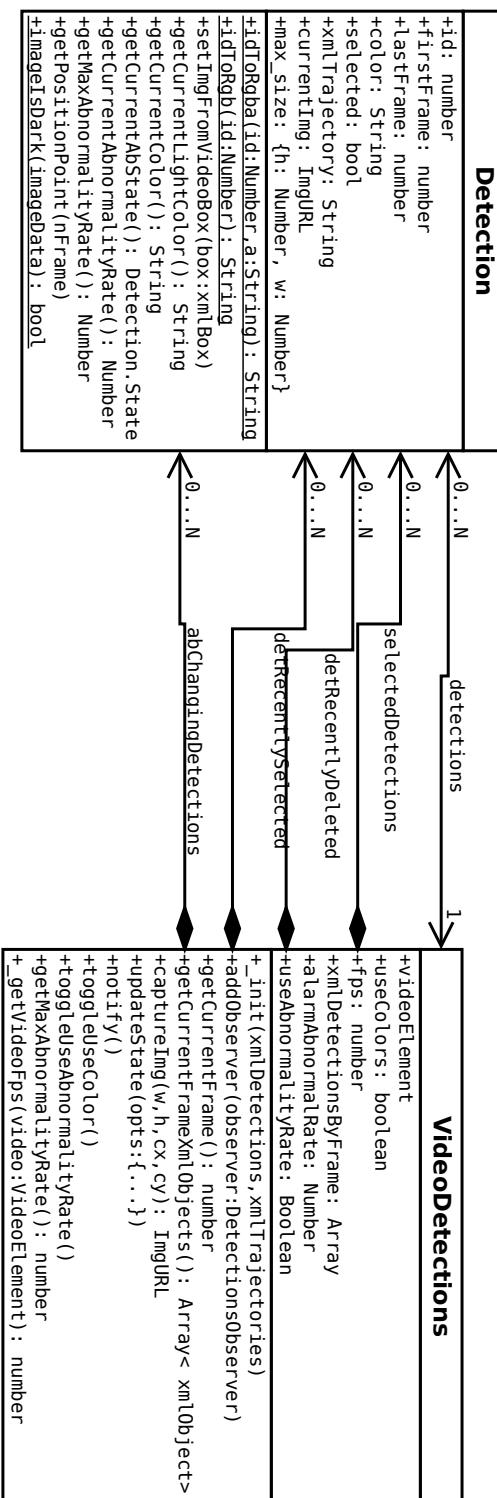


Figura 6.19: Diagrama de clases que amosa detalladamente a relación entre as clases Detection e VideoDetections

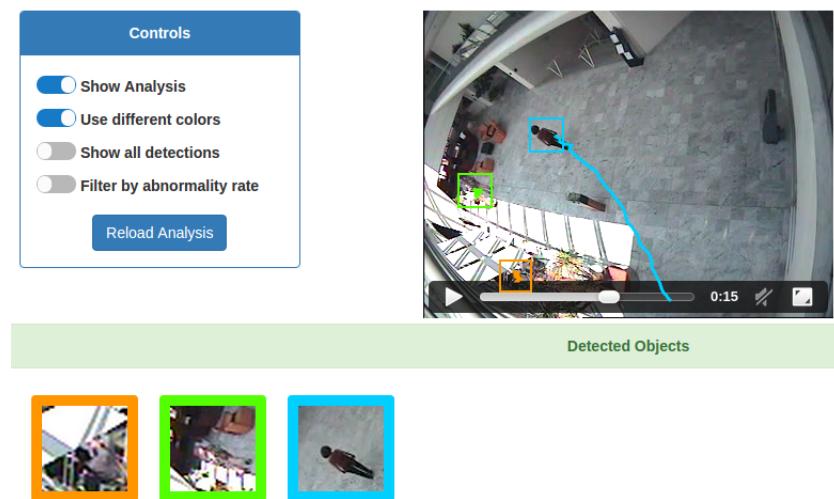


Figura 6.20: Captura de pantalla que amosa a lista de obxectos detectados

empregar esta URL para logo mostrar a imaxe, por poñer o caso, no fondo (background-image) de un elemento HTML. Esta URI calcúlase no método da clase VideoDetections “captureImg” empregando unha chamada a “canvas.toDataURL()” onde canvas é un elemento HTMLCanvasElement obtido a partir do fotograma actual do vídeo.

Por último para mostrar a información asociada a este obxecto detectado de algúñ xeito, empregase a componente da librería bootstrap popover, que é unha ventá flotante que xurde ao carón da detección cando o rato pasa por enriba dela. Este popover personalizase coa cor concreta desa detección (ver figura 6.21).

## 6.7. Sexto Sprint: Detección do comportamento anómalo

Un dos factores nos que destacan os algoritmos dos laboratorios VARPA é precisamente a análise de comportamento co fin de detectar o comportamento anormal de un dos obxectos implicados nunha escena. Isto faise calculando cal é o camiño polo que un obxecto soe viaxar do punto A ao punto B, para calquera que sexan os puntos A e B, de xeito que podemos medir o anormal que é o seu comportamento en función do que o seu camiño se diferencie do camiño habitual. Isto pode axudar a detectar por exemplo un peón que non cruza polo paso de cebra, un coche facendo unha manobra perigosa...

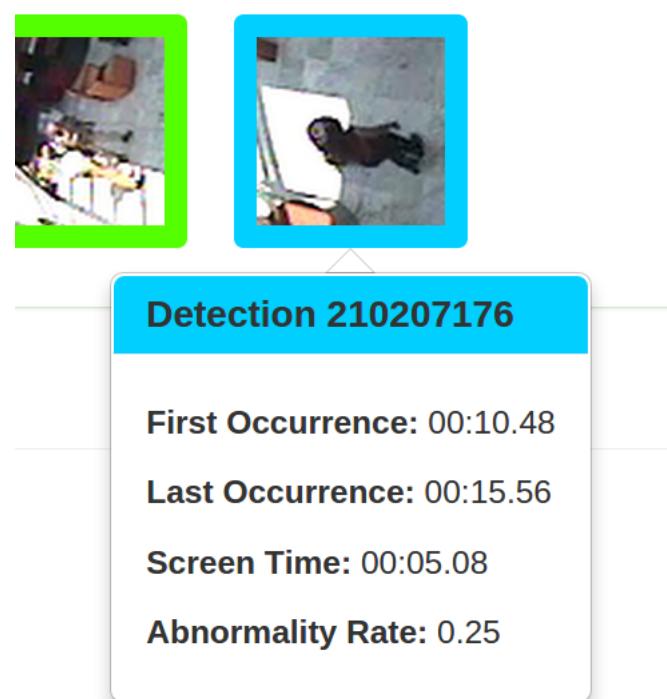


Figura 6.21: Captura de pantalla que amosa o popover da lista de deteccións actuais

O sistema de recoñecemento facilitado polo laboratorio foi deseñado para poder devolver un ratio de esa anormalidade en cada unha das chamadas ao sistema de análise de Alto Nivel, que como se dixo anteriormente executase cada determinado período de tempo. Dado que é este sistema o que proporciona a traxectoria do obxecto, o ratio de anormalidade para cada un dos momentos asocíouse pois a cada un dos puntos da traxectoria como se pode ver na imaxe 6.22. Como se pode comprobar nesta imaxe, existen unha serie de momentos iniciais nos que o ratio inda non puido ser calculado, e neste caso o valor asociado será 0.

```
</trajectory>
<trajectory id="173031122">
<point frame="269" abnormality="0" x="529" y="310"></point>
<point frame="275" abnormality="0" x="530" y="309"></point>
<point frame="281" abnormality="0" x="531" y="308"></point>
<point frame="283" abnormality="0" x="531" y="308"></point>
<point frame="289" abnormality="0" x="531" y="307"></point>
<point frame="295" abnormality="0.2" x="532" y="307"></point>
<point frame="301" abnormality="0.2" x="533" y="305"></point>
<point frame="307" abnormality="0.2" x="536" y="302"></point>
<point frame="313" abnormality="0.25" x="542" y="296"></point>
<point frame="319" abnormality="0.222222" x="549" y="287"></point>
<point frame="325" abnormality="0.2" x="559" y="276"></point>
<point frame="331" abnormality="0.181818" x="568" y="264"></point>
<point frame="337" abnormality="0.166667" x="577" y="252"></point>
<point frame="343" abnormality="0.153846" x="591" y="238"></point>
<point frame="349" abnormality="0.214286" x="596" y="225"></point>
<point frame="355" abnormality="0.180474" x="605" y="213"></point>
<point frame="361" abnormality="0.33586" x="613" y="202"></point>
</trajectory>
<trajectory id="182093078">
```

Figura 6.22: Ratio de anormalidade gardado nun documento XML

Non obstante un número é algo moi pouco gráfico, que difícilmente pode axudar a resaltar unha detección anómala de unha que non o é, por este motivo se deseñou unha serie de componentes que permiten seleccionar un límite (un ratio de anormalidade máximo) a partir do cal un obxecto será considerado sospeitoso. A nivel de interface web, este límite pode marcarse cunha barra selectora chamada slider, que se importou da libraría jQuery UI[27], e que se complementou cunha entrada de texto para dar a posibilidade de que o valor se seleccione ou ben escribíndoo, ou ben a través do slider (vexase figura 6.23). Os valores a seleccionar variarán sempre entre 0 e o máximo valor atopado entre os ratios de anormalidade das deteccións.

