



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, 18 de xullo 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 increíbles 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 vólvese 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 encárgase de detectar a todos aqueles obxectos ou persoas que aparecen nunha secuencia de vídeo, aplicando un algoritmo para medir o 'grao de anormalidade' da súa conduta en base aos movementos que realizan.

Palabras chave

Análise Comportamento, Comportamento Humano, Aplicación Web, Django.

Índice xeral

1. Introducción e Obxectivos	1
2. Fundamentos Teóricos e Conceptos Previos	3
2.1. Análise do comportamento	3
2.1.1. OpenCV	4
2.2. Extensible Markup Language (XML)	5
2.3. Programación Web	5
2.3.1. Desenvolvemento Áxil	6
2.3.2. Soporte para transaccións	6
2.3.3. Object-Relational Mapping (ORM)	6
2.3.4. Xestión de Layout	7
2.3.5. Outras cuestións da web	7
3. Análise de antecedentes e alternativas	9
4. Metodoloxía seguida no desenvolvemento de Proxecto	11
4.1. Persoas	11
4.2. Reunións	12
4.3. Control de Versións con GitHub	13
4.4. Integración Continua con Travis CI	13
4.5. Control da cobertura con Coveralls	15
4.6. Xestión de Incidencias e Control de Proxecto con YouTrack	15

5. Estudo das tecnoloxías para a creación de aplicacións web baseadas en vídeo	17
5.1. Back-End	18
5.1.1. Java	18
5.1.2. C#	18
5.1.3. C - C++	18
5.1.4. Python + Django	19
5.2. Front-End	19
5.2.1. Vídeo en Flash	19
5.2.2. Vídeo HTML5 + Javascript	20
5.3. Título sección	21
5.3.1. Título subsección	21
5.3.2. Título subsección	22
6. Funcionalidades Destacadas	25
6.1. Control de Usuarios	25
6.2. Reprodución de Vídeo	25
6.3. Carga de Vídeo	25
6.4. Análise de Vídeo	25
6.5. Detección de Obxectos	25
6.6. Traxectorias	25
6.7. Comportamento anormal	25
6.7.1. Título	25
7. Probas Realizadas	27
7.1. Probas Unitarias	27
7.1.1. Probas Automáticas	27
7.2. Probas de Integración	27
7.2.1. Probas funcionais Selenium	27
7.3. Probas de Sistema	27
7.4. Probas de Aceptación	27

8. Calidade	29
9. Planificación e Avaliación de Custes	31
10.Resultados e Conclusións	33
11.Liñas Futuras	35
11.1. Vídeo en Directo	35
A. Título	37
A.1. Lista de Acrónimos	37
A.2. Manual de Usuario	37
A.3. Manual de referencias Técnicas	37
A.4. Notas acerca da Terminoloxía	37

Índice de figuras

2.1. Clásica arquitectura dunha aplicación web empresarial	6
4.1. Imaxe de parte do ficheiro .travis.yml	14
5.1. Esta es la etiqueta de la figura	22
5.2. Esta es la etiqueta de la figura con borde	22

Índice de Táboas

5.1. Tabla de ejemplo	23
---------------------------------	----

Capítulo 1

Introdución e Obxectivos

O seguimento de obxectos é o proceso de estimar no tempo a localización de un ou mais 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 estudado. 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, propónse 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.

A arquitectura web baseada no modelo cliente-servidor consiste nun lado servidor que distribúe os recursos como poden ser o contido multimedia ou as páxinas web ao outro lado, o cliente, que típicamente corre nun navegador web interpretando as páxinas html e o código javascript asociado a estas.

Seleccionase pois esta arquitectura xa que a diferencia das arquitecturas de escri-

torio, proporciona ás persoas que acceden á web independencia do Sistema Operativo empregado e acceso 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

En este capítulo se exponen los fundamentos teóricos en los que se basa el proyecto o que se utilizan en él. Normalmente cada proyecto trata de una temática, de la que hay una serie de conocimientos teóricos relacionados y que deben incluirse en la memoria del proyecto, para aportarle el carácter científico que conlleva todo trabajo académico. Se requiere una explicación detallada de todo lo necesario, para dar una visión profunda de lo que debe conocerse para afrontar la realización del proyecto. Por una parte sirve como exposición de los elementos científicos en los que se basa el proyecto, pero a la vez, como explicación de lo que se ha estudiado para la elaboración del mismo.

