

Universidades de Burgos, León y
Valladolid

Máster universitario

Inteligencia de Negocio y Big Data en Entornos Seguros



Trabajo Fin de Máster

**Algo muy largo y no sé que más
puedo hacer aquí**

Presentado por José Miguel Ramírez Sanz
en Universidad de Burgos — 17 de abril
de 2020

Tutor: Dr. José Francisco Díez Pastor
Dr. Álar Arnaiz Gonzalez

Universidades de Burgos, León y Valladolid



Máster universitario en Inteligencia de Negocio y Big Data en Entornos Seguros

D. nombre tutor, profesor del departamento de nombre departamento, área de nombre área.

Expone:

Que el alumno D. José Miguel Ramírez Sanz, con DNI 71303106R, ha realizado el Trabajo final de Máster en Inteligencia de Negocio y Big Data en Entornos Seguros titulado título de TFM.

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 17 de abril de 2020

Vº. Bº. del Tutor:

Vº. Bº. del co-tutor:

Dr. José Francisco Díez Pastor

Dr. Álar Arnaiz Gonzalez

Resumen

En este primer apartado se hace una **breve** presentación del tema que se aborda en el proyecto.

Descriptores

Palabras separadas por comas que identifiquen el contenido del proyecto Ej: servidor web, buscador de vuelos, android ...

Abstract

A **brief** presentation of the topic addressed in the project.

Keywords

keywords separated by commas.

Índice general

Índice general	III
Índice de figuras	V
Índice de tablas	VI
 Memoria	 1
1. Introducción	3
2. Objetivos del proyecto	5
2.1. Objetivos funcionales	5
2.2. Objetivos técnicos	5
3. Conceptos teóricos	7
3.1. Secciones	7
3.2. Referencias	7
3.3. Imágenes	8
3.4. Listas de items	8
3.5. Tablas	9
4. Técnicas y herramientas	11
4.1. Visión por computador ~ Detección de movimiento	11
5. Aspectos relevantes del desarrollo del proyecto	15
5.1. Desarrollo FIS-HUBU	15
5.2. Investigación algoritmos de visión por computador	19

5.3. Investigación de <i>Detectron2</i>	20
6. Trabajos relacionados	21
7. Conclusiones y Líneas de trabajo futuras	23
 Apéndices	 24
Apéndice A Plan de Proyecto Software	27
A.1. Introducción	27
A.2. Planificación temporal	27
A.3. Estudio de viabilidad	29
 Apéndice B Especificación de Requisitos	 31
B.1. Introducción	31
B.2. Objetivos generales	31
B.3. Catalogo de requisitos	31
B.4. Especificación de requisitos	31
 Apéndice C Especificación de diseño	 33
C.1. Introducción	33
C.2. Diseño de datos	33
C.3. Diseño procedimental	33
C.4. Diseño arquitectónico	33
 Apéndice D Documentación técnica de programación	 35
D.1. Introducción	35
D.2. Estructura de directorios	35
D.3. Manual del programador	35
D.4. Compilación, instalación y ejecución del proyecto	35
D.5. Pruebas del sistema	35
 Apéndice E Documentación de usuario	 37
E.1. Introducción	37
E.2. Requisitos de usuarios	37
E.3. Instalación	37
E.4. Manual del usuario	37
 Bibliografía	 39

Índice de figuras

3.1. Autómata para una expresión vacía	8
4.2. Logo de Detectron2.	12
4.3. Logo de TensorFlow.	12
5.4. Menú principal de un responsable.	17
5.5. Menú de estadísticas.	17
5.6. Ejemplo de la evolución de un paciente vista por un responsable.	18
5.7. Menú principal de los pacientes.	18

Índice de tablas

3.1. Herramientas y tecnologías utilizadas en cada parte del proyecto	10
---	----

Memoria

Introducción

Descripción del contenido del trabajo y del estructura de la memoria y del resto de materiales entregados.

