



**EDUCACIÓN**

SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO NACIONAL DE MÉXICO

Instituto Tecnológico de Orizaba

“2020, Año de Leona Vicario, Benemérita Madre de la Patria”

DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

OPCIÓN I.- TESIS

**TRABAJO PROFESIONAL**

**“Sistema de Tele-cuidado de Adultos Mayores basado  
en Analítica Big Data para el Internet de las Cosas”**

QUE PARA OBTENER EL GRADO DE:  
**MAESTRO EN SISTEMAS  
COMPUTACIONALES**

PRESENTA:

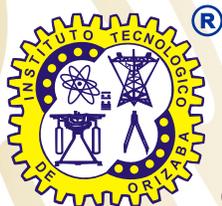
*I.C.I. Jesús Miguel Echevarría Díaz*

DIRECTOR DE TESIS:

*Dr. José Luis Sánchez Cervantes*

CODIRECTOR DE TESIS:

*Dr. Luis Omar Colombo Mendoza*



ORIZABA, VERACRUZ, MÉXICO.

OCTUBRE 2020



"2020, Año de Leona Vicario, Benemérita Madre de la Patria"

Orizaba, Veracruz, 20/10/2020  
Dependencia: **División de Estudios de  
Posgrado e Investigación**  
Asunto: **Autorización de Impresión**  
OPCION: I

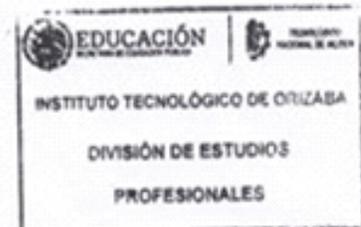
**C. JESÚS MIGUEL ECHEVARRÍA DÍAZ**  
Candidato a Grado de Maestro en:  
**SISTEMAS COMPUTACIONALES**  
**P R E S E N T E . -**

De acuerdo con el Reglamento de Titulación vigente de los Centros de Enseñanza Técnica Superior, dependiente de la Dirección General de Institutos Tecnológicos de la Secretaría de Educación Pública y habiendo cumplido con todas las indicaciones que la Comisión Revisora le hizo respecto a su Trabajo Profesional titulado:

**" SISTEMA DE TELE-CUIDADO DE ADULTOS MAYORES BASADO EN  
ANALÍTICA BIG DATA PARA EL INTERNET DE LAS COSAS"**

Comunico a Usted que este Departamento concede su autorización para que proceda a la impresión del mismo.

**A T E N T A M E N T E**  
Excelencia en Educación Tecnológica®  
CIENCIA - TÉCNICA - CULTURAS



*Delma Vargas López*  
**LI. DELMA VARGAS LÓPEZ**  
**JEFA DE LA DIVISIÓN DE ESTUDIOS  
PROFESIONALES**





"2020, Año de Leona Vicario, Benemérita Madre de la Patria"

FECHA: 22/07/2020

ASUNTO: Revisión del Trabajo Escrito

**C. MARIO LEONCIO ARRIJOJA RODRIGUEZ**

JEFE DE LA DIVISION DE ESTUDIOS  
DE POSGRADO E INVESTIGACION  
P R E S E N T E.

Los que suscriben miembros del jurado, han realizado la revisión de la Tesis del (la) C.:

**JESÚS MIGUEL ECHEVARRÍA DÍAZ**  
No. DE CONTROL: **M18011502**

La cual lleva el título de:

**"Sistema de Tele-cuidado de Adultos Mayores basado en  
Analítica Big Data para el Internet de las Cosas"**

Y concluyen que se acepta.

A T E N T A M E N T E

PRESIDENTE: DR. JOSÉ LUIS SÁNCHEZ CERVANTES

SECRETARIO: DR. GINER ALOR HERNÁNDEZ

VOCAL: M.R.T. IGNACIO LÓPEZ MARTÍNEZ

VOCAL SUP.: DR. LUIS OMAR COLOMBO MENDOZA

EGRSADO (A) DE LA MAESTRIA EN **SISTEMAS COMPUTACIONALES**

OPCION: **1 Tesis**



*A Servando y Sara, que me empujaron a ser lo que hoy soy*

*A Ore, que siempre ha estado ahí para mí*

*A mis abuelos, todos*

*A Giselle, a quien amo*

## Agradecimientos

Mi eterno y especial agradecimiento a Juana Eufrasia Rodríguez, mi abuela loca, su amor incondicional ha sido siempre mi mayor fuerza, y a Humberto López, aunque ya no estés conmigo abuelo.

Le agradezco infinitamente a mi padre y madre, que se han quitado de sí mismos para hacerme fuerte, para darme vida y visión, para guiarme; ellos saben que ha sido un camino muy difícil, y gracias a ellos he logrado recorrerlo. Gracias a mi hermano, por ser mi mayor reto en la vida y mi mayor apoyo a la vez, hoy nos hacemos máster los dos.

A Giselle, quien, a pesar de la distancia, ha resistido a mi lado durante dos largos años, ya voy a casa mi amor.

A mi abuelo Orestes y mi tía Coco, gracias por todo su apoyo y cariño, los extraño siempre. A Daríel, que siempre me hace reír a pesar de todo. A mi abuela Nana, nunca te olvido. Todos tienen un lugar especial en mi corazón.