2.1. Análise do comportamento

É un dos campos de investigación mas activos hoxe en día. A idea principal deste tipo de sistemas é a de detectar calquera acción levada a cabo polos obxectos involucrados. Un obxecto é calquera cousa que debe ser seguida, polo que dependendo do tipo de problema estes obxectos poden ser dunha natureza ou doutra. Por outra parte, o tipo de accións depende do que o sistema trate de detectar, xa poden ser comportamentos individuais(camiñar, correr, loitar...) ou grupais (reunirse, abandonar un grupo de

persoas...).

Á súa vez os sistemas de análise de comportamento poder dividirse en tres técnicas diferentes[1]:

- **Detección:** A partires de unha secuencia de vídeo como entrada obtén os distintos obxectos que aparecen en cada fotograma da escena. Para este fin empréganse técnicas de visión por computador.
- **Seguimento:** A partires da información obtida na detección, asígnanselle identificadores a cada obxecto detectado no vídeo, agrupando se procede distintos obxectos baixo o mesmo identificador, se se considera que estes obxectos forman parte de un grupo ou unha mesma detección.
- **Análise do comportamento de Alto Nivel:** A partires da información obtida nos dous pasos anteriores pódese catalogar o comportamento de cada detección empregando técnicas de recoñecemento de patróns.

Os resultados mais 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
- A velocidade de un obxecto determinado.

2.1.1. 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 requiren recoñecemento de obxectos.

OpenCV é multiplataforma, existindo versión para GNU/Linux, Mac OS X e Windows. Contén mais 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.

A aplicación a desenvolver ten que garantir a compatibilidade cos algoritmos creados sobre OpenCV. Isto farase definindo unha interface de liña de comando e un formato de ficheiro XML (Extensible Markup Language) no que as aplicación de recoñecemento deben escribir os datos da súa análise.

2.2. Extensible Markup Language (XML)

XML é unha linguaxe de marcas desenvolvida polo World Wide Web Consortium (W3C) e empregado para almacenar datos de forma clara e lexible. Permite definir a gramática de linguaxes específicas para estruturar así grandes documentos. XML é especialmente útil para comunicar varias aplicacións que traballan en tecnoloxías diferentes, xa que grazas á súa simplicidade é extremadamente sinxelo integrar os datos.

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. [2]

Cando un documento en formato XML segue as directrices definidas no ficheiro DTD asociado, dise que este ficheiro esta ben formado(well formed en ingles), e para validar iso empregarase na elaboración do traballo algún avaliador de XML en liña como por exemplo o da W3Chools.[3]

2.3. Programación Web

A programación web de Aplicacións 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 aplicacións web pódese ver no gráfico 2.1

Os puntos claves da programación clave son:

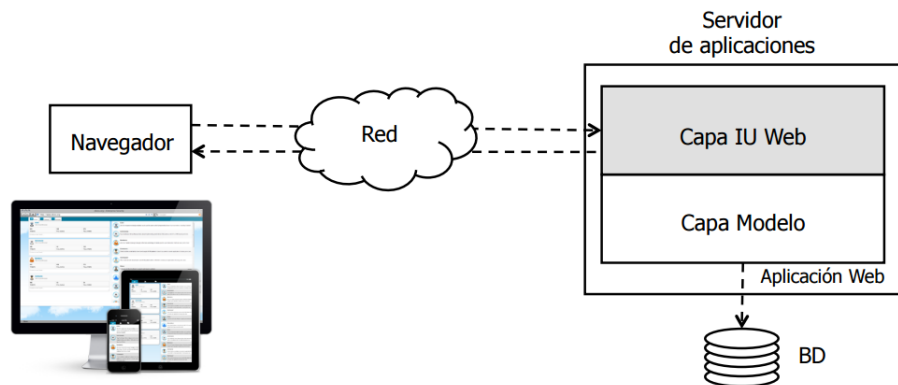


Figura 2.1: Clásica arquitectura dunha aplicación web empresarial

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.

2.3.2. 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 webs debido á inestabilidade da rede, e á concorrencia dos distintos clientes conectados.