Objetivos del proyecto

En este apartado se va a comentar los objetivos funcionales, técnicos y personales que se tienen en el desarrollo de este proyecto.

2.1. Objetivos funcionales

En este subapartado se va a comentar los distintos objetivos que se querían cumplir con el desarrollo del proyecto:

- Desarrollar una aplicación web para poder realizar y evaluar la rehabilitaciones *online* de los pacientes con Parkinson.
- Desarrollar una aplicación web accesible y fácil de utilizar, sobre todo para los pacientes.
- Crear un herramienta capaz de comparar las rehabilitaciones hechas por los pacientes con los ejercicios bases.

2.2. Objetivos técnicos

En este subapartado se va a comentar los distintos objetivos relacionados con las técnicas y herramientas que se quieren aprender y utilizar:

- Crear una aplicación web para realizar rehabilitaciones *online*.
- Crear una herramienta de comparación de ejercicios a partir de algoritmos de visión por computador y detección de movimientos.

- Utilizar el lenguaje de programación *Python* para la comparación de ejercicios.
- Realizar el desarrollo del proyecto utilizando un repositorio *Git*, en concreto en *GitHub* para poder controlar las tareas y las versiones del proyecto.
- Utilizar la extensión de *Git* llamada *ZenHub* para controlar el estado de las tareas y la temporalidad de estas.
- Seguir el modelo *SCRUM* para desarrollar el proyecto de forma incremental.

2.3. Objetivos personales

Por último, se van a comentar los objetivos personales que se tienen en este proyecto, donde se encuentran objetivos desde probar conocimientos adquiridos en el máster como mejorar algunas cualidades personales.

- Poder ayudar a las personas mayores con Parkinson para que puedan realizar las rehabilitaciones necesarias sin necesidad de salir de sus casas. Además, mejorando esta rehabilitación a partir de la comparación de los ejercicios realizados.
- Mejorar mis capacidades comunicativas y de exposición en las diversas presentaciones del proyecto a los responsables.
- Usar los conocimientos adquiridos duran la carrera y duran el máster.
- Mejorar mis conocimientos sobre visión por computador.
- Conocer los distintos algoritmos de comparación y clasificación de instancias para los datos obtenidos sobre los ejercicios.

Conceptos teóricos

En aquellos proyectos que necesiten para su comprensión y desarrollo de unos conceptos teóricos de una determinada materia o de un determinado dominio de conocimiento, debe existir un apartado que sintetice dichos conceptos.

Algunos conceptos teóricos de \LaTeX ¹.

3.1. Secciones

Las secciones se incluyen con el comando `section`.

Subsecciones

Además de secciones tenemos subsecciones.

Subsubsecciones

Y subsecciones.

3.2. Referencias

Las referencias se incluyen en el texto usando `cite` [5]. Para citar webs, artículos o libros [3].

¹Créditos a los proyectos de Álvaro López Cantero: Configurador de Presupuestos y Roberto Izquierdo Amo: PLQuiz

3.3. Imágenes

Se pueden incluir imágenes con los comandos standard de \LaTeX , pero esta plantilla dispone de comandos propios como por ejemplo el siguiente:



Figura 3.1: Autómata para una expresión vacía

3.4. Listas de items

Existen tres posibilidades:

- primer item.
- segundo item.

1. primer item.
2. segundo item.

Primer item más información sobre el primer item.

Segundo item más información sobre el segundo item.

▪

3.5. Tablas

Igualmente se pueden usar los comandos específicos de \LaTeX o bien usar alguno de los comandos de la plantilla.