Quiero agradecer especialmente a la maestra Celia, mi primer contacto con México, mi coordinadora y tutora, quien me ayudó incondicionalmente de principio a fin, le estoy eternamente agradecido. Agradezco a la genial maestra Beatriz, quien resistió galantemente mis constantes embates de preguntas; a los maestros Luis Ángel y Nacho, excelentes profesores y personas, quienes me ayudaron tanto en el aspecto docente como personal, cuando más lo necesitaba. Mil gracias al doctor José Luis, quien fue más allá de su rol de asesor y me hizo sentir parte de la familia. A todos mis profesores del instituto, es un privilegio ser su alumno, gracias por sus enseñanzas. A mi jurado de tesis, quienes me apoyaron durante todo el trayecto, y me obligaron a mejorar hasta alcanzar un resultado de excelencia. A mis compañeros del instituto, nunca olvidaré los momentos que compartimos, y a todas las personas que me ayudaron, de lejos y de cerca, durante este difícil trayecto.

Finalmente, quiero agradecer al Consejo Nacional de Ciencia y Tecnología, cuyo apoyo fue clave para la exitosa culminación de este proyecto.

## Contenido

Resumen .....	xii
Abstract .....	xiii
Introducción.....	14
Capítulo 1 Antecedentes.....	16
1.1 Marco teórico.....	16
1.1.1 Big Data.....	16
1.1.2 IoT .....	16
1.1.3 Analítica Big Data .....	16
1.1.4 Modelado Big Data .....	17
1.1.5 Tele-cuidado.....	17
1.1.6 Signo vital .....	17
1.1.7 Aprendizaje automático .....	17
1.1.8 Visión por computadora (VPC) .....	18
1.1.9 <i>Wearable</i> .....	18
1.1.10 Servicios <i>web</i> .....	18
1.1.11 Servicios <i>web</i> RESTful .....	19
1.1.12 Base de datos relacional .....	19
1.1.13 Base de datos no relacional .....	19
1.1.14 Apache Cassandra .....	20
1.1.15 Apache Spark.....	20
1.1.16 OpenCV.....	20
1.1.17 Java.....	21
1.1.18 Scala .....	21

1.1.19	Python .....	21
1.1.20	Angular .....	21
1.1.21	Node.js .....	22
1.1.22	Raspberry PI 3 B+ (RPi3B+) .....	22
1.1.23	Fitbit Inc.....	22
1.2	Situación tecnológica, económica y operativa de la empresa .....	23
1.3	Planteamiento del problema.....	23
1.4	Objetivos general y específicos.....	24
1.4.1	Objetivo General.....	24
1.4.2	Objetivos Específicos .....	24
1.5	Justificación.....	25
Capítulo 2	Estado de la práctica .....	26
2.1	Trabajos relacionados .....	26
2.2	Análisis comparativo .....	34
2.3	Propuesta de solución.....	54
2.3.1	Justificación de la solución seleccionada .....	56
Capítulo 3	Aplicación de la Metodología .....	59
3.1	Sprint 0.....	59
3.1.1	Arquitectura de la solución.....	60
3.1.2	Flujo de trabajo .....	63
3.1.3	Especificación de requisitos de la aplicación .....	65
3.1.4	Modelos de bases de datos .....	66
3.2	Sprint 1 .....	71
3.2.1	Registro de la aplicación en el dominio <a href="https://dev.fitbit.com">https://dev.fitbit.com</a> .....	71

3.3	Sprint 2.....	72
3.3.1	Autenticación de usuarios.....	72
3.3.2	Gestión de usuarios.....	73
3.3.3	Gestión de cuentas de adultos mayores.....	73
3.4	Sprint 3.....	76
3.4.1	Módulo de recopilación de datos provenientes del DVP.....	76
3.5	Sprint 4.....	81
3.5.1	Analítica en el servidor. Árbol de decisión.....	81
3.5.2	Sistema de notificaciones.....	83
3.6	Sprint 5.....	84
3.6.1	Módulo de VPC.....	84
3.7	Sprint 6.....	94
3.7.1	Módulo de aprendizaje automático con Apache Spark.....	94
3.8	Sprint 7.....	99
3.8.1	Panel de incidentes.....	99
Capítulo 4	Resultados.....	101
4.1	Caso de estudio.....	101
4.1.1	Detección de accidentes y parámetros de salud irregulares.....	101
4.2	Evaluación y resultados.....	104
4.2.1	Haar Cascade.....	106
4.2.2	Histograma de Gradientes Orientados (HOG).....	106
4.2.3	Red Neuronal Profunda (DNN).....	107
Capítulo 5	Conclusiones y recomendaciones.....	109
5.1	Conclusiones.....	109

5.2 Recomendaciones.....	109
Anexos y apéndices .....	110
Bibliografía .....	112

## ÍNDICE DE FIGURAS

Figura 1. Diseño conceptual de la solución .....	60
Figura 2. Arquitectura del sistema .....	61
Figura 3. Diagrama de paquetes del sistema .....	65
Figura 4. Diagrama entidad-relación .....	66
Figura 5. Modelo de base de datos objeto-relacional .....	67
Figura 6. Diagrama lógico .....	68
Figura 7. Diagrama físico .....	69
Figura 8. Autenticación – diagrama de secuencia .....	72
Figura 9. Vinculación de cuenta de adulto mayor – diagrama de secuencia.....	73
Figura 10. Renovar token de acceso.....	74
Figura 11. Interfaz del dominio <a href="https://www.fitbit.com">https://www.fitbit.com</a> para autenticación por OAuth	74
Figura 12. Interfaz de asignación de permisos .....	75
Figura 13. Solicitud de token de acceso – diagrama de secuencia.....	75
Figura 14. Pre-procesamiento de valores de ritmo cardíaco .....	83
Figura 15. Buzón de notificaciones .....	84
Figura 16. Instancias de pruebas con el algoritmo Haar Cascade .....	86
Figura 17. Instancias de pruebas con el algoritmo HOG .....	88
Figura 18. Persona detectada por modelo pre-entrenado.....	90
Figura 19. Detección de caída.....	91
Figura 20. Detección de marcador rectangular verde. Espacio HSV (superior izquierda), máscara binaria (superior derecha), intersección bit-a-bit (inferior izquierda), y resultado resaltado con contorno (inferior derecha).....	92
Figura 21. Panel de incidentes .....	100
Figura 22. Datos registrados durante el día del incidente .....	102
Figura 23. Datos del incidente, y parámetros de salud registrados previos al mismo .....	103
Figura 24. Accidente detectado por el módulo de VPC .....	104