2.3.3. Object-Relational Mapping (ORM)

Os mapeado obxecto-relación, é unha técnica de programación para converter datos de unha linguaxe Orientada a Obxectos (OO) a datos de un sistema relacional no que son persistidos e viceversa. Dispoñer de un mapeador obxecto-relacional ben integrado coa tecnoloxía a empregar minimiza drasticamente o tempo preciso para o desenvolvemento das funcionalidades.

2.3.4. Xestión de Layout

Tamén resulta moi practico dispoñer dunha linguaxe de prantilla que permitan xerar contido html dinamicamente. Algúns exemplos son o Sistema de Templates de Django, o Sistema JSP de Spring, a librería Thymeleaf ou os compoñentes de ASP.NET. Todos eles axudan a xerar contido HTML de xeito sinxelo e escalable.

2.3.5. Outras cuestións da web

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

Capítulo 3

Análise de antecedentes e alternativas

Se trata de realizar un estudio de alternativas o “estado del arte” o un análisis comparativo de alternativas.

Se exponen las diferentes alternativas que se han evaluado o que se consideran de interés, a lo realizado en el proyecto. Fundamentalmente se trata de otras herramientas existentes que realizan algo similar, sean o no comerciales, o de prototipos de investigación relacionados, o de estudios que tratan aspectos similares.

Buscar por internet produtos que fagan algo similar...

Capítulo 4

Metodoloxía seguida no desenvolvemento de Proxecto

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

Esta metodoloxía áxil tamén chamada melé pola súa inspiración no Rugby, permite un desenvolvemento rápido en situacións de requisitos inestables. Apoiase no seu carácter iterativo e incremental, dividindo 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). 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. Persoas

Os tres papeis que se definen nesta metodoloxía [4] foron adaptados do seguinte xeito:

- **ProductOwner:** O papel de ProductOwner, que define os requisitos da aplicación estivo representado polo director de proxecto Brais Cancela, que participou na creación do Anteprojecto. 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.
- **ScrumMaster e Development Team:** Ambos papeis leváronse a cabo polo autor, xa que carece de sentido definir ambas figuras nun equipo de unha soa persoa. De este xeito á par que se desenvolvía o proxecto, íase asegurando o cumprimento das regras de SCRUM.

4.2. Reunións

As reunións pola súa parte modifícanse do seguinte xeito:

- **Sprint Planning Meeting:** Esta reunión mantén o mesmo formato que no SCRUM puro, 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.
- **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 menos ao dunha 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.
- **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, planifícase a videira Iteración.
- **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.

4.3. Control de Versións con GitHub

Os sistemas de control de versións permiten a xestión dos distintos cambios efectuados sobre un produto software ou sobre a súa configuración. Facilitando a administración das distintas versións do produto.

En concreto, GitHub é unha plataforma de desenvolvemento colaborativo que emprega o sistema de control de versións Git. Escolleuse 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.

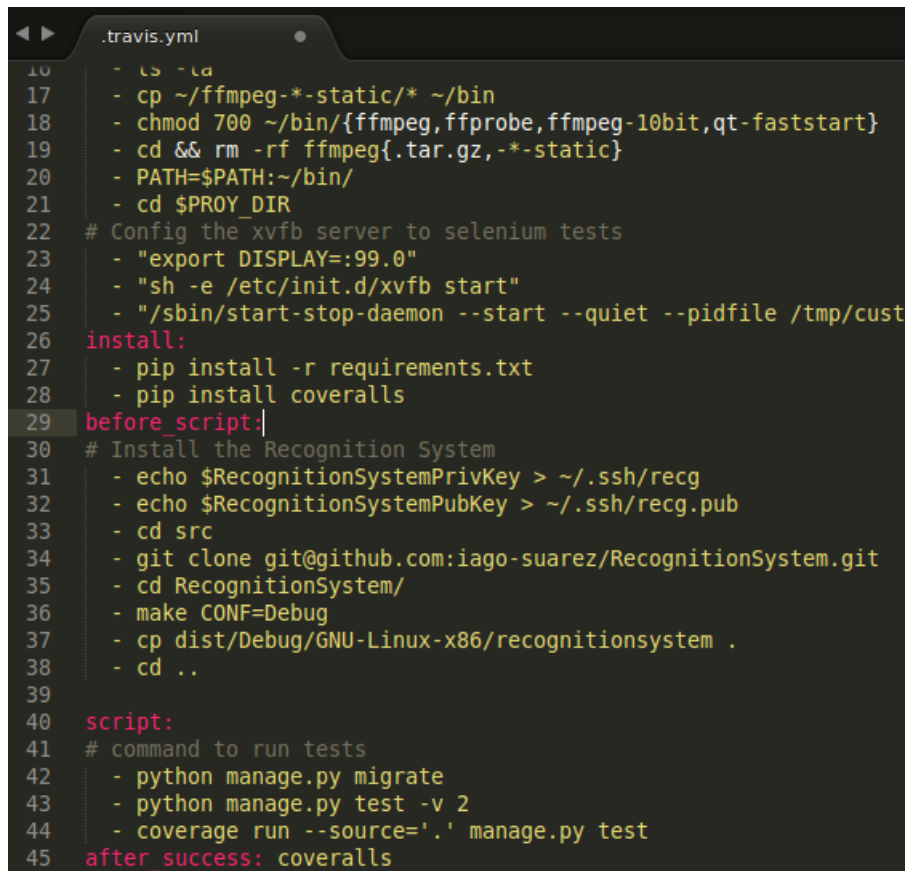
A páxina do proxecto é: <https://github.com/iago-suarez/ancoweb-TFG>