Agora que xa se pode seleccionar o valor a partir do cal queremos marcar unha

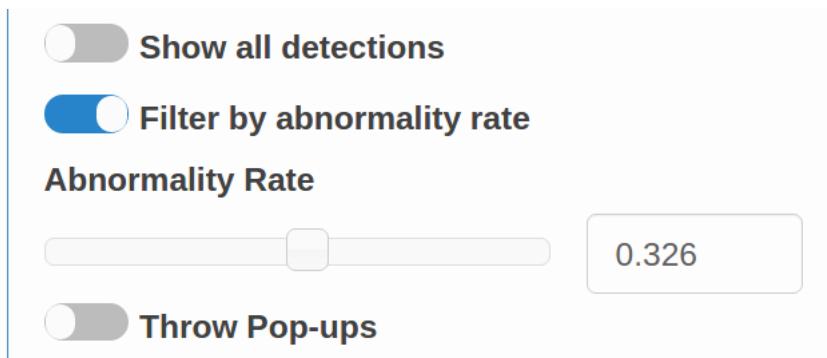


Figura 6.23: Captura de pantalla do compoñente slider da biblioteca jQuery UI

detección como anómala, falemos do xeito no que se vaia realizar este marcado. A idea xeral consiste en inabilitar o uso de diferentes cores e marcar de azul todas as deteccións, con excepción de aquellas que teñan un comportamento anormal que serán marcadas de vermello e aquellas cuxo comportamento inda non fose analizado que serán marcada en cor negra como se pode ver na imaxe 6.24. Este uso de cores no só se emprega nas capas *< canvas >* de información que se amosan sobre o vídeo senón tamén na táboa de obxectos detectados e na lista de deteccións actuais.

Para conseguir isto, centralizase a xestión da cor sobre o obxecto Detection que se pode ver en detalle no diagrama 6.13, facendo que sexa este o que determine de forma unívoca de que cor se ten que debuxar. Para isto créanse os métodos “getCurrentColor” que devolve a cor para as capas *< canvas >* e “getCurrentLightColor” que devolve un color máis claro para a táboa e a lista de deteccións. O obxecto Detection á súa vez calculará a cor a devolver en función do seu ratio de anormalidade e o valor das propiedades: videoDetections.alarmAbnormalRate, videoDetections.useAbnormalityRate e videoDetections.useColors.

A parte deste mecanismo para centralizar a xestión do color que será empregado por tódolos observadores, os observadores DetectionsTableObserver e CurrentDetectionsObserver precisan coñecer aquellas deteccións que cambiaron a un estado sospeitoso, de confianza ou sen datos, con este obxectivo crease a lista **abChangingDetections** na clase VideoDetections (ver diagrama 6.19), que indicará aquellas deteccións que cambiaron de estado e polo tanto, no caso da táboa e a lista de deteccións, precisan mudar

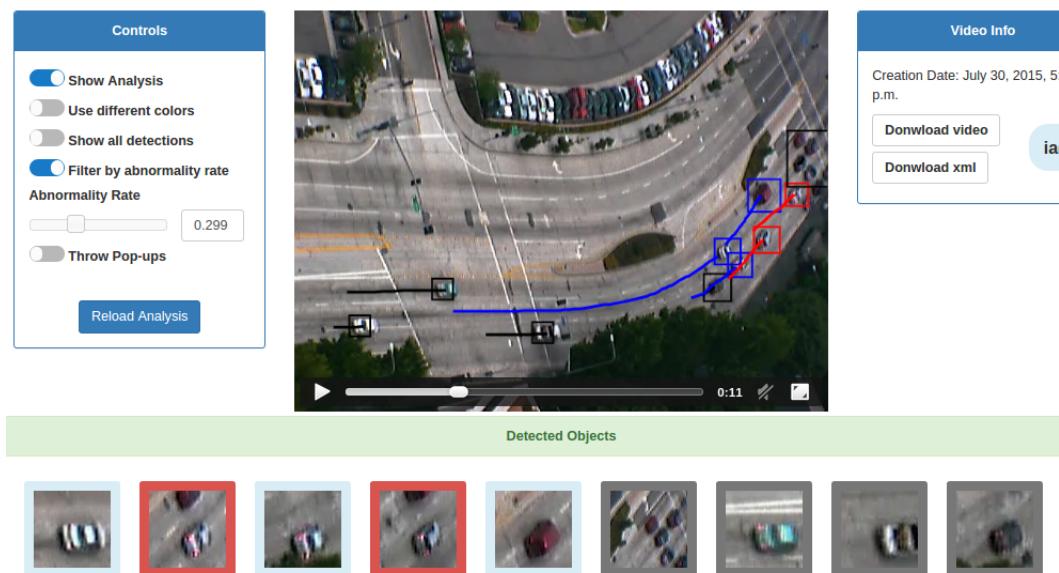


Figura 6.24: Captura de pantalla que amosa o comportamento das deteccións sospeitosas

a súa cor.

### 6.7.1. Popups de deteccións sospeitosas

Para completar a funcionalidade de detectar o comportamento anómalo, solicitase unha faceta a maiores do marcado das deteccións sospeitosas, deberase dar a opción de lanzar unha nova ventá por cada obxecto sospeitoso detectado coa posibilidade de ver este obxecto de preto ao largo do seu percorrido.

Tras estudar varios xeito de facer isto, a mellor solución parece ser a de empregar pop-ups tamén chamados ventás emergentes, que inda que normalmente se evitan por estares asociados con contido publicitario, neste caso compren á perfección coa tarefa que se desexa realizar.

Xa que haberá que ampliar a imaxe facendo un efecto zoom para poder mostrar a zoa do vídeo na que está a detección con máis detalle, proponse a construción dun sistema que mediante unha barra selectora (slider) permita achegarse ou afastarse para ver máis de preto ou de lonxe o obxecto detectado. Para estas ventás emergentes empregarase

un novo ficheiro .css e tamén un novo ficheiro .js, que conterá os mecanismos precisos para crear este efecto zoom.

A esta nova ventá pasaranse como parámetros da URL o identificador da detección e a ULR do ficheiro XML que contén a análise. Con estes datos o ficheiro javascript principal chamado “*suspicious-popup.js*”, que segue unha distribución similar ao de “*video-player.js*” para a páxina que amosa o vídeo, obterá o ficheiro XML e en base a el iniciará o sistema. Esta páxina tamén constará dun elemento *<video>*, con capas *<canvas>* superpostas, a primeira delas ocultará o vídeo por completo mostrando a porción del que sexa preciso para lograr o efecto zoom e as sucesivas mostrarán a información desexada.

Ao manter a estrutura dun elemento *<video>* coas capas *<canvas>* superpostas, a mellor arquitectura posible é de novo a dun patrón Observador, no que o suxeito manteña ademais das referencias ao vídeo, os datos sobre o nivel de zoom, que será necesariamente empregado polos seus observadores. Para esta tarefa reempregouse gran parte do deseño do diagrama 6.11 engadindo as características precisas nunha clase herdada de VideoDetections que se nomea ZoomVideoDetections como se pode ver na figura 6.25.