Figura 25. Echevarría-Díaz, J. M., Sánchez-Cervantes, J. L., Colombo-Mendoza, L. O., Alor-Hernández, G. & Martínez-López I. (2020). An Architecture for an IoT-based Telecare System for the Elderly Using Big Data Analytics. Research in Computing Science, 149, 53-59. .... 110

Figura 26. Carta de uso de sistema – Empresa Rodas..... 111

## ÍNDICE DE LISTADOS

Listado 1. Sentencias CQL utilizadas en la creación del esquema de base de datos para Cassandra.....	70
Listado 2. Instancia del módulo <code>fitbit-node</code> utilizando los metadatos de la aplicación .....	76
Listado 3. Obtener perfil de usuario .....	77
Listado 4. Obtener datos de ritmo cardíaco.....	77
Listado 5. Obtener cantidad de pasos.....	78
Listado 6. Obtener cantidad de pisos .....	79
Listado 7. Obtener estatus de sueño.....	79
Listado 8. Resumen de zonas cardíacas y calorías quemadas.....	80
Listado 9. Instancia del controlador dedicado para la ejecución de sentencias en Cassandra.....	80
Listado 10. Ejemplo de persistencia de datos en Cassandra .....	81
Listado 11. Ejemplo de persistencia de datos en PostgreSQL.....	81
Listado 12. Se inicializa el flujo de video y se comienzan a procesar las imágenes en un ciclo <code>while</code> .....	85
Listado 13. Detección de personas en video usando Haar Cascade .....	86
Listado 14. Detección de personas en video usando descriptor HOG .....	88
Listado 15. Carga del modelo pre-entrenado, filtro de detecciones y determinación de coordenadas de las personas en cada imagen .....	90
Listado 16. Captura de imagen del incidente o accidente .....	93
Listado 17. Envío de notificación desde el módulo de VPC .....	94
Listado 18. Entrenamiento del modelo SVM .....	96
Listado 19. Entrenamiento del modelo Regresión Logística.....	97
Listado 20. Pre-procesamiento de datos, clasificación y persistencia de incidente con Spark.....	99

## ÍNDICE DE TABLAS

TABLA I Análisis comparativo de los artículos estudiados .....	53
TABLA II Alternativas de solución .....	55
TABLA III Matriz de confusión generada a partir de las clasificaciones mediante Haar Cascade .....	106
TABLA IV Matriz de confusión generada a partir de las clasificaciones mediante HOG .....	106
TABLA V Matriz de confusión generada a partir de las clasificaciones mediante DNN .....	107

## Resumen

El envejecimiento de la población es un problema real y constante a nivel mundial. La población más longeva representa el sector demográfico más afectado en cuestiones de salud y seguridad en el hogar. Con el advenimiento de Big Data de manera paralela a la omnipresencia del Internet de las Cosas, las alternativas de solución basadas en sistemas de tele-cuidado, son cada vez más deseadas en el tratamiento y cuidado remoto de las personas más longevas o frágiles, permitiéndoles permanecer en sus hogares y continuar con su modo de vida de manera relativamente independiente. Esta situación ha creado una demanda continua de soluciones de tele-cuidado novedosas e innovadoras.

Con el objetivo de presentar una alternativa de solución a la problemática en cuestión, el presente proyecto de investigación propone diseñar y desarrollar un sistema de tele-cuidado basado en el Internet de las Cosas para monitorizar, de manera remota, signos vitales y actividades de adultos mayores, generar alertas y recomendaciones basadas en Analítica Big Data, y detectar y prevenir accidentes.

## **Abstract**

The aging of world population is a real and constant issue. Longest-living individuals make for the most affected demographic in matters of health and safety at home. Along with the advent of Big Data and the omnipresence of the Internet of Things, telecare system-based alternatives are increasingly more accepted and looked for in the caring of our elder and more fragile family members, allowing them to stay in their homes and continue their normal day-to-day life. This situation has spawned a continuous demand for new and innovative telecare-based solutions.

With the aim of providing a solution to the problem at hand, this research project proposes to design and develop a telecare system based on the Internet of Things, to monitor, from a remote location, vital signs and activities of the elderly, to generate alerts based on Big Data Analytics, and detect and prevent accidents.

## Introducción

Los fenómenos como Big Data y el Internet de las Cosas son relativamente nuevos en el escenario de las tecnologías de comunicación a nivel mundial, pero ambos crecen de manera acelerada. El Internet de las Cosas representa una ubicua plataforma de evolución para disímiles campos de investigación y desarrollo tales como el transporte y la agricultura, por mencionar algunos. El Internet de las Cosas se aplica actualmente en varias esferas de la sociedad, dando lugar al surgimiento de sistemas, hogares y ciudades inteligentes. Dichos sistemas generan volúmenes inmensos de datos a un ritmo acelerado, presentados tanto en formato de texto e imágenes, como de sonido y video; este inmenso y anónimo conjunto de datos se conoce como Big Data. Esto representa tanto una oportunidad como un reto sin precedentes, siendo indispensable un mecanismo de recolección y procesamiento de datos que vaya más allá de las capacidades de los sistemas de *software* que tradicionalmente se usan con este fin. La Analítica Big Data y las herramientas basadas en ella se presentan como la solución natural a este problema, y se diseñan específicamente para lidiar con grandes cantidades de datos heterogéneos, estructurados y no estructurados, generados de manera constante y a un ritmo acelerado.