4.4. Integración Continua con Travis CI

A Integración Contínua (CI do ingles Continuous Integration) é un modelo informático que consiste en facer integracións automáticas dun proxecto o mais 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` 4.1, que contén as accións a realizar antes, durante e tras a as probas.

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

A screenshot of a code editor showing a portion of a .travis.yml file. The file is named ".travis.yml" and is located in a directory. The code is written in YAML format and includes several sections: a list of commands for installing FFmpeg, a section for configuring the xvfb server for Selenium tests, an "install:" section for installing dependencies, a "before_script:" section for installing the Recognition System, a "script:" section for running tests, and an "after_success:" section for running coveralls. The code is color-coded with syntax highlighting.

```
16 - ls -la
17 - cp ~/ffmpeg-*-static/* ~/bin
18 - chmod 700 ~/bin/{ffmpeg,ffprobe,ffmpeg-10bit,qt-faststart}
19 - cd && rm -rf ffmpeg{.tar.gz,-*-static}
20 - PATH=$PATH:~/bin/
21 - cd $PROY_DIR
22 # Config the xvfb server to selenium tests
23 - "export DISPLAY=:99.0"
24 - "sh -e /etc/init.d/xvfb start"
25 - "/sbin/start-stop-daemon --start --quiet --pidfile /tmp/cust
26 install:
27 - pip install -r requirements.txt
28 - pip install coveralls
29 before_script:
30 # Install the Recognition System
31 - echo $RecognitionSystemPrivKey > ~/.ssh/recg
32 - echo $RecognitionSystemPubKey > ~/.ssh/recg.pub
33 - cd src
34 - git clone git@github.com:iago-suarez/RecognitionSystem.git
35 - cd RecognitionSystem/
36 - make CONF=Debug
37 - cp dist/Debug/GNU-Linux-x86/recognitionssystem .
38 - cd ..
39
40 script:
41 # command to run tests
42 - python manage.py migrate
43 - python manage.py test -v 2
44 - coverage run --source='.' manage.py test
45 after_success: coveralls
```

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

4.5. Control da cobertura con Coveralls

4.6. Xestión de Incidencias e Control de Proxecto con YouTrack

Capítulo 5

Estudo das tecnoloxías para a creación de aplicacións web baseadas en vídeo

A continuación mostrase un estudo feito sobre as tecnoloxías mais empregadas hoxe en día de cara ao deseño de aplicacións web baseadas en vídeo. Os criterios que se valoraron á hora de escoller unha destas metodoloxías son os seguintes:

- **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 aquilo que permita axilidade e flexibilidade.

5.1. Back-End

O Back-End é a parte que procesa os datos proporcionados polo Front-End, atende as súas peticións, e xestiona o modelo de datos e os procesos implicados na aplicación. Neste caso en concreto, o Back-End será o encargado de interactuar directamente co sistema de Análise de Vídeo.