Este sistema é iniciado na carga do XML e ao igual que na páxina que reproduce o vídeo, cando a carga finaliza crease un bucle coa función *updateStatus* que actualizará en cada fotograma o estado dos componentes chamando á función *ZoomVideoDetections.updateState()*. Esta función tamén é chamada cando o vídeo lanza o evento play, ou ao cambiar o nivel de zoom mediante a barra selectora (slider). O resultado final mostrase na figura 6.26.

A continuación explicase en detalle o mecanismo empregado para crear o efecto zoom e para levar a cabo o seguimento do obxecto mantendo este nivel de zoom. Empréganse basicamente dúas variables, por un lado o tamaño da fracción de vídeo a amosar (*w\_size*), que variará entre un mínimo(dúas veces o tamaño da detección) e un máximo que é o tamaño do vídeo, e por outro lado o centro (center) da escena que será unha variable composta de dúas componentes (*x,y*) que indican a distancia en píxeles á parte superior esquerda do vídeo e que corresponde en cada momento coa

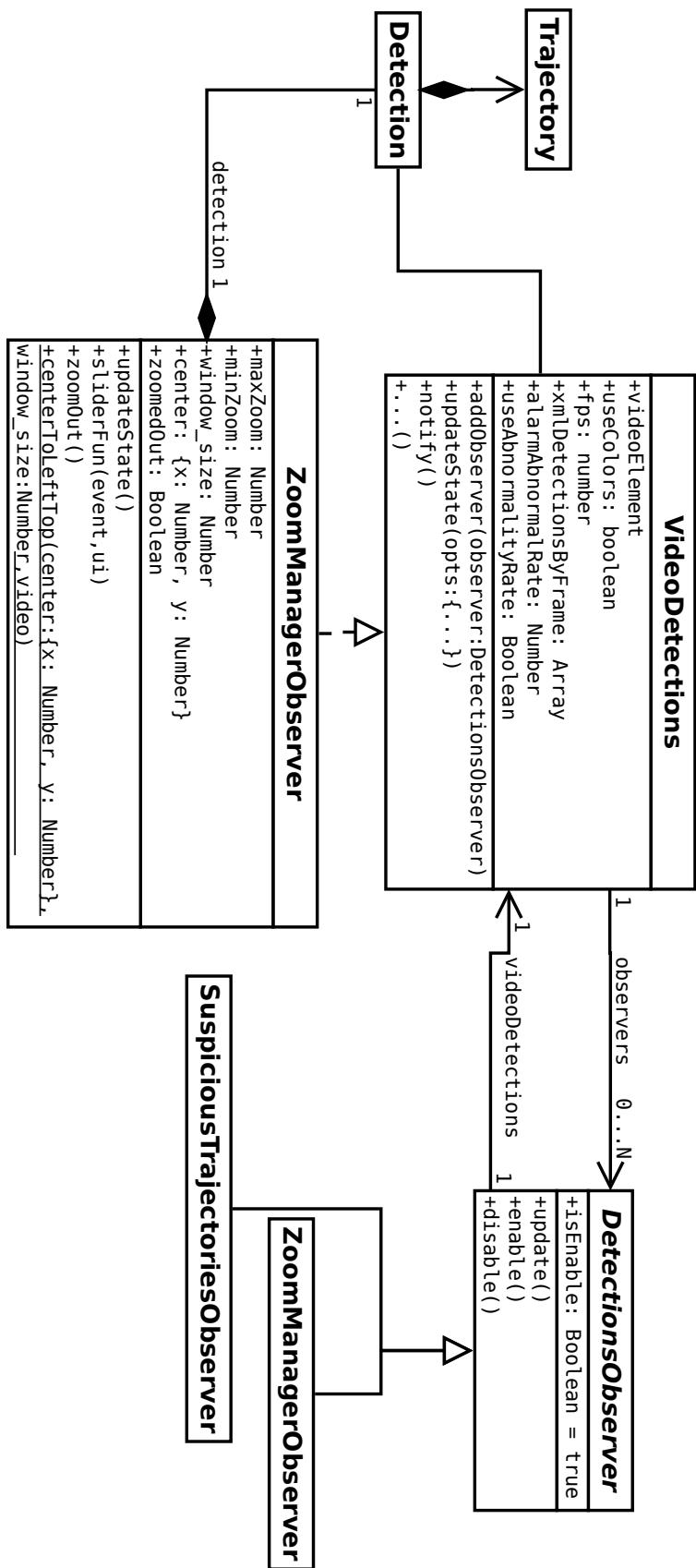


Figura 6.25: Diagrama de clases simplificado para a ventá emerxente de deteccións sospeitosas

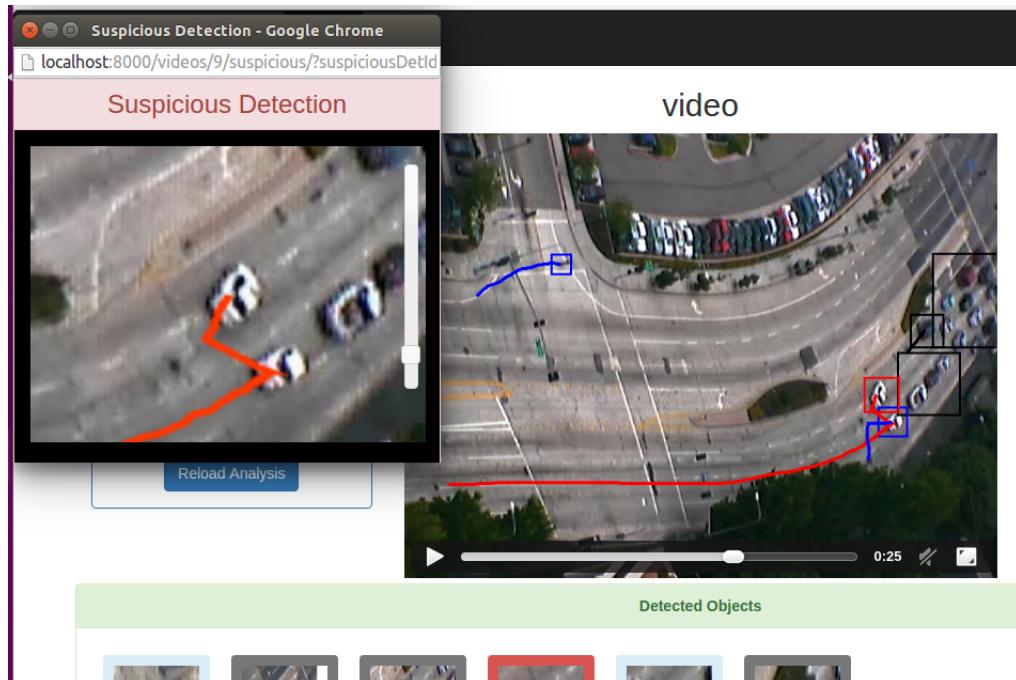


Figura 6.26: Captura de pantalla que amosa a ventá de deteccións sospeitosas

posición do obxecto sospeitoso.

A función `ZoomVideoDetections.centerToLeftTop` é a encargada de calcular dado un tamaño de ventá e un centro, esta distancia á marxe esquerda superior do vídeo. Este cálculo non é sinxelo, xa que hai que evitar que o recadro a seleccionar exceda das dimensíons do vídeo como se pode comprobar na figura 6.27.

## 6.8. Sétimo Sprint: Memoria e posta en producción

Esta última fase adicouse principalmente a completar a memoria, a solucionar pequenos problemas acumulados nas anteriores versións e á publicación de esta aplicación baixo un entorno de producción.