El campo de la salud y la medicina no queda fuera del alcance de las soluciones basadas en el uso de sensores y dispositivos, que forman parte del Internet de las Cosas, especialmente a través del uso de sistemas de tele-cuidado. La popularidad de los sistemas de tele-cuidado aumenta con el envejecimiento sostenido de la población mundial y la prevalencia de las enfermedades crónicas, y radica en la permanente necesidad de permitir a las personas de la tercera edad y a personas con condiciones crónicas mantener su independencia, y continuar viviendo en sus hogares.

El presente proyecto de tesis propone el diseño y desarrollo de un sistema de tele-cuidado basado en el Internet de las Cosas para monitorizar, de manera remota, signos vitales y actividades de adultos mayores, generar alertas basadas en Analítica Big Data para la prevención y detección de accidentes. Para esto se analizó el estado

del arte de la investigación en tele-cuidado y Analítica Big Data para el cuidado de la salud, así como dispositivos domóticos para videovigilancia de actividades en adultos mayores. Asimismo, se diseñó la arquitectura tecnológica de la plataforma que permite la integración de, al menos un dispositivo vestible o ponible (DVP), un dispositivo domótico de videovigilancia, módulos para el procesamiento de datos y Analítica Big Data, y la aplicación web. Se desarrolló un módulo de recopilación de datos procedentes de dispositivos vestibles o ponibles, y dispositivos domóticos de videovigilancia para monitorizar signos vitales y actividades de adultos mayores, y un módulo para la detección de anomalías en signos vitales y actividades de adultos mayores, para la generación de alertas basadas en Analítica Big Data. Finalmente, se desarrolló la aplicación web, y se validó el sistema implementado mediante el diseño y la conducción de, al menos, un par de casos de estudio de prevención y manejo de incidentes y accidentes en el cuidado de la salud de adultos mayores.

Para ofrecer una mejor visión de esta investigación, el presente documento está constituido por cinco capítulos. El primero de ellos tiene como objetivo proveer los conceptos básicos del dominio del problema (marco teórico), así como el planteamiento del problema, objetivo general, objetivos específicos y la justificación del proyecto; en el segundo capítulo se incluye un estudio del estado del arte a través del análisis de quince artículos científicos relacionados con el tema de investigación, y enfocadas en el desarrollo de sistemas basados en Analítica Big Data y el Internet de las Cosas, asimismo se establece la propuesta de solución. En el capítulo tres se muestran el diseño conceptual de solución y la arquitectura del sistema, así como la descripción de sus capas y componentes, y la interacción entre ellos. Asimismo, dicho capítulo incluye una descripción detallada de las actividades realizadas, distribuidas en *sprints*, de acuerdo a la metodología seleccionada. Los resultados obtenidos y casos de estudio realizados conforman el capítulo cuatro de la tesis. Finalmente, en el capítulo cinco, se resumen las conclusiones alcanzadas durante la realización del trabajo, así como las recomendaciones para trabajos futuros.

# Capítulo 1      Antecedentes

## 1.1 Marco teórico

En esta sección, se presentan los conceptos y tecnologías analizados en relación con el tema de investigación que abarca el presente proyecto.

### 1.1.1 Big Data

En la literatura estudiada no se identificó un consenso en cuanto a la definición formal del término **Big Data**, sin embargo, todos los intentos de definirlo comparten pautas en común, tales como el gran volumen, velocidad y heterogeneidad que caracterizan a la información que lo conforman. Durante el desarrollo del presente trabajo se adopta la siguiente definición del término: conjunto de datos (estructurados, no estructurados y semi-estructurados) cuyo tamaño, velocidad, tipo y complejidad exceden la habilidad del software convencional para ser recolectados, procesados y analizados en un lapso de tiempo razonable [1].

### 1.1.2 IoT

IoT (*Internet of Things*, Internet de las Cosas) provee una plataforma de comunicación para sensores y dispositivos dentro de un entorno inteligente y habilita el intercambio de información entre plataformas de manera conveniente [2]. IoT se adopta en el desarrollo de sistemas inteligentes, tales como oficinas inteligentes, agricultura inteligente y transporte inteligente, por mencionar algunos. [3].

### 1.1.3 Analítica Big Data

BDA (*Big Data Analytics*, Analítica Big Data) es el proceso de examinación de grandes conjuntos de datos – o Big Data – que incluyen una gran variedad de tipos de datos [4] para revelar patrones ocultos, correlaciones desconocidas, tendencias del mercado, e información novedosa en general [5] que ayuda a las organizaciones a tomar decisiones informadas. Es una forma de analítica avanzada que involucra aplicaciones complejas, con elementos como modelos predictivos y algoritmos estadísticos, ejecutados sobre sistemas de análisis de alto rendimiento.