Herramientas	App	AngularJS	API REST	BD	Memoria
HTML5		X			
CSS3		X			
BOOTSTRAP		X			
JavaScript		X			
AngularJS		X			
Bower		X			
PHP			X		
Karma + Jasmine		X			
Slim framework			X		
Idiorm			X		
Composer			X		
JSON		X	X		
PhpStorm		X	X		
MySQL				X	
PhpMyAdmin				X	
Git + BitBucket		X	X	X	X
MikTeX					X
TeXMaker					X
Astah					X
Balsamiq Mockups		X			
VersionOne		X	X	X	X

Tabla 3.1: Herramientas y tecnologías utilizadas en cada parte del proyecto

Técnicas y herramientas

En este apartado se van a comentar las distintas técnicas y herramientas utilizadas en el desarrollo del proyecto.

4.1. Visión por computador ~ Detección de movimiento

La visión por computador, también llamada visión artificial, es la parte de la ciencia de la computación orientada a la recogida y al tratamiento de imágenes y vídeos [?].

Por la tipología de los datos con los que se trabaja en la visión por computador el procesamiento de éstos es muy costoso, este elevado coste computacional es causado en gran parte por la calidad de las imágenes y en el caso de los vídeos, además de la calidad de la imagen, por la cantidad de fotogramas por segundo con el que se ha grabado éste. ¿AÑADIR AQUI QUE POR ESO SE HA USADO GAMMA (TENDRÍA QUE EXPLICAR ANTES QUE ES GAMMA)?

Una de las principales funciones que tiene la visión artificial es la detección de movimiento, la cual consiste en primero detectar la posición de cada una de las partes del cuerpo (depende del propio algoritmo la división que se quiera realizar sobre el cuerpo) para posteriormente realizar un seguimiento de estos elementos.

Existen distintos algoritmos de detección de movimiento, cada uno diferenciado por el uso de distintos lenguajes de programación y diferentes métodos de inteligencia artificial con los que entrenar el modelo. Como en este

proyecto se ha querido trabajar en *Python* se han investigado los siguientes algoritmos que están implementados en este lenguaje de programación.

Detectron

Detectron es un proyecto de inteligencia artificial de *Facebook* centrado en la detección de objetos y personas con el uso de algoritmos de *deep learning* [2]. El proyecto se apoya en una de las librerías de inteligencia artificial más usadas en *Python*, *PyTorch*.

El proyecto cuenta ya con una segunda versión llamada Detectron2, figura 4.2, en la cual se han mejorado y añadido más modelos [6].



Figura 4.2: Logo de Detectron2.

PoseNet

PoseNet es un proyecto de *Google* orientado unicamente al seguimiento del movimiento de las personas centrando su análisis en los puntos claves del cuerpo humano. El proyecto se basa en la librería de inteligencia artificial de la propia compañía *Google* llamada *TensorFlow*, figura 4.3 [1, 4].



Figura 4.3: Logo de TensorFlow.

4.1. VISIÓN POR COMPUTADOR ~ DETECCIÓN DE MOVIMIENTO 13

TF-Pose-Estimator

Aspectos relevantes del desarrollo del proyecto

En este apartado se va a comentar, a manera de resumen temporal, el desarrollo del proyecto. Es en este apartado donde se comentarán las opciones y decisiones tomadas, los problemas surgidos y todos los aspectos importantes. Se ha considerado que la mejor forma de organizar este apartado es en seccionaes donde se comenta cada apartado del desarrollo.

5.1. Desarrollo FIS-HUBU

FIS-HUBU es una aplicación web que permite realizar vídeo llamadas entre responsables y pacientes con Parkinson para realizar rehabilitaciones de manera *online*, es decir, sin la necesidad de desplazarse hasta la consulta o el hospital. Esta aplicación, que ha sido desarrollada junto con mi compañero José Luis Garrido Labrador, permite por parte del responsable observar y evaluar la evolución del estado de un paciente, esta evolución también es visible para el paciente que puede ver su propio progreso.

La aplicación puede dividirse en distintas partes que serán comentadas a continuación.

Vídeo llamadas

El punto principal de la aplicación, y por ende su principal uso, son las vídeo llamadas entre pacientes y responsables o terapeutas que permitan sustituir la rehabilitaciones presenciales en consulta por rehabilitaciones *online*, permitiendo así que éstas se puedan dar más a menudo y puedan

ser accesibles para un mayor número de personas, sobre todo para aquellos pacientes que no se pueden desplazar.