Para a publicación desta aplicación web, realizarase a súa montaxe sobre un servidor on-line, é dicir, unha máquina conectada a internet que dispón de conexión directa a unha dirección IP fixa, e a capacidade para manterse en funcionamento de forma continua. A esta IP referenciará un servidor DNS, que se contratará xunto co dominio

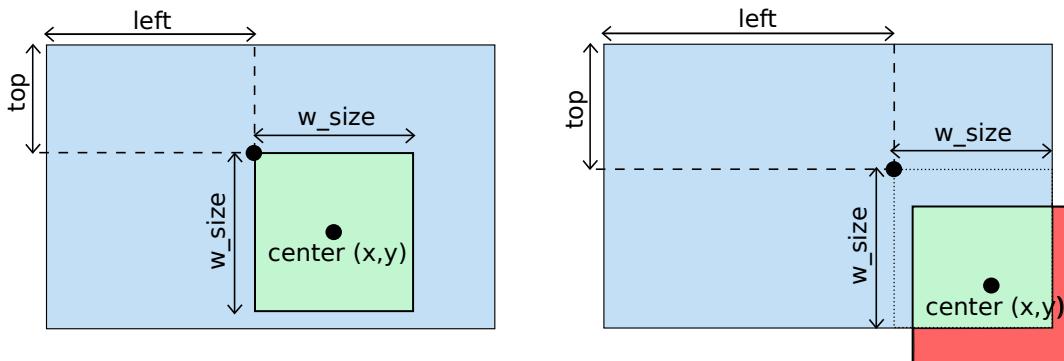


Figura 6.27: Diagrama que explica os problemas resoltos para implementar o efecto zoom

[www.ancoweb.es](http://www.ancoweb.es) a través do cal poderase acceder á páxina.

### 6.8.1. Dominio

O dominio ten que ser rexistrado cos datos da persoa responsable, neste caso o autor do proxecto, a través dunha empresa rexistradora. Seleccionase a empresa 1&1 [37] xa que tamén oferta servizo DNS(Domain Name System) para apuntar ao servidor que contén a aplicación.

Unha vez creado o dominio, a única configuración a establecer para que este funcione é a dirección do servidor, que indicaremos no rexistro de tipo A como se pode ver na figura 6.28 onde a dirección IP do servidor é 45.55.51.164.

### 6.8.2. Hosting

Na dirección IP 45.55.51.164 reside o servidor montado para distribuír a aplicación web a todo aquel que se conecte. Este servidor pertence á empresa DigitalOcean [38] que se seleccionou de entre outros servizos de hosting por permitir a posibilidade de crear o entorno de produción dende cero, podendo así instalar todos os paquetes preciso para a integración das múltiples tecnoloxías implicadas neste proxecto.

Neste caso seleccionouse como punto de partida unha máquina Ubuntu Server 14.04

The screenshot shows the 1&1 Control Panel interface. At the top, there's a navigation bar with links for 'Centro de dominios', 'Inicio', 'Dominios', 'Mis planes', 'Mis datos', '1&1 Tienda Clientes', 'Ayuda', and 'Cerrar sesión'. Below the navigation bar, the main header reads 'Centro de dominios' and 'Comenzar tour de producto'. To the right, it displays 'ID de cliente: 452779324' and a dropdown menu showing '55019422 - 1&1 Pack Dominio'. A sub-header 'Gestión de dominios y certificados SSL' is present. Below this, a horizontal menu bar includes 'Dominios' (which is underlined), 'Transferencias', 'Pre-registros', 'Reservas', and 'Certificados SSL'. Under the 'Dominios' menu, there are two buttons: 'Solicitar dominio' and 'Exportar lista de dominios'. The main content area is titled 'Dominio' and 'Detalles'. It lists a domain entry for 'ancoweb.es' with the note 'Espacio web: /.'. Below this, detailed information is provided: 'Detalles del dominio' (Destino: Espacio web, Objetivo: /, Tipo: Dominio adicional), 'Configuración DNS' (Servidor DNS 1&1, A 45.55.51.164, MX mx00.1and1.es), and 'Configuración de dominio' (with options to 'Editar destino', 'Editar contactos del dominio', and 'Modificar configuración DNS'). A vertical 'Feedback' button is located on the right side of the details panel.

Figura 6.28: Panel de control da ferramenta 1and1

con 1GB de memoria RAM, un só núcleo de CPU e 30GB de memoria SSD, todo elo localizado en Nova York.

### 6.8.3. Configuración do Servidor

A partir da base que nos proporciona DigitalOcean, que é unha máquina Ubuntu Server 14.04 completamente limpia, pódese seguir a guía contida no ficheiro production-README.md situado na base do proxecto que indica con detalle todos os comandos a executar para una correcta configuración.

Como peza central escolleuse o servidor Apache2, este servidor será o encargado de servir tódolos arquivos, tanto páxinas web como ficheiros multimedia e recursos da aplicación. Mais este servidor de por si só, non pode executar o framework de Django que se precisa para atender as peticións dos clientes, co que se engade un módulo chamado mod\_wsgi que permite executar código Python sobre o servidor mediante o ficheiro do proxecto “src/ancoweb/wsgi.py”. A dificultade especia está en configurar adecuadamente todo o entorno para executar Python 3.4, pois é unha versión soportada por mod\_wsgi recentemente.

### 6.8.4. Axustes de Produción

En producción existen toda unha serie de parámetros que cambian en relación co entorno de desenvolvemento, para soportalos creouse un novo ficheiro de configuración que herda do anterior chamado “settings\_production.py” e que está situado no directorio “/src/ancoweb”.

Para indicar ao sistema que empregue este ficheiro en lugar de settings.py, o que se fixo foi modificar o ficheiro “src/ancoweb/wsgi.py” facendo que este apunte ao ficheiro de producción en lugar de ao de desenvolvemento, así cada vez que o proxecto se execute mediante mod\_wsgi empregaránse os axustes de producción e en caso contrario os axustes por defecto.

Un dos axustes de configuración más importantes é o da variable STATIC\_ROOT, esta indica onde se almacenarán os recursos da aplicación, e neste caso decidiuse empregar o directorio “/var/www/static/” polo que antes de ser despregada todos o recursos

precisos de tódalas aplicacións incluídas deben ser movidos a este directorio empregando o comando:

```
sudo python manage.py collectstatic  
--settings=ancoweb.settings_production
```

O outro axuste de gran interese é o lugar no que se almacenan os ficheiros multimedia, pois a meirande parte da aplicación xira en torno a eles. Neste caso a variable en conflito é MEDIA\_ROOT, que no caso do entorno de producción tomará o valor “/var/www/media/”, e en caso de estar en desenvolvemento “MEDIA\_ROOT = str(BASE\_DIR / ‘media/’)”. Isto obriga ao igual que cos recursos do proxecto a mover os arquivos multimedia cada vez que se deseñe despregar a aplicación, neste caso co comando:

```
sudo cp -R ancoweb-TFG/src/media/ /var/www
```



# Capítulo 7

## Validación

Á hora de deseñar probas é importante abrancar a maior parte do código posible, neste proxecto isto foi todo un reto, pois o alto nivel de integración dificulta enormemente a realización das probas. Pese a todo, logrouse probar tanto o código realizado en Python-Django así como o código da capa cliente en javascript, empregando para elo distintos modelos e bibliotecas de probas que vemos a continuación.

### 7.1. Probas Unitarias

As probas unitarias proban as funcionalidades más básica do software. Executanse sempre no ámbito de un só modulo para probar o correcto funcionamento de este, simulando se fose preciso a súa interacción con outros módulos (este proceso chamase mocking).

No ámbito da nosa aplicación as probas de cada módulo recóllense nos ficheiros tests.py de cada un deles e nos ficheiros da carpeta “src/static/site/tests”.

#### 7.1.1. Política de acceso ás páxinas web

Como resulta lóxico non todo o mundo pode acceder a tódalas páxinas da aplicación, algunas delas están reservadas para o administrador, outras para o propio usuario logueado, unhas terceiras para calquera usuario logueado e por último hai páxinas de dominio público.

É importante de cara a non cometer errores de seguridade, que este correcto comportamento sexa comprobado, así na táboa excel que se atopa no ficheiro docs/urlsMap.ods pódese observar con detalle que política de acceso segue cada unha das páxinas da aplicación segundo a súa URL. Foi a partir deste mapa de urls dende onde se elaborou a base dos test de unidade para o acceso ás páxinas, que comprobar para cada unha das páxinas da aplicación que un usuario cos permisos adecuados poida acceder e que calquera dos demais reciba o erro axeitado.

### 7.1.2. Probas da Capa Web con Javascript

As probas da capa web escrita en javascript apoiaranse no framework Qunit de jQuery que proporciona un xeito sinxelo de crear probas unitarias sempre e cando o código javascript estea convintemente separado do HTML que forma a vista da capa web.