#### **1.1.4 Modelado Big Data**

Los principios establecidos para el modelado en bases de datos relacionales como lo es el modelado enfocado a datos, el establecimiento de relaciones bien definidas, y los procesos de normalización para lograr consistencia y evitar redundancia en los datos, no necesariamente aplican al modelado Big Data, ya que en este se prioriza la velocidad de acceso y escritura sobre los datos, aunque esto implique la duplicación de los mismos, para soportar las diferentes consultas al sistema. Dichas consultas y el modelo conceptual de la aplicación se definen desde un principio, y guiarán el proceso de modelado en todas sus fases: conceptual, lógico y físico.

#### **1.1.5 Tele-cuidado**

Tele-cuidado es el uso de las tecnologías de la comunicación para proveer salud y cuidado social directamente al usuario o paciente [6]. Los términos tele-cuidado y tele-salud conforman, de manera general, la tele-medicina. En concreto, el término tele-cuidado hace referencia al uso de las tecnologías de las comunicaciones para proveer cuidados de salud y sociales de manera directa a los usuarios o pacientes.

#### **1.1.6 Signo vital**

Los signos vitales se utilizan para medir las funciones básicas del cuerpo. Estas medidas ayudan en la valoración de la salud física general de una persona, proveer pistas a posibles enfermedades y mostrar progreso hacia la recuperación [7]. Los rangos normales para los signos vitales de una persona varían con la edad, el peso, el género y la salud en general. Existen cuatro signos vitales principales: temperatura corporal, presión sanguínea, ritmo cardíaco y ritmo respiratorio.

#### **1.1.7 Aprendizaje automático**

Aprendizaje automático (ML, por sus siglas en inglés) es un método de análisis de datos que automatiza la construcción de un modelo analítico. Es una rama de la inteligencia artificial basada en la idea de que los sistemas son capaces de “aprender” a partir de la información, identificar patrones y “tomar” decisiones con mínima intervención humana [8]. De forma general, el aprendizaje automático se sub-divide en tres tipos: aprendizaje supervisado (una vez se entrena el modelo, este es capaz de

“tomar” una decisión o realizar una predicción cuando recibe nuevos datos), no supervisado (el modelo “aprende” a través de la “observación” y encuentra “estructuras” en los datos) y de reforzamiento (es la habilidad de un agente de interactuar con su entorno y “descubre” cuál es el mejor resultado) [9].

### **1.1.8 Visión por computadora (VPC)**

VPC es la rama de las ciencias computacionales que se enfoca en permitir a las computadoras ver, identificar y procesar imágenes de una manera similar a la visión humana, permitiendo el reconocimiento de objetos en dichas imágenes [10]. Las tareas de VPC incluyen métodos para la adquisición, procesamiento, análisis y comprensión de imágenes digitales y, en general, involucran la extracción de datos multi-dimensionales del mundo real, con el objetivo de producir información numérica o simbólica en la forma de decisiones [11].

### **1.1.9 *Wearable***

El término inglés *wearable* – ponible o vestible – se refiere a una tecnología o dispositivo de cómputo que se incorpora en artículos de vestir o accesorios usados sobre el cuerpo o implantados en el mismo. Estos dispositivos son capaces de llevar a cabo varias de las mismas tareas realizadas por celulares o portátiles, por mencionar algunos. Mediante el uso de *wearables* se obtiene información biológica del usuario, así como monitorizar sus funciones fisiológicas y signos vitales [12].

### **1.1.10 Servicios *web***

Un servicio *web* es un término genérico para una función de *software*, máquina-a-máquina e interoperable, alojada en una ubicación direccionable en la red. Un servicio *web* posee una interfaz, la cual oculta los detalles de implementación, que permite su uso de manera independiente al *hardware* o la plataforma de *software* en la que es implementado, así como del lenguaje de programación en que se codifique. Esta independencia insta a las aplicaciones basadas en servicios *web* a tener bajo acoplamiento, ser orientadas a componentes y, ser implementaciones multi-tecnologías. Los servicios *web* se usan en solitario o en conjunto con otros servicios *web* para realizar una agregación compleja o una transacción de negocios [13].

### 1.1.11 Servicios *web* RESTful

Los servicios RESTful HTTP y SOAP (*Simple Object Access Protocol*, Protocolo Simple de Acceso a Objetos) / WSDL (*Web Services Description Language*, Lenguaje de Descripción de Servicios *Web*) son dos estilos arquitectónicos para la construcción de servicios *web*, donde los servicios basados en SOAP/WSDL siguen un enfoque centrado en operaciones y los servicios basados en RESTful siguen un enfoque centrado en recursos [14]. Un servicio *web* RESTful se adhiere a las reglas establecidas por el estilo arquitectónico REST<sup>1</sup>, el cual especifica un conjunto de restricciones que, aplicadas a un servicio *web*, induce propiedades deseables como rendimiento, escalabilidad y modificabilidad, que permiten a los servicios trabajar mejor en la *web*.

### 1.1.12 Base de datos relacional

Una base de datos relacional se basa en el modelo relacional<sup>2</sup> y utiliza una colección de tablas para representar, ambos, datos y las relaciones entre esos datos. También incluye un DML (*Data Manipulation Language*, Lenguaje de Manipulación de Datos) y un DDL (*Data Definition Language*, Lenguaje de Definición de Datos) [15]. A cada tabla de la colección se le asigna un nombre único [16].

### 1.1.13 Base de datos no relacional

Una base de datos no relacional es aquella que no usa el esquema tabular de filas y columnas encontrado en la mayoría de los sistemas de bases de datos tradicionales. En su lugar, estas bases de datos usan un modelo de almacenamiento optimizado para los requerimientos específicos del tipo de datos almacenado. Por ejemplo, los datos se guardan como pares clave/valor, como documentos JSON (*JavaScript Object*

---

1 REST (*Representational State Transfer*, Transferencia de Estado Representacional), estilo arquitectónico creado por Roy Fielding en su tesis doctoral en el año 2000.