Primero se realizó una investigación sobre las principales plataformas de vídeo llamadas. Lo que se estaba buscando de estas plataformas era:

- Creación de llamadas de manera sencilla y automática. Si es posible a partir de url.
- Vídeo llamada estable sin necesidad de una gran conexión.
- Plataforma que permita grabar la cámara de los pacientes.
- Plataforma gratuita.

Dentro de las plataformas que se investigaron están las más conocidas aplicaciones de este tipo como puede ser *Skype*, pero al final se decidió utilizar *Jitsi* ya que proporciona todas las necesidades anteriormente comentadas, y además permite en un futuro poder crear un servidor propio donde poder modificar parámetros como la calidad de las llamadas.

Responsable

Parte de la aplicación donde los responsables pueden realizar las siguientes tareas:

- Iniciar un vídeo llamada con un paciente.
- Evaluar la evolución de los pacientes.
- Modificar las evaluaciones de los pacientes.
- Comprobar la evolución de los pacientes.

La interfaz de la aplicación para tipo de usuario es sencilla y clara, como se puede ver en el menú principal, figura 5.4, o en el menú de estadísticas, figuras 5.5 y 5.6.



Figura 5.4: Menú principal de un responsable.



Figura 5.5: Menú de estadísticas.

Paciente

Los pacientes que van a utilizar la aplicación, son pacientes mayores con Parkinson. Esto se ha tenido muy en cuenta tanto en el diseño como en la creación de la parte de la aplicación orientada en los pacientes, se ha intentado que esta parte sea lo más accesible posible, para que todos los pacientes puedan manejarse bien con la aplicación y así poder sacarle el mayor provecho.

Para poder realizar una aplicación lo más accesible posible primero se ha de saber la forma que van a tener los pacientes de interactuar con ésta. En este caso los pacientes van a utilizar un mando de SNES² con botones de colores, es por ello que se ha aprovechado estos colores para poder mostrar en la interfaz de la aplicación que botón han de pulsar para realizar que acción. Además, se ha creado un botón de ayuda que carga una página donde se puede ver la acción que realiza cada botón del mando. Un ejemplo de la interfaz de esta aplicación se puede ver en la figura 5.7.

²SNES: *Super Nintendo Entertainment System*

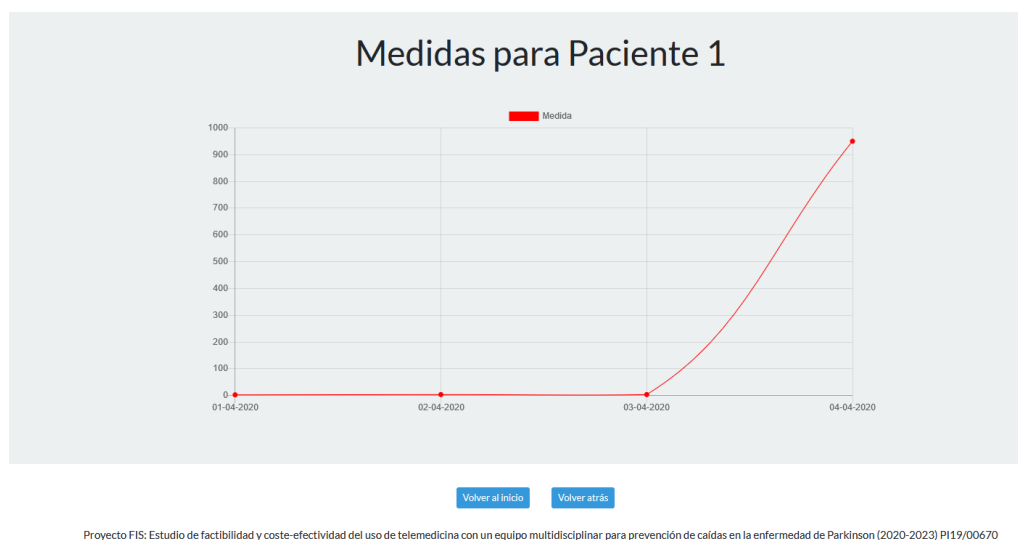


Figura 5.6: Ejemplo de la evolución de un paciente vista por un responsable.