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

5.1.1. Java

Java é unha das linguaxes mais 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. A 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 estudados para ver o seu funcionamento é Red5 [5].

5.1.2. 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.

5.1.3. C - C++

C++ presenta como era de esperar a maior compatibilidade co algoritmo implementado, non obstante, inda que existen algunhas utilidades que facilitan o desenvolvemento

web con esta linguaxe como Wt (Web Toolkit)[6], o grado de axilidade está moi por baixo do que facilitan o resto das combinacións. Tamén pode resultar de interese o coñecido proxecto Icecast[7], que fai streaming de vídeo sobre unha interface web.

5.1.4. Python + Django

Python presentase como a mellor opción para desenvolver o lado servidor, por unha parte dispón do módulo Subprocess[8] que permite executar calquera comando pola terminal maximizando así a modularidade e a integración co algoritmo en C++. Por outra parte Django[9] 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.

5.2. Front-End

É a parte do Sistema encargada de interactuar co usuario. O maior requisito de esta parte é ser capaz de representar todas as capas de información complexa que xera o Back-End. E tratándose dunha aplicación web estudáranse aqueles xeitos que permitan a reprodución de vídeo, e a maiores o debuxado de figuras en movemento sobre este vídeo.

5.2.1. Vídeo en Flash

Video Flash é a tecnoloxía de reprodución de vídeo mais 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, e á vez 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 estudados son 100 % libres, Flash é un programa propietario para

o que é preciso adquirir unha licencia.

5.2.2. Vídeo HTML5 + Javascript

Está é unha das combinacións mais empregadas actualmente, xa que segue o estándar do W3C (World Wide Web Consortium)[10]. Nel defínese 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.

Para a modificación da súa aparencia pódese empregar *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[11] que poden empregarse dende código *Javascript*.

Isto resulta de moita utilidade, xa que 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[12] despexan toda dúbida posíbel.

Tamén é de interese a librería de estilos web *Twiter Bootstrap* [13] para axilizar o deseño web con CSS3, Así como a librería de Javascript *JQuery* [14] que minimiza de xeito considerable as liñas de código a implementar.

Esta es una referencia a un artículo [15].

Esta es una referencia a un capítulo dentro de un libro [16].

Esta es una referencia a un libro [17].

Esta es una referencia a un artículo dentro de los proceedings de un congreso [18].

Esta es una referencia a una url [19].

Esto está escrito en negrita

Esto está en enfatizado

Este texto está centrado

Esto es una lista:

- Primer elemento
- Segundo elemento
- Lista dentro de otra lista:
 - Primer subelemento
 - Segundo subelemento

5.3. Título sección

Esto es una descripción:

Palabra descripción de la palabra

Palabra descripción de la palabra

Y esto una lista numerada:

1. Elemento
2. Elemento
3. Elemento
4. Elemento

Podemos incluir una figura y referenciarla de esta forma 5.1. Además podemos poner la página en la que está: 22

5.3.1. Título subsección

Existen varias formas de incluir ecuaciones matemáticas. Las más utilizadas son las siguientes:



Figura 5.1: Esta es la etiqueta de la figura



Figura 5.2: Esta es la etiqueta de la figura con borde

- Para introducir expresiones matemáticas en el texto, se utiliza como delimitador el símbolo del dólar. Por ejemplo $a \rightarrow b$.
- Para introducir ecuaciones matemáticas, se utiliza el entorno `equation`:

$$\gamma = \frac{\bar{\alpha}}{\sqrt{\beta}} \quad (5.1)$$

$$E(v) = \int_0^1 \int_0^1 \int_0^1 E_{int}(v(r, s, t)) + E_{ext}(v(r, s, t)) dr ds dt \quad (5.2)$$

Un subíndice se especifica con el guión bajo y un superíndice con el circunflejo. Si el super/subíndice contiene varios caracteres, estos deben estar delimitados por llaves. Consultar el manual para comprobar como se pueden introducir símbolos y expresiones matemáticas en latex.

5.3.2. Título subsección

Esto es una subsección¹.