Como resulta lóxico, estas probas estarán escritas en javascript e almacenadas no directorio do proxecto “/src/static/site/tests”, podendo executarse de dous xeitos diferentes: Ou ben como unha páxina web pertencente á aplicación (isto favorece o desenvolvemento ágil), ou ben como unha proba das realizadas polo comando:

```
python manage.py test
```

Para poder executar un código javascript dende os tests da aplicación, precisase un lanzador ou runner que lance estes tests contra algún navegador de liña de comandos, neste caso a opción seleccionada foi a combinación do paquete django.js (v0.8.1) en combinación co navegador de liña de comandos phantomJS.

Django.js é un conxunto de utilidades que permiten a integración de código javascript en Django. máis en concreto neste proxecto empréganse aquelas que teñen que ver co testing de aplicacíons [39], destacando as clases QUnitSuite e PhantomJsRunner que se empregan para lanzar os tests como parte das probas da aplicación mediante a clase creada StaticJsTestCase, obtendo os resultados dos tests mentres que tamén se levanta o servidor para atender as posibles peticións do navegador de liña de comandos.

Por outro lado PhantomJS é un navegador WebKit de liña de comandos, cunha API Javascript que da soporte rápido e nativo para varios estándares web que resultan moi do noso interese, como son a manipulación DOM, os selectores CSS, JSON, Canvas e SVG. PhantomJS será chamado implícitamente polo runner de Django.js cando se executen os test, mentres que no caso da vista QUnitView os tests executaranse directamente no navegador que realice a petición.

Pero só con probas unitarias non se pode asegurar a calidade dos software xa que moitos dos módulos están pensados para interactuar entre eles, polo que fanse necesarias as Probas de Integración.

## 7.2. Probas de Integración

As probas de integración so aquelas que proban o funcionamento conxunto de varios módulos da aplicación, realizanse tras o éxito das probas unitarias tamén sen que haxa interacción humana.

No noso caso agruparemos as probas de integración nun paquete a parte, para evitar así mesturalas coas probas unitarias de cada módulo. A estes efectos creamos a clase SeleniumAncowebTest que extende StaticLiveServerTestCase engadindo ademais os métodos login\_user(self, user, password) e logout\_user(self, user) xa que todo los tests que comproben outros módulos precisando de un usuario logueado considéranse tests de integración.

### 7.2.1. Probas Funcionais Selenium

As probas funcionais son aquellas nas que se lle dita ao sistema cales serán as saídas a unha determinada serie de entradas co fin de comprobar que a funcionalidade é a correcta. No caso da nosa aplicación empregaremos tests funcionais para as probas de integración apoiándonos en Selenium.

Selenium é un framework para a realización de probas funcionais que permite lanzar un navegador e indicar as accións a realizar sobre el xunto cunha serie de comprobacións para verificar que estas accións provocan na páxina web o efecto desexado. Resultan de especial relevancia na programación web, xa que o seu functionamento asemellase

moito ao que un humano faría para comprobar o correcto funcionamento da web sendo por tanto moi intuitivo.

No noso caso comprobanse con Selenium, o logueado de Usuarios, a subida de vídeos e o listado de vídeos.

### 7.3. Probas de Sistema

As probas de sistema neste caso consisten na comprobación manual de que a páxina se desprega correctamente no entorno de producción. Para esta proba procedese subindo á plataforma mediante o dominio [www.ancoweb.es](http://www.ancoweb.es) dous vídeo diferentes: Un representativo de un entorno aberto e público e outro pertencente a un espazo interior, posteriormente comprobase que ambos vídeos foron subidos e analizados correctamente así como que as distintas componentes que interactúan na páxina de reproducción do vídeo funcionan de forma coherente co contido de este.

### 7.4. Probas de Aceptación

Por último están as probas de Aceptación fanse co obxectivo de comprobar se o software cumpre coas expectativas que o cliente tiña de el. A estes efectos cada vez que se finalizaba unha funcionalidade realizouse, acorde coa metodoloxía, unha proba completa por parte do titor Brais Cancela que garantise que todo o implementado era acorde co que se desexaba. Ocasionalmente o proxecto tamén foi revisado polos outros titores aportándolle así un carácter máis plural.

### 7.5. Calidade

Os parámetros de calidade empregados para a codificación do código fonte son os seguintes:

- Para o código **Javascript**: JavaScript Style Guide and Coding Conventions [40], tamén apoiado nas JavaScript Best Practices [41].

- Con respecto ao código **Python** seguiuse a convención PEP8 Style Guide for Python Code [42].



## Capítulo 8

# Planificación e Avaliación de Custes

A planificación e a avaliación de custes son de vital importancia nun proxecto destas características, pois aseguran que o proxecto de desenvolverá correctamente cumplindo con todas as especificacións desexadas. Neste caso a realización do proxecto abarca dende comezos de febreiro de 2015 cando teñen lugar as primeiras reunións ate setembro de ese mesmo ano no que finaliza a execución desta aplicación e a elaboración da memoria co obxectivo de presentar o conxunto do traballo a mediados de setembro.

Como se explicou no capítulo referente á metodoloxía, o proxecto subdividirase nunha serie de iteracións de aproximadamente un mes de duración<sup>1</sup> e trinta horas de traballo aproximadamente<sup>2</sup>, nas que se acometerán unha serie de funcionalidades concretas. Para o seguimento do avance do proxecto, o seu control e avaliación empregase a ferramenta YouTrack, que a parte de xestionar de forma ágil as tarefas a realizar, permite xerar informes de proxecto como os que se verán a continuación. A parte disto, por cada versión estable pódese atopar no repositorio de GitHub, unha Release co seu número de versión e o seu comentario asociado. As Releases pódense ver na seguinte

---

<sup>1</sup>As datas non son de un mes exacto xa que se adaptaron estes períodos en función da carga lectiva e de outras variables de carácter persoal para cumplir así coa faceta incremental do proxecto, facendo que cada nova Release aportase unha serie de características de interese.

<sup>2</sup>A excepción do último sprint pertencente ao mes de Agosto, no que posuía disponibilidade a tempo completo

lista:

- **v0.1 - Cargar e Visualizar vídeos** (1 febreiro - 1 marzo)
- **v0.2 - Subida e Conversión de vídeos** (1 marzo - 19 abril)
- **v0.3 - Xerar e mostrar deteccións básicas** (19 abril - 15 maio)
- **v0.4 - Arranxar bug's e mellorar a estratexia de probas** (15 maio - 11 xuño)
- **v0.5 - Traxectorias** (11 xuño - 10 xulio)
- **v0.6 - Detección do comportamento anómalo** (10 xulio - 24 xulio)
- **v1.0 - Memoria e posta en producción** (24 xulio - 30 agosto)

A primeira iteración do proxecto estivo principalmente centrada na formación sobre o framework Django, xa que o fin primordial era o de crear unha primeira páxina web que permitise subir e visualizar vídeos. Tamén estivo centrada en coñecer de preto as posibilidades ofertadas pola ferramenta GitHub para poder subir os avances realizados. Non se empregaba por tanto ningunha ferramenta para o seguimento de incidencias ou a integración continua.

Durante a segunda e terceira iteración cando xa se dominaban tanto Django como GitHub, procedeuse á busca dunha ferramenta para este seguimento de incidencias. Nun comezo, empregouse a propia ferramenta integrada en GitHub que permite un seguimento mínimo das tarefas e as metas a alcanzar, e xa na terceira iteración optouse de forma definitiva por YouTrack, migrando os issue's acumulados en GitHub a esta nova plataforma más completa. Durante este período tamén se foi configurando Travis CI como servidor de integración continua, inda que a ampla diversidade de linguaxes e tecnoloxías presentes no proxecto fixo que esta integración continua fallase eventualmente por motivos de configuración.

Na gráfica 8.1 podemos ver o fluxo acumulado de tarefas no cal se distinguen as tarefas **abertas**, **en curso** e **solucionadas**. A gráfica amosa un fluxo crecente de tarefas solucionadas correspondentes ás tarefas realizadas en cada iteración e tamén

unha serie de tarefas pendentes que corresponden as tarefas do Product Backlog que nunca chegaron a entrar en ningún Sprint por falta de prioridade e que conformarán a meirande parte das liñas futuras do capítulo 10.