2 Propuesto por Edgar Frank Codd en 1970. Dicho modelo organiza los datos en una o más relaciones de atributos (columnas) y tuplas (filas), con una clave única identificando cada tupla.

*Notation*, Notación de Objeto de JavaScript), o como grafos consistentes de aristas y vértices.

#### **1.1.14 Apache Cassandra™**

Apache Cassandra™ es un sistema de base de datos distribuido transaccional y escalable que ofrece alta disponibilidad, capaz de soportar enormes conjuntos de datos en *clusters* con miles de nodos desplegados en múltiples centros de datos [17], sin un punto de fallo principal [18].

#### **1.1.15 Apache Spark™**

Apache Spark™ es un motor analítico unificado para el procesamiento de datos a gran escala, que alcanza un alto rendimiento para datos por lotes o en flujo continuo, usando un programador de tareas DAG (*Directed Acyclic Graph*, Grafo Acíclico Dirigido) de última generación, un optimizador de consultas y un motor de ejecución físico. Apache Spark™ posee un fundamento arquitectónico basado en el RDD<sup>1</sup> (*Resilient Distributed Dataset*, Conjunto de datos Resistente Distribuido), un multi-conjunto de elementos de datos de sólo lectura, distribuido sobre un *cluster* de máquinas, que se mantiene en un estado tolerante a fallos [19].

#### **1.1.16 OpenCV**

OpenCV (*Open Source Computer Vision Library*, Biblioteca de Código Abierto para Visión por Computadora) se distribuye bajo una licencia BSD (*Berkeley Software Distribution*, Distribución de *Software* de Berkeley) y por lo tanto es libre para el uso académico y comercial. Tiene interfaces para C++, Python y Java y soporta Windows, Linux, Mac OS, iOS y Android. OpenCV presenta un diseño enfocado en la eficiencia computacional y con un enfoque en aplicaciones en tiempo real. Codificada en C/C++ optimizado, la biblioteca toma ventaja del procesamiento multi-núcleo [20].

---

<sup>1</sup> Una abstracción de memoria distribuida que permite a los desarrolladores realizar cómputos en memoria, en *clusters* de gran tamaño manteniendo la tolerancia a fallos.

### 1.1.17 Java

Java es un lenguaje de programación de propósito general, concurrente, basado en clases, orientado a objetos y diseñado específicamente para minimizar dependencias de implementación. Su principal intención es permitir a los desarrolladores “codificar una vez, ejecutar en cualquier lugar”, eslogan que significa que el código Java compilado funciona en todas las plataformas que soporten el lenguaje sin la necesidad de re-compilación [21]. Extensión de archivo con código fuente: `.java`.

### 1.1.18 Scala

Scala es un lenguaje de programación multi-paradigma, diseñado para expresar patrones de programación comunes, en forma concisa, elegante y con verificación segura de tipos de datos. Scala integra elementos de lenguajes orientados a objetos, y funcionales [22].

### 1.1.19 Python

Python es un lenguaje de programación que posee estructuras de datos de alto nivel eficientes y un enfoque simple y efectivo a la programación orientada a objetos. Su sintaxis elegante y sistema de tipos dinámico, así como su naturaleza interpretada, lo convierten en un lenguaje ideal para *scripting* y el desarrollo rápido de aplicaciones [23]. Extensión de archivo con código fuente: `.py`.

### 1.1.20 Angular

Angular es un marco de trabajo para aplicaciones web, de código abierto y basado en TypeScript<sup>1</sup>. Este marco de trabajo es un producto del Equipo Angular en Google™. Angular facilita la construcción de aplicaciones para la *Web*, combinando plantillas declarativas, inyección de dependencias, y mejores prácticas integradas para abordar los retos del desarrollo. Angular permite el desarrollo de aplicaciones para la *Web*, móviles y de escritorio.

---

<sup>1</sup> Súper-conjunto de JavaScript que permite trabajar con tipos de datos y compila a JavaScript simple. Ejecutable en cualquier visualizador *web* en cualquier sistema operativo.

### 1.1.21 Node.js®

Node.js es un entorno de servidor libre y de código abierto, capaz de ejecutarse en varias plataformas (Windows, Linux, Unix, macOS, entre otros). Node.js usa JavaScript como lenguaje en el servidor. A diferencia de lenguajes como PHP, Java o ASP, que funcionan de manera síncrona – reciben la petición del cliente, esperan mientras el sistema de archivos abre y lee el archivo solicitado, retornan el contenido y, sólo entonces, vuelven a estar listos para la próxima petición – Node.js posee una naturaleza intrínsecamente asíncrona, lo cual le permite enviar la petición al sistema de archivos y estar listo para procesar la próxima sin necesidad de esperar respuesta de la anterior. Node.js ejecuta una programación de hilo único de ejecución, asíncrona y sin bloqueo, lo cual es muy eficiente a nivel de memoria. Node.js genera páginas con contenido dinámico; crear, abrir, leer, escribir, eliminar y borrar archivos en el servidor; recopilar datos de un formulario y, añadir, eliminar y modificar información en la base de datos [24].

### 1.1.22 Raspberry Pi 3 B+ (RPi3B+)

Raspberry Pi es una serie de SBC (*Small Board Computer*, Computadora de Placa Pequeña) que permite conexión a un monitor de computadora o televisor, y usa teclado y ratón estándares. En 2018 surge el modelo B+, el cual presenta un procesador a 1.4 GHz de frecuencia, con conexión Gigabit Ethernet (transmisión limitada a cables de 300 Mbit/s a través de la conexión USB 2.0 interna) o Wi-Fi doble banda a 2.4/5 GHz (100 Mbit/s). El dispositivo también ofrece capacidades PeE (*Power over Ethernet*, Potencia sobre Ethernet), e inicio por USB y por red.