Así introducimos texto sin ningún tipo de formato latex:

```
4 drwxr-xr-x  2 noelia imagen    4096 2005-09-12 12:09 figures
4 -rwxr--r--  1 noelia imagen    585 2005-09-12 16:56 Makefile
```

¹Así se hace una nota a pie de página

```

4 -rw-r--r--  1 noelia imagen    647 2005-09-12 17:38 memoria.aux
4 -rw-r--r--  1 noelia imagen   1011 2005-09-12 17:18 memoria.bbl
4 -rw-r--r--  1 noelia imagen   1171 2005-09-12 17:18 memoria.blg
16 -rw-r--r--  1 noelia imagen  13440 2005-09-12 17:38 memoria.dvi
4 -rw-r--r--  1 noelia imagen    412 2005-09-12 17:38 memoria.glg
4 -rw-r--r--  1 noelia imagen    188 2005-09-12 17:38 memoria.glo
4 -rw-r--r--  1 noelia imagen    241 2005-09-12 17:38 memoria.gls
4 -rw-r--r--  1 noelia imagen    299 2005-09-12 17:38 memoria.ist
4 -rw-r--r--  1 noelia imagen    283 2005-09-12 17:38 memoria.lof
12 -rw-r--r--  1 noelia imagen   8997 2005-09-12 17:38 memoria.log
4 -rw-r--r--  1 noelia imagen     87 2005-09-12 17:38 memoria.lot
108 -rw-r--r--  1 noelia imagen 103492 2005-09-12 17:18 memoria.pdf
280 -rw-r--r--  1 noelia imagen 278687 2005-09-12 17:18 memoria.ps
4 -rwxr--r--  1 noelia imagen   1887 2005-09-12 17:40 memoria.tex
4 -rw-r--r--  1 noelia imagen    735 2005-09-12 17:38 memoria.toc

```

Título subsubsección

La figura 5.1 muestra un ejemplo de tabla básica. Para crear una tabla se utiliza el entorno `tabular` dentro del flotante `table`.

Tras `\begin{tabular}` se declara el número de columnas de la tabla. Cada columna se especifica con una letra:

c Si el texto en la columna está centrado

l Si el texto en la columna está alineado a la izquierda

r Si el texto en la columna está alineado a la derecha

Táboa 5.1: Tabla de ejemplo

Fila 1 Col. 1 larálala	Fila 1 Col. 2 lara	Fila 1 Col. 3 lalalalalalalala
Fila 2 Col. 1	Fila 2 Col. 2	Fila 2 Col. 3

Así `\begin{tabular}{rrrr}` indica 4 columnas con texto alineado a la derecha y `\begin{tabular}{cl}` indica dos columnas, la primera centrada y la segunda alineada a la izquierda.

Entre `\begin{tabular}{...}` y `\end{tabular}` se escribe el contenido de la tabla. Las columnas se separan con `&` y las filas con `\\`. Un ejemplo de una fila con tres columnas sería el siguiente:

$$\text{aaa} \& \text{bbbb} \& \text{cccc} \quad \backslash\backslash^2.$$

Para incluir líneas horizontales existe el comando `\hline`. Tras `\begin{tabular}{...}` podemos incluir un `\hline`, pero **ojo**, en cada fila `\hline` siempre debe ir tras `\\`.

También es posible incluir líneas verticales, pero los manuales de estilo no lo aconsejan. Las líneas verticales se definen en la declaración con barras verticales. Por ejemplo, con `{|c|c|c}` se crearían 3 líneas verticales, una antes de la primera columna, otra entre la primera y la segunda columna, y la tercera, entre la segunda y tercera columna.

Dentro de las tablas podemos incluir expresiones matemáticas, texto enfatizado, negrita, etc. Es posible también fusionar varias filas o varias columnas dentro de una misma tabla. Para ello, consultar en manuales los comandos/paquetes `multirow` y `multicol`.

²Importante: debe coincidir el número de columnas en la declaración con el número de columnas que se escriben dentro de la tabla

Capítulo 6

Funcionalidades Destacadas

Bla bla bla

6.1. Control de Usuarios

6.2. Reproducción de Vídeo

6.3. Carga de Vídeo

6.4. Análise de Vídeo

6.5. Detección de Obxectos

6.6. Traxectorias

6.7. Comportamento anormal

Bla bla bla

6.7.1. Título

Bla bla bla

Título

Bla bla bla

Capítulo 7

Pruebas Realizadas

7.1. Pruebas Unitarias

7.1.1. Pruebas Automáticas

Estaría ben!