Por outra parte o gráfico de evolución do proxecto (figura 8.2) marcou durante o desenvolvemento dos sprints a liña idónea e por tanto a velocidade en relación ao ritmo ideal. Nótese que a gráfica estima por número de tarefas e non por horas estimadas, isto explica as desviacións como as do mes de agosto, xa que ao dedicarse esta iteración praticamente á memoria e ser a memoria unha única tarefa, a gráfica permanece estanca áinda que se sigan a sumar horas de traballo de forma continua.

Nas figuras 8.3 e 8.4 pódense ver toda-las horas adicadas fronte ás horas previstas, xeralmente a predición é acertada, inda que como é lóxico non é cen por cen exacta.

Os datos do total de horas empregadas para a versión v1.0 suman 238 horas, que a un prezo de 12 €a hora fan un custo total de 2.856,00 €.

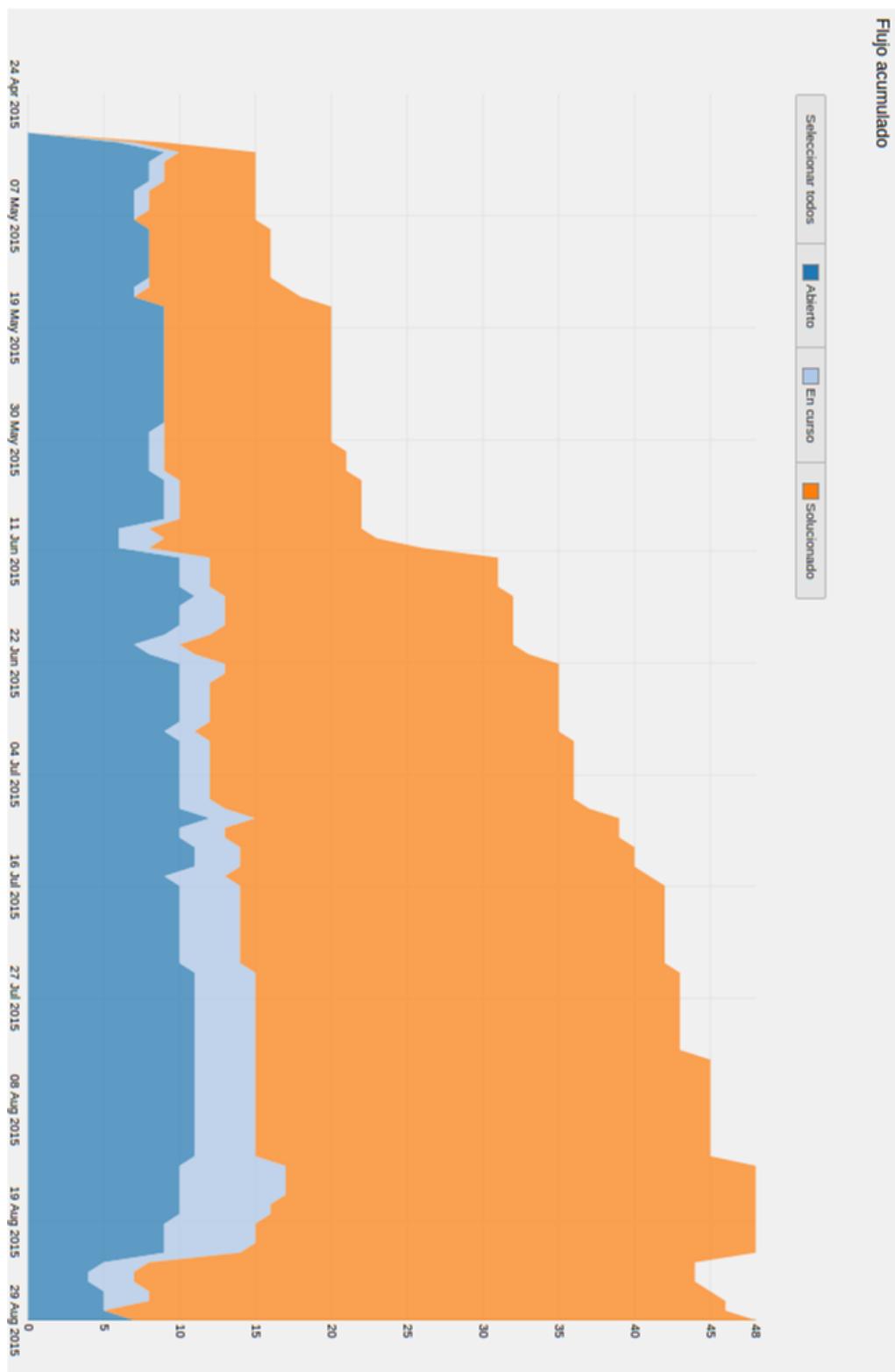


Figura 8.1: Gráfica do fluxo de tarefas acumulado ao longo do proxecto

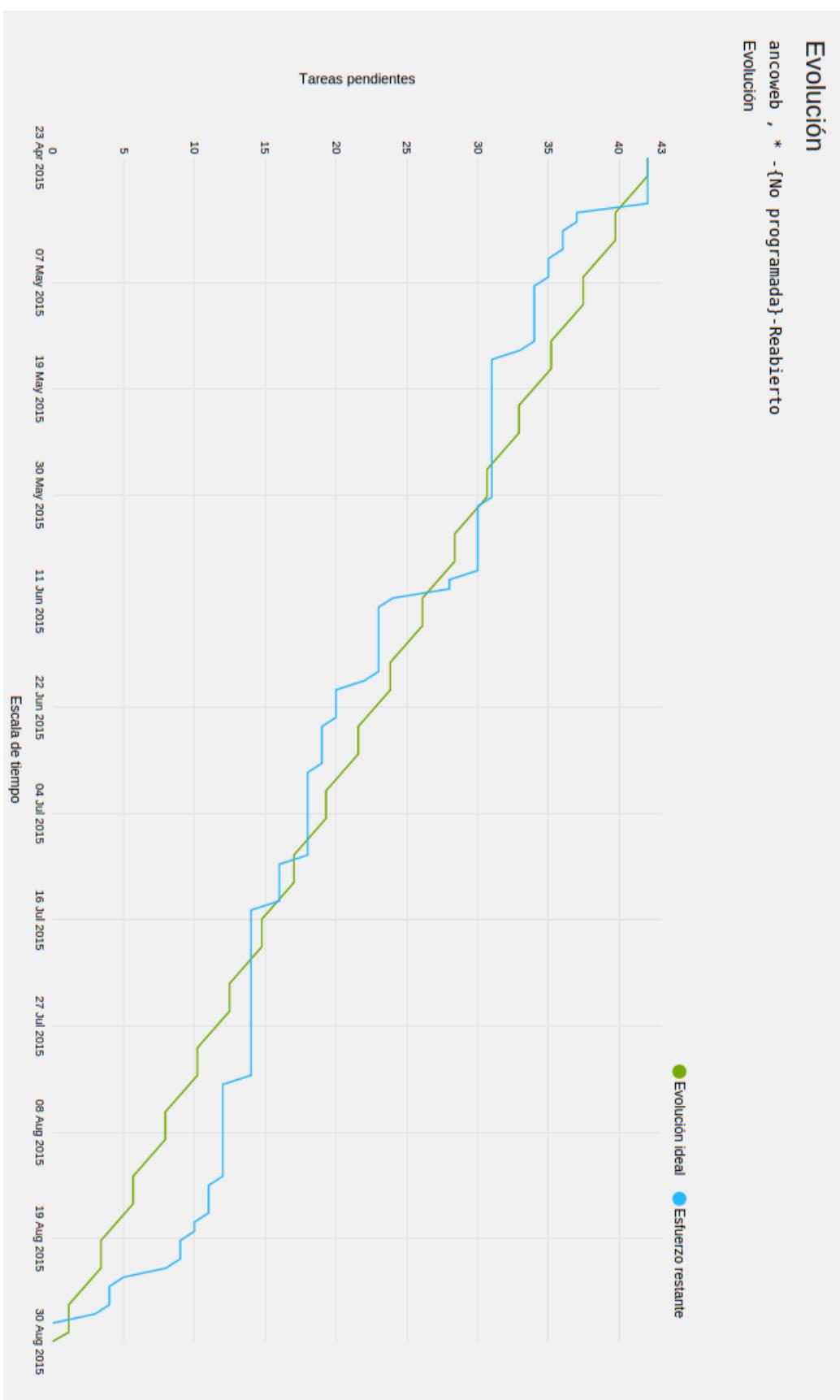


Figura 8.2: Gráfica da evolución do proxecto

		INCIDENCIAS , AGRUPAR POR SOLUCIONADO EN VERSIÓN(ES)	TIEMPO ESTIMADO	TIEMPO EMPLEADO
<b>v0.2 - Subida y conversión de vídeo completada</b>				
ancoweb-10	Boton Finish en Notificación	3h	16h	13h
ancoweb-12	Convertir Video a distintos formatos	5h		2h
ancoweb-52	Crear sistema de Notificaciones	8h		3h
<b>v0.3 - Xerar e mostrar elementos básicos do XML</b>				
ancoweb-13	Analizar las librerías de Brais	4h	21h 10m	26h 39m
ancoweb-11	Capa canvas sobre el video	3h		2h 47m
ancoweb-14	Create Esquema de un Programa para Shell	2h		4h
ancoweb-19	Eliminar Formato .ogv	10m		2h 30m
ancoweb-17	Ensamblar ambas aplicaciones	2h		7m
ancoweb-15	Implementar XmlParser	5h		3h
ancoweb-8	Rediseñar Autenticación	2h		8h 30m
ancoweb-9	Seguimiento en pantalla completa	3h		1h 45m
<b>v0.4 - Solventados bugs e mellorada a estratexia de probas</b>				
ancoweb-20	Boton de reanalisar video	4h	33h 30m	39h 25m
ancoweb-16	Cancelar Subida	3h		6h
ancoweb-2	Gestionar el comportamiento web durante el análisis	2h		1h
ancoweb-18	No mostrar la notificación en el panel de seleccionar imagen	30m		25m
ancoweb-23	Política de Acceso al panel	6h		10h
ancoweb-22	Testear sistema en otra máquina y arreglar los errores que surjan	12h		11h
ancoweb-21	Test capa Web	6h		10h

Figura 8.3: Táboa de horas adicadas ao proxecto por versión (1)

<b>v0.5 - Traectorias</b>							
ancoweb-28	Descargar Vídeo	2h	15m	36h 55m			
ancoweb-27	Descargar/Ver XML del análisis	2h	15m	15m			
ancoweb-31	Exportar trayectorias al fichero xml	10h	9h 45m	9h 45m			
ancoweb-29	Listar objetos detectados	6h	3h	3h			
ancoweb-36	Mostrar elementos detectados como recuadros	5h	9h 45m	9h 45m			
ancoweb-32	Mostrar trayectorias	5h	3h 40m	3h 40m			
ancoweb-6	Paginar la ListView de videos	1h	30m	30m			
ancoweb-34	Pintar de colores os Obxetos	4h	3h 45m	3h 45m			
ancoweb-25	Pruebas del código Javascript	10h	6h	6h			
<b>v0.6 - Detección do comportamento anómalo</b>							
ancoweb-40	Aclarar y Aleatorizar Colores de las detecciones	2h	1h 5m	1h 5m			
ancoweb-44	Añadir milisegundos a las fechas	1h	1h	1h			
ancoweb-47	Bug en PhantomJS	5h	4h 40m	4h 40m			
ancoweb-46	Cacular el tiempo entre la actualización de estado de la página Video Details	1h	45m	45m			
ancoweb-42	Compativilidad con otros navegadores	—	1h 45m	1h 45m			
ancoweb-33	Filtrar por grado de Anormalidad	10h	14h 45m	14h 45m			
ancoweb-39	Implementar patrón observador	6h	7h 15m	7h 15m			
ancoweb-43	Lanzar alerta con el video recortado	10h	13h 30m	13h 30m			
ancoweb-38	Utilizar hover sobre los objetos detectados	6h	2h	2h			
<b>v0.7 - Memoria y últimos detalles</b>							
ancoweb-48	Arreglar aspecto visual	5h	5h	5h			
ancoweb-50	Cobertura de los test de la capa web	10h	4h	4h			
ancoweb-45	Crear esquema XML	3h	2h 45m	2h 45m			
ancoweb-24	Elaborar Memoria	64h	51h 15m	51h 15m			
ancoweb-35	Fijar tamaño máximo de los videos	2h 30m	1h 30m	1h 30m			
ancoweb-49	Resolver problemas de Integración Continua	6h	10h	10h			
Tiempo total empleado: 237h 14m de 247h 10m							

Figura 8.4: Táboa de horas adicadas ao proxecto por versión (2)