### 1.1.23 Fitbit Inc.

Fitbit, Inc., es un proveedor de dispositivos para salud física. La plataforma de la compañía combina la conexión entre estos dispositivos con *software* y servicios, incluyendo un panel de control en línea y aplicaciones para dispositivos móviles, analítica de datos, herramientas sociales y motivacionales, por mencionar algunos. Su plataforma incluye una familia de dispositivos ponibles que monitorizan el estado físico y la salud. Entre dichos dispositivos se encuentran Fitbit Zip, Fitbit One, Fitbit Flex,

Fitbit Flex 2, Fitbit Alta, Fitbit Charge, Fitbit Blaze, Fitbit Charge HR, Fitbit Charge 2, Fitbit Surge, por mencionar algunos. Estos dispositivos registran automáticamente los pasos, calorías quemadas, distancia viajada y minutos activo del usuario, así como el ritmo cardíaco, duración y calidad del sueño, mostrando retroalimentación en tiempo real [25].

## **1.2 Situación tecnológica, económica y operativa de la empresa**

El Instituto Tecnológico de Orizaba es una institución de educación superior orientada al desarrollo de la ciencia y la técnica mediante la formación de recursos humanos y la investigación en dichas áreas, para el fortalecimiento económico y social de México. Es un establecimiento público ubicado en la ciudad de Orizaba, en el estado de Veracruz, México. El instituto depende de la Secretaría de Educación Pública (SEP) y pertenece al Tecnológico Nacional de México [26]. Su misión es fortalecer los servicios educativos a través de la cobertura, equidad, promoción e inclusión, en la formación integral de los estudiantes impulsando la innovación, ciencia y tecnología; para consolidar la vinculación con pertinencia en los diferentes sectores estratégicos, modernizando la gestión institucional con transparencia y rendición de cuentas en un ámbito sustentable [27].

## **1.3 Planteamiento del problema**

A medida que avanza el envejecimiento de la población mundial y aumenta la prevalencia de condiciones médicas crónicas, el tele-cuidado es cada vez más utilizado en asistir a las personas más longevas o frágiles, permitiéndoles permanecer en sus hogares y continuar con su modo de vida de manera relativamente independiente. Los gobiernos y proveedores de cuidados de la salud también están enfocados en el uso del tele-cuidado para reducir las admisiones a hospitales y los gastos que esto conlleva [6].

Esta situación genera una demanda continua de soluciones de tele-cuidado novedosas e innovadoras. Al igual que varios campos de la investigación y los negocios en Internet, la medicina y el cuidado de la salud encuentran una plataforma de evolución

en el IoT y la proliferación de dispositivos que la sustentan, conllevando a la aparición del paradigma del Internet de las Cosas Médico (mIoT, por sus siglas en inglés). Este fenómeno revolucionó la manera en que los servicios de tele-cuidado, tele-medicina y, en general, tele-salud, se presentan a los usuarios.

## **1.4 Objetivos general y específicos**

### **1.4.1 Objetivo General**

Desarrollar un sistema de tele-cuidado basado en el Internet de las Cosas para monitorizar, de manera remota, signos vitales y actividades de adultos mayores, generar alertas y recomendaciones basadas en Analítica Big Data para la prevención y detección de accidentes.

### **1.4.2 Objetivos Específicos**

- 1) Analizar el estado del arte de la investigación en tele-cuidado y Analítica Big Data para el cuidado de la salud.
- 2) Analizar dispositivos domóticos para videovigilancia de actividades en adultos mayores.
- 3) Diseñar la arquitectura tecnológica de la plataforma que permita la integración de, al menos un dispositivo vestible o ponible, un dispositivo domótico de videovigilancia, módulos para el procesamiento de datos y Analítica Big Data, y aplicaciones para dispositivos móviles y *web*.
- 4) Desarrollar un módulo de recopilación de datos procedentes de dispositivos vestibles o ponibles, y dispositivos domóticos de videovigilancia para monitorizar signos vitales y actividades de adultos mayores.
- 5) Desarrollar un módulo para la detección de anomalías en signos vitales y actividades de adultos mayores, para la generación de alertas y recomendaciones basadas en Analítica Big Data.
- 6) Desarrollar las aplicaciones para dispositivos móviles y visualizadores *web*, para la interacción de los usuarios con el sistema.

- 7) Validar el sistema implementado mediante el diseño y la conducción de, al menos, un par de casos de estudio de prevención y manejo de incidentes y accidentes en el cuidado de la salud de adultos mayores.

## 1.5 Justificación

Existen tres funciones relacionadas, de manera general, con los sistemas de telecuidado: 1) Monitorización de la seguridad en el hogar, por ejemplo, mediante el uso de detectores de caídas; 2) Monitorización de signos vitales y, 3) Información y soporte vía telefónica o por Internet [6].

La popularidad de estos sistemas radica principalmente en dos aspectos: 1) La posibilidad de permitir a personas de la tercera edad y a personas con condiciones crónicas mantener su independencia al continuar viviendo en sus hogares o residencias y, 2) Reducir costos asociados al cuidado de la salud al controlar las admisiones en hospitales.

El campo de BDA juega un rol importante en la evolución de las prácticas del cuidado de la salud, proveyendo herramientas para acumular, administrar, analizar y asimilar grandes volúmenes de datos heterogéneos, estructurados y no estructurados, generados por sistemas de cuidado de la salud existentes.