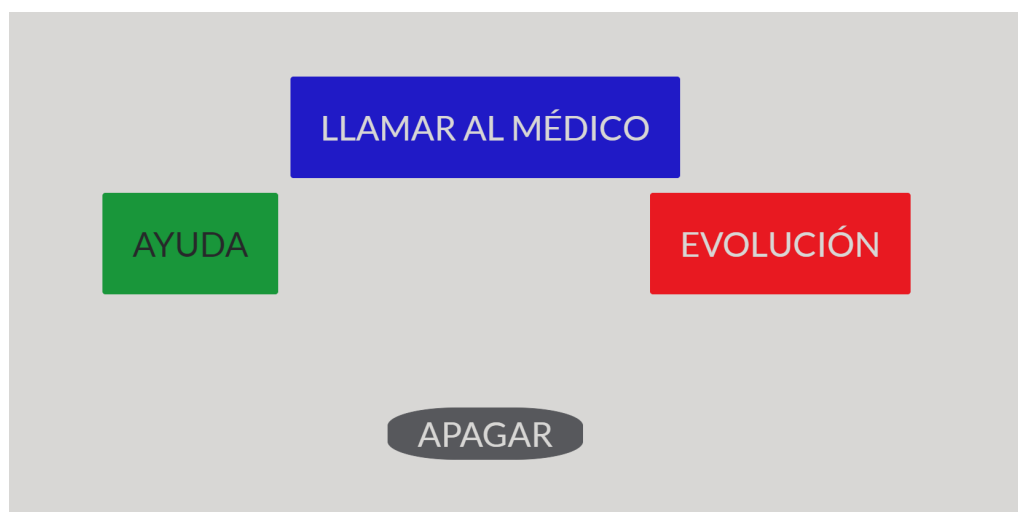


Figura 5.7: Menú principal de los pacientes.

Dispositivos necesarios

Para poder utilizar la aplicación se necesitan distintos dispositivos, que dependiendo del tipo de usuario (responsable o paciente) son unos u otros.

Para los responsables lo único que se necesita es un ordenador con conexión a *Internet* y una cámara conectada a éste.

Por otro lado, los pacientes se presupone que no disponen de ningún dispositivo capaz de cargar la aplicación, es por ello que a cada paciente se le proporciona:

- MSI
- CAMARA

Conexión

Como ya se ha comentado, el principal uso de la aplicación es las rehabilitaciones *online* para pacientes, mayoritariamente de tercera edad, que no se pueden desplazar a las consultas. Muchas de estas personas mayores viven en lugares donde no se tiene contratada ninguna línea de *Internet*, es por ello que además del desarrollo de la aplicación se contrató una serie de *routers* 4G para poder proporcionar conexión a los dispositivos necesarios.

5.2. Investigación algoritmos de visión por computador

Tras haber desarrollado la versión inicial, donde en un futuro se quiere añadir los resultados de este proyecto, se pasó a realizar la primera investigación sobre los distintos algoritmos de visión por computador capaces de detectar y seguir los movimientos de una persona.

De estos algoritmos se necesita:

- Posibilidad de cargar un modelo existente o crear un modelo capaz de detectar a la persona que sale en la imagen.
- Posibilidad de cargar un modelo existente o crear un modelo capaz de detectar la posición de la persona.
- Que el procesamiento de nuevos fotogramas para detectar a la persona y su posición se realice en poco tiempo.
- Que la salida del procesamiento de los fotogramas pueda servir para una posterior comparación con el ejercicio base.