7.2. Pruebas de Integración

7.2.1. Pruebas funcionais Selenium

7.3. Pruebas de Sistema

7.4. Pruebas de Aceptación

Capítulo 8

Calidade

Os parámetros de calidade empregados para a codificación do código fonte son: JavaScript Style Guide and Coding Conventions[20] JavaScript Best Practices:[21] PEP8 Style Guide for Python Code: [22]

Capítulo 9

Planificación e Avaliación de Custes

Capítulo 10

Resultados e Conclusións

Podemos concluír que ...

Capítulo 11

Liñas Futuras

Aquí meter todas as tarxetas que queden pendentes + paridas varias...

11.1. Vídeo en Directo

Escribir aquí todo lo documentado del vídeo en streaming

Apéndice A

Título

Bla bla bla

A.1. Lista de Acrónimos

Bla bla bla

- IDE
- BD

A.2. Manual de Usuario

A.3. Manual de referencias Técnicas

A.4. 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

Se pueden escribir unas líneas en las que se justifique el motivo de la inclusión de términos en inglés, por ejemplo por formar parte de la jerga, ser de uso frecuente y estar completamente aceptados por comités científicos de la temática del proyecto, etc.; en otro caso deben buscarse traducciones adecuadas.

Bibliografía

- [1] Barizo, B. C., *Understanding target trajectory behavior: a dynamic scene modeling approach*. PhD thesis, Universidade da Coruña, 2015.
- [2] “Document type definition w3schools tutorial.” Available on: http://www.w3schools.com/xml/xml_dtd_intro.asp.
- [3] “Validador xml da w3schools.” Available on: http://www.w3schools.com/xml/xml_validator.asp.
- [4] Schwaber, K. and Sutherland, J., *La Guía Definitiva de Scrum: Las Reglas del Juego*. Scrum.Org and ScrumInc, 2013.
- [5] “Red5 web server on github.” Available on: <https://github.com/Red5/red5-server>.
- [6] “Web toolkit for c++ page.” Available on: <http://www.webtoolkit.eu/wt>.
- [7] “Icecast server on github.” Available on: <https://github.com/paluh/icecast>.
- [8] “Python subprocess module.” Available on: <https://docs.python.org/3.4/library/subprocess.html>.
- [9] “Django web page.” Available on: <https://www.djangoproject.com/>.
- [10] “W3schools html5 video web page.” Available on: http://www.w3schools.com/html/html5_video.asp.
- [11] “Html audio and video dom reference.” Available on: http://www.w3schools.com/tags/ref_av_dom.asp.

-
- [12] “Kevin roast canvas examples.” Available on: <http://www.kevs3d.co.uk/dev/index.html>.
 - [13] “Twitter bootstrap web page.” Available on: <http://getbootstrap.com/>.
 - [14] “Páxina oficial de jquery.” Available on: <https://jquery.com/>.
 - [15] Aamport, L. A., “The gnats and gnus document preparation system,” *G-Animal’s Journal*, 1986.
 - [16] Knuth, D. E., *Fundamental Algorithms*, ch. 1.2. Addison-Wesley, 1973.
 - [17] Knuth, D. E., *Seminumerical Algorithms*. Addison-Wesley, 1981.
 - [18] Oaho, A. V., Ullman, J. D., and Yannakakis, M., “On notions of information transfer in VLSI circuits,” in *Proc. Fifteenth Annual ACM* (Oz, W. V. and Yannakakis, M., eds.), no. 17 in All ACM Conferences, (Boston), pp. 133–139, Academic Press, Mar. 1983. This is a full INPROCEEDINGS entry.
 - [19] Comon, H., Dauchet, M., Gilleron, R., Jacquemard, F., Lugiez, D., Tison, S., and Tommasi, M., “Tree automata techniques and applications.” Available on: <http://www.grappa.univ-lille3.fr/tata>, 1997. release October, 1st 2002.
 - [20] “Javascript style guide and coding conventions.” Available on: http://www.w3schools.com/js/js_conventions.asp.
 - [21] “Javascript best practices.” Available on: http://www.w3schools.com/js/js_best_practices.asp.
 - [22] “Pep8 style guide for python code.” Available on: <https://www.python.org/dev/peps/pep-0008/>.