## **Capítulo 9**

# **Resultados e Conclusíons**

Pódese concluir por tanto, que este proxecto é unha ferramenta útil na busca áxil de novos métodos, tanto de seguimento de obxectos como de análise de alto nivel.

O resultado ao que se chegou dista moito de ser o idóneo, pero sen ser perfecto é un avance significativo capaz de amosar as capacidades deste tipo de sistemas, sentando un precedente para que nun futuro próximo se siga a mellorar esta aplicación de cara a dar visibilidade a estes traballos de investigación que moitas veces teñen tan complicado amosar a potencia dos seus resultados.



# Capítulo 10

## Liñas Futuras

En canto ás liñas futuras da aplicación, estas están compostas en parte por incidencias do Product Backlog que por falta de tempo ou por ter unha baixa prioridade quedaron como tarefas abertas no sistema de Xestión de Incidencias, e tamén en parte por outras funcionalidades que se consideran de interese no futuro pero que en ningún momento chegaron a entrar no Product Backlog polas súas esixencias en tempo.

### 10.1. Incidencias rexistradas como abertas ou re-abertas en YouTrack

- **Cambiar as sentencias Popen shell='TRUE':** Popen é unha clase empregada dende o código Python do servidor para executar dende a terminal unha acción determinada, pero ao executar co parámetro shell=TRUE, a información de se se produciu algúin erro perdeuse. Deberíase empregar co modo shell=False inda que isto require de varias adaptacións no código.
- **Internacionalización:** Estaría ben traducir a aplicación a Español e Galego.
- **Cancelar Subida:** Actualmente existe un bug que non permite cancelar unha subida mentres esta está en proceso, o problema está asociado ao uso de thread's debido ao acceso a Base de datos.
- **Modificar o algoritmo de actualización da barra de progreso:** En un

futuro, deberíase modificar o algoritmo de progreso de unha notificación de subida para intentar minimizar o número de consultas AJAX ao servidor, de xeito que se incremente o retardo entre peticións cando o avance de barra de progreso sexa lento e se acelere este cando o avance sexa veloz.

- **Mostrar Potencial:** Unha funcionalidade futura sería a de mostrar en cada momento cales son os puntos polo que pasan máis obxectos, isto está almacenado no sistema de recoñecemento como unha matriz que contén o potencial de cada punto da imaxe, e a tarefa consistiría en amosar estes potenciais nunha capa de información superposta ao vídeo.
- **Mostrar velocidad:** Mostrar a velocidade de cada un dos obxectos en escena así como permitir filtrar por un límite de velocidade ao igual que se fai co comportamento anómalo.
- **Cobertura dos test da capa web:** Esta tarefa foi desprazada a unha rama a parte, xa que o esforzo para a súa configuración é cuantioso e dada a súa baixa prioridade non se chegou a completar. A solución pasa por empregar algúns ferramentas como blanketJS que recolla a cobertura en phantoJS, e logo enviar esta ao servidor de coveralls.
- **Error ao subir dous vídeos simultaneamente** Revisar esta casuística na que se soben dous vídeos simultaneamente pois en certos momentos parece ter un comportamento erróneo.
- **Illar Tracking da análise de alto nivel:** De cara a facilitar a investigación estaría ben poder realizar solo o análise de alto nivel proporcionando a lista de deteccións, ou simplemente empregar a análise de baixo nível para detectar obxectos.
- **Engadir compatibilidade con IE:** Actualmente a aplicación non é compatible con Internet Explorer, en un futuro estaría ben soportar este navegador.

## 10.2. Outras funcionalidades de interese

### 10.2.1. Vídeo en Directo

De cara a un futuro a funcionalidade máis desexada é a de facer o análise en directo empregando o vídeo procedente dunha cámara. Pero isto prantexa unha serie de retos importantes como é a sincronización do vídeo co tempo de análise ou o envío dos datos desta análise á capa web, xa que isto rompe co modelo HTTP de peticións a un servidor.