Teniendo todos estos factores en cuenta se estudió qué herramientas se pueden usar para esta fase del trabajo. Se buscó herramientas programadas

en *Python* para poder conectarse bien con el resto del proyecto y porque es uno de los lenguajes con los cuales se tiene más soltura, tanto por parte del alumno como por parte de los tutores para resolver dudas y ayudar en los problemas. Las herramientas encontradas fuera:

- *TF-Pose-Estimator*.
- *PoseNet*.
- *Detectron2*.

De cada una de estas herramientas se realizó una investigación y experimentación para probar si cumplían las necesidades requeridas. Al finalizar esta etapa, la única herramienta que permitía realizar todas las necesidades era *Detectron2*. Además, permite con un modelo ya creado por los propios desarrolladores realizar las tareas de predicción de elementos y de la posición de la persona.

5.3. Investigación de *Detectron2*

PROBLEMAS CON EL GUARDADO DE VÍDEO Y EXTRACCIÓN
DE LAS CARACTERÍSTICAS PONER EL UMBRAL, PONER COMO
MEJORA CAMBIANDO EL UMBRAL

Trabajos relacionados

Este apartado sería parecido a un estado del arte de una tesis o tesina. En un trabajo final de máster no parece tan obligada su presencia, aunque se puede dejar a juicio del tutor el incluir un pequeño resumen comentado de los trabajos y proyectos ya realizados en el campo del proyecto en curso.

Conclusiones y Líneas de trabajo futuras

Todo proyecto debe incluir las conclusiones que se derivan de su desarrollo. Éstas pueden ser de diferente índole, dependiendo de la tipología del proyecto, pero normalmente van a estar presentes un conjunto de conclusiones relacionadas con los resultados del proyecto y un conjunto de conclusiones técnicas. Además, resulta muy útil realizar un informe crítico indicando cómo se puede mejorar el proyecto, o cómo se puede continuar trabajando en la línea del proyecto realizado.

Apéndice

Apéndice A

Plan de Proyecto Software

A.1. Introducción

La planificación de un proyecto es una fase esencial en desarrollo de éste, ya que permite comprobar y guiar el proyecto según las necesidades actuales y futuras.

Dentro de este plan para el desarrollo del proyecto se pueden diferenciar dos puntos sobre los que se puede enfocar este estudio:

- **Temporal:** enfoque en el que se analiza la evolución del proyecto en el tiempo. Al utilizar una metodología SCRUM, se ha dividido el desarrollo en *sprints* de 2 semanas.
- **Viabilidad:** comprobación de la adecuación del proyecto tanto desde el ámbito económico como desde el ámbito legal.

A.2. Planificación temporal

Como ya se ha comentado, la planificación temporal se ha dividido en distintos *sprints* de 2 semanas cada uno, en lo cuales se realizaron distintas tareas y reuniones. Cabe destacar que antes de realizar el primer *sprint* se desarrolló la aplicación FIS sobre la cuál será implementado en el futuro el proyecto desarrollado.

Sprint0

En este *sprint* se creó la aplicación para el Hospital Universitario de Burgos llamada FIS-HUBU. Esta aplicación, como se ha visto en el apartado 5.1, sirve para realizar vídeo llamadas entre médicos y los pacientes con Parkinson en las cuales se realizan rehabilitaciones para mejorar su calidad de vida. Como se ha explicado, esta aplicación cuenta con dos partes, una para los médicos o responsables y otra para los pacientes. Las tareas realizadas en este *sprint* son las siguientes:

- Investigación sobre plataformas de vídeo llamadas.
- Creación de la estructura base de la página web.
- Diseño e implementación de la base de datos.
- Creación del login por usuario, guardando una *cookie* para mantener la sesión.
- *Backend* de las páginas de los pacientes, donde se incluye la creación de las llamadas, el calculo de la evolución a partir de los datos en la base de datos, la grabación de la cámara del paciente y subida a un servidor propio de la universidad...
- *Backend* de las páginas de los responsables.