A pesar de las complejidades inherentes de los datos del cuidado de la salud, existe potencial y beneficio en desarrollar e implementar soluciones Big Data en este contexto [28].

El presente trabajo de tesis propone una arquitectura basada en IoT, procesando la información proveniente de, al menos, un DVP, y un dispositivo doméstico – cámara de videovigilancia. Ambos dispositivos proveerán medios de monitorización constante sobre el usuario – signos vitales y entorno – transmitiendo información de manera constante. Dicha información se procesará y analizará haciendo uso de técnicas BDA para descubrir patrones y situaciones de peligro potenciales para el usuario, con base en esto se emitirán alertas a las entidades pertinentes a través de una aplicación *web*.

## Capítulo 2 Estado de la práctica

Como primera fase del proceso de investigación, se realizó un estudio del estado del arte de las tecnologías relacionadas con Big Data y el paradigma IoT en el mundo de las comunicaciones. Todos los estudios revisados reafirman de una manera u otra el constante crecimiento y omnipresencia de estos fenómenos, y cómo continúan cambiando la percepción de la comunidad de desarrolladores e investigadores hacia los procesos de almacenamiento y análisis de información. El inmenso volumen de datos generado por los billones de dispositivos interconectados a través de Internet plantea un reto real en el manejo y control de la información, pero al mismo tiempo crea una oportunidad sin precedentes para el aprovechamiento de la misma, abriéndose canales de comunicación no concebidos hace más de dos décadas. Esto es aún más cierto en el campo de la salud; el advenimiento de IoT y Big Data revoluciona los conceptos de tele-salud, tele-cuidado y tele-medicina, cambiando el enfoque con que se enfrenta el cuidado de pacientes, especialmente aquellos con enfermedades crónicas y los adultos mayores.

### 2.1 Trabajos relacionados

El impacto de las técnicas de análisis Big Data en organizaciones de cuidado de la salud para implementar y presentar casos de uso novedosos para potenciales *frameworks* (marcos de trabajo) en el cuidado de la salud, muestra líneas de investigación prometedoras. La información obtenida a partir de sistemas de cuidado de la salud se clasifica en tres tipos de manera general: 1) Estructurada, 2) Semi-estructurada y 3) No estructurada. La variedad de fuentes de información y sus posibles formatos, o la ausencia de uno, presenta un reto al intentar construir *frameworks* eficientes y viables para su recolección y procesamiento.

En [29] se analizó el impacto de Big Data en el cuidado de la salud desde la perspectiva de varias partes interesadas. Dichas partes incluyen los pacientes, médicos, operadores de hospitales, investigadores clínicos y farmacéuticos y aseguradores.

Se revisaron dieciocho *frameworks*, agrupándolos por fuentes de datos, capacidades analíticas y áreas de aplicación. También se estudiaron las implicaciones de cinco herramientas Big Data para el desarrollo de *frameworks* para el cuidado de la salud, y se resumieron los resultados más significativos en el intento de establecer varios *frameworks* en el campo del cuidado de la salud, tales como la asistencia computacional dirigida por datos, plataformas Big Data para el procesamiento de signos vitales, la integración de tecnologías Big Data, en la “nube” y móviles, la obtención de registros médicos basados en síntomas de acuerdo a la consulta del usuario, por mencionar algunos.

Dinesen et al. [30] señalaron la importancia y necesidad inminente del avance y evolución de soluciones de tele-salud innovadoras y de última generación en la gestión de enfermedades crónicas. Se describieron los retos implicados en promover la implementación de la tele-salud y avanzar en su adopción, así mismo, se presentó un programa de investigación global para tele-salud personalizada para la gestión de enfermedades crónicas. Además, los autores abordaron situaciones problemáticas referentes a reembolsos, licencias y jurisdicciones, específicamente en los Estados Unidos y la Unión Europea.

Este programa de investigación busca proveer una orientación comprensiva en un contexto global para la tele-salud personalizada, asimismo requiere refinamientos específicos para lidiar con las variaciones de los sistemas de salud de cada país.

En [31] se exploraron los avances recientes en BDA para sistemas IoT, así como los requerimientos clave para gestionar Big Data y habilitar procesos analíticos en un entorno IoT. Se identificaron oportunidades que surgen de la convergencia entre Big Data, procesos de análisis e IoT, también se discutió el rol de BDA en aplicaciones IoT.

Entre los requerimientos para el procesamiento y análisis Big Data explicados por Ahmed et al. [31], se encuentran la conectividad, el almacenamiento, la calidad del servicio, análisis en tiempo real y las pruebas, en el entorno IoT. Los autores también

analizaron plataformas como Apache Hadoop, 1010Data, Cloudera Data Hub, SAP-Hana (sistema de gestión de base de datos relacional, en memoria y orientado a columnas), HP-HAVE*n* (*Hewlett-Packard - Hadoop Autonomy Vertica ArcSight*), Hortonworks, Pivotal Big Data suite, Infobright y MapR. Finalmente, presentaron una taxonomía de soluciones analíticas y Big Data para IoT y analizaron las líneas de investigación abiertas, realizándose la visión de BDA en IoT como direcciones futuras de investigación.

Angelova et al. [32] identificaron patrones en el uso de dispositivos y asociaciones entre dichos patrones. Mediante análisis estadísticos y aprendizaje automático, los autores establecieron la relación entre características de los usuarios y las activaciones de los dispositivos. Para llevar a cabo su investigación, aplicaron reglas de asociación y