A solución para o problema da sincronización entre o vídeo e o tracking, pasa por retardar o envío do vídeo unha determinada cantidade de tempo, suficiente como para que se procese e se faga a análise.

Non obstante como o tempo de análise non se pode prever de forma exacta, habería necesariamente que facer axustes para que cando a análise fose máis veloz que a reprodución do vídeo, a capa web mantese un buffer cos resultados recibidos e os amosase no momento correcto. E por outra parte se o vídeo evolucionase máis rápido que a análise, o lado servidor debería reducir o número de fotogramas a analizar descartando por exemplo un de cada dous ou de cada tres fotogramas.

Para o problema do envío dos datos á capa web, o máis lóxico podería ser empregar unha modificación do protocolo RTSP (Real-Time Streaming Protocol) visto na sección 2.4.3, de xeito que por un lado se envíe o vídeo en directo, e por outra parte toda a información precisa para a súa reproducción máis os datos en XML sobre as deteccións.

### 10.2.2. Panel de Control

Se se chegase a desenvolver un módulo que permita streaming de vídeo en directo sería de gran interese ter un panel no que se poidan visualizar as imaxes procedentes de distintas cámaras simultaneamente. Inda que isto tamén podería facerse integrando a aplicación con algunha outra aplicación de vídeo-vixilancia open-source como as citadas no apartado 3.2.



# **Apéndice A**

## **Apéndice**

- IDE
- BD

### **A.1. Notas acerca da Terminoloxía**

- Scrum Team
- Product Owner
- Development Team
- Scrum Master
- Spring
- Sprint Planning Meeting
- Sprint Goal
- Daily Scrum
- Sprint Review
- Sprint Retrospective
- Product Backlog

- Sprint Backlog

# Bibliografía

- [1] Barizo, B. C., *Understanding target trajectory behavior: a dynamic scene modeling approach.* PhD thesis, Universidade da Coruña, 2015.
- [2] “Páxina web do proxecto cyeweb.” Dispoñible en: [http://www.novosun.com/Online\\_Manual/en-US/CyeWeb/Main\\_control.aspx](http://www.novosun.com/Online_Manual/en-US/CyeWeb/Main_control.aspx).
- [3] “Páxina web do proxecto aventura - cerebrus intelligent video analytics.” Dispoñible en: <http://www.aventuracctv.com/Intelligent-Video-Analytics-Software/>.
- [4] “Páxina web da empresa nolous sobre a análise do comportamento humano.” Dispoñible en: <http://www.noldus.com/human-behavior-research>.
- [5] “Páxina web do proxecto winanalyze, sobre seguimento de puntos de interese.” Dispoñible en: <http://winanalyze.com/>.
- [6] “Páxina web das publicacións científicas asociadas coa empresa mikromak.” Dispoñible en: <http://www.mikromak.com/en/downloads.htm>.
- [7] “Proxecto huygens object tracker da empresa scientific volume imaging.” Dispoñible en: <https://svi.nl/ObjectTracker>.
- [8] “Páxina web do proxecto ispy.” Dispoñible en: <http://www.ispyconnect.com/>.
- [9] “Plugins de visión por computador para ispy.” Dispoñible en: <http://www.iplugs.eu/iplugs/>.

- [10] “Páxina web do proxecto zoneminder.” Dispoñible en: <http://www.zoneminder.com/>.
- [11] “Guía para a detección de movemento en zoneminder.” Dispoñible en: [http://www.zoneminder.com/wiki/index.php/How\\_to\\_setup\\_motion\\_detection#Configure\\_motion\\_detection\\_areas](http://www.zoneminder.com/wiki/index.php/How_to_setup_motion_detection#Configure_motion_detection_areas).
- [12] “Páxina web do proxecto swistrack.” Dispoñible en: <https://en.wikibooks.org/wiki/SwisTrack>.
- [13] “Páxina web do proxecto community core vision.” Dispoñible en: <http://ccv.nuigroup.com/#home>.
- [14] “Páxina web do proxecto bio-traking.” Dispoñible en: <http://www.bio-tracking.org/>.
- [15] “Páxina web do proxecto ctrax.” Dispoñible en: <http://ctrax.sourceforge.net/>.
- [16] Schwaber, K. and Sutherland, J., *La Guía Definitiva de Scrum: Las Reglas del Juego*. Scrum.Org and ScrumInc, 2013.
- [17] “Red5 web server on github.” Dispoñible en: <https://github.com/Red5/red5-server>.
- [18] “Web toolkit for c++ page.” Dispoñible en: <http://www.webtoolkit.eu/wt>.
- [19] “Icecast server on github.” Dispoñible en: <https://github.com/paluh/icecast>.
- [20] “Python subprocess module.” Dispoñible en: <https://docs.python.org/3.4/library/subprocess.html>.
- [21] “Django web page.” Dispoñible en: <https://www.djangoproject.com/>.
- [22] “W3schools html5 video tag web page.” Dispoñible en: [http://www.w3schools.com/html/html5\\_video.asp](http://www.w3schools.com/html/html5_video.asp).

- [23] “Html audio and video dom reference.” Disponible en: [http://www.w3schools.com/tags/ref\\_av\\_dom.asp](http://www.w3schools.com/tags/ref_av_dom.asp).
- [24] “Kevin roast canvas examples.” Disponible en: <http://www.kevs3d.co.uk/dev/index.html>.
- [25] “Recomendáns de jquery-qunit á hora de escribir código testeable.” Disponible en: <http://qunitjs.com/intro/#make-things-testable>.
- [26] “Páxina web da librería image picker empregada no proxecto.” Disponible en: <http://rvera.github.io/image-picker>.
- [27] “Compoñente slider de jquery ui.” Disponible en: <http://api.jqueryui.com/slider/>.
- [28] “Páxina web do plugin para jquery tablesorster.” Disponible en: <http://tablesorter.com/docs/>.
- [29] “Document type definition w3schools tutorial.” Disponible en: [http://www.w3schools.com/xml/xml\\_dtd\\_intro.asp](http://www.w3schools.com/xml/xml_dtd_intro.asp).
- [30] “Validador xml da w3schools.” Disponible en: [http://www.w3schools.com/xml/xml\\_validator.asp](http://www.w3schools.com/xml/xml_validator.asp).
- [31] “Páxina en github do proxecto edge.” Disponible en: <https://github.com/arocks/edge>.
- [32] “Proxecto django-progressbarupload publicado por ouhouhsami.” Disponible en: <https://github.com/ouhouhsami/django-progressbarupload>.
- [33] “Clase temporaryfileuploadhandler.” Disponible en: [https://docs.djangoproject.com/en/1.8/\\_modules/django/core/files/uploadhandler/#TemporaryFileUploadHandler](https://docs.djangoproject.com/en/1.8/_modules/django/core/files/uploadhandler/#TemporaryFileUploadHandler).
- [34] “W3schools html5 source tag web page.” Disponible en: [http://www.w3schools.com/tags/tag\\_source.asp](http://www.w3schools.com/tags/tag_source.asp).

- [35] Pleiffer, S., *Vídeo con HTML5*. ANAYA, 2011.
- [36] “Páxina de referencia sobre os obxectos data uri en javascript.” Dispoñible en: [https://developer.mozilla.org/en-US/docs/Web/HTTP/data\\_URI](https://developer.mozilla.org/en-US/docs/Web/HTTP/data_URI).
- [37] “Páxina da empresa xestora de dominios 1and1.” Dispoñible en: <https://www.1and1.es/>.
- [38] “Páxina da empresa de hosting digitalocean.” Dispoñible en: <https://www.digitalocean.com/>.
- [39] “Ferramentas para a realización de probas de django.js.” Dispoñible en: <http://djangojs.readthedocs.org/en/latest/test.html#test-cases>.
- [40] “Javascript style guide and coding conventions.” Dispoñible en: [http://www.w3schools.com/js/js\\_conventions.asp](http://www.w3schools.com/js/js_conventions.asp).
- [41] “Javascript best practices.” Dispoñible en: [http://www.w3schools.com/js/js\\_best\\_practices.asp](http://www.w3schools.com/js/js_best_practices.asp).
- [42] “Pep8 style guide for python code.” Dispoñible en: <https://www.python.org/dev/peps/pep-0008/>.