Tras desarrollar la aplicación se empezaron con las tareas de explotación de la aplicación, en donde se cogieron los dispositivos para los pacientes (AQUI ESCRIBIR EL MODELO DEL BICHO0 Y LA CAMARA) y se les instalaron todo el *software* necesario para funcionar correctamente, y con ello poder grabar los vídeos y subirlos a un servidor propio. Debido a la aparición del COVID-19 y el posterior confinamiento sufrido y las consiguientes medidas de seguridad no se pudo empezar a instalar los dispositivos en las casa de los pacientes.

Sprint1: 17/02/2020 - 26/02/2020

Este es el primer *sprint* real del desarrollo de la parte del proyecto de visión artificial, en el se realizaron las primeras tareas de creación del repositorio en *GitHub* y la creación de la estructura del mismo. Además, se creo el documento *LaTeX* a partir de la plantilla de la asignatura. Las tareas más relevantes en este primer *sprint* fueron las relacionadas con la configuración de *Gamma* y los primeros pasos en la investigación de

algoritmos de visión por computador y detección de movimiento, como son *Detectron2* y *PoseNet*.

Sprint2: 27/02/2020 - 11/03/2020

En este *sprint* se realizó una tarea fundamental dentro del desarrollo del proyecto, que es la investigación y elección del algoritmo de visión artificial para la obtención de la postura, que se usará en los fotogramas de los vídeos recogidos con la aplicación FIS-HUBU y compararlos con otros vídeos base para saber la exactitud del ejercicio. Esta tarea de investigación de los distintos algoritmos dio como resultado que el *Detectron2* (apartado 4.1) es el algoritmo que más se amoldaba al problema del proyecto.

Además, en este *sprint* se realizaron las tareas de documentación necesarias sobre los algoritmos candidatos y sobre la selección del mejor.

Sprint3: 12/03/2020 - 17/04/2020

En este tercer *sprint* se profundizó en la investigación de los distintos algoritmos y modelos de *Detectron2*. En esta investigación se encontró un modelo ya entrenado capaz de obtener la posición de la persona fotograma a fotograma de una forma extraordinariamente precisa.

Una vez se encontró el modelo se pasó a interpretar la salida del modelo, para poder ajustarlo (principalmente para que solo detecte la persona que mejor identifica, normalmente el centro de la pantalla) y obtener datos para su posterior tratamiento y comparación.

Por último, en este *sprint* se realizaron diversas tareas de documentación.

A.3. Estudio de viabilidad

Viabilidad económica

Viabilidad legal

Apéndice B

Especificación de Requisitos

- B.1. Introducción
- B.2. Objetivos generales
- B.3. Catalogo de requisitos
- B.4. Especificación de requisitos

Apéndice C

Especificación de diseño

- C.1. Introducción
- C.2. Diseño de datos
- C.3. Diseño procedimental
- C.4. Diseño arquitectónico

Apéndice D

Documentación técnica de programación

- D.1. Introducción
- D.2. Estructura de directorios
- D.3. Manual del programador
- D.4. Compilación, instalación y ejecución del proyecto
- D.5. Pruebas del sistema

Apéndice E

Documentación de usuario

- E.1. Introducción
- E.2. Requisitos de usuarios
- E.3. Instalación
- E.4. Manual del usuario

Bibliografía

- [1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.
- [2] Ross Girshick, Ilija Radosavovic, Georgia Gkioxari, Piotr Dollár, and Kaiming He. Detectron. <https://github.com/facebookresearch/detectron>, 2018.
- [3] John R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, 1992.
- [4] TensorFlow. Pose estimation. https://www.tensorflow.org/lite/models/pose_estimation/overview, feb 2020.
- [5] Wikipedia. Latex — wikipedia, la enciclopedia libre, 2015. [Internet; descargado 30-septiembre-2015].
- [6] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. <https://github.com/facebookresearch/detectron2>, 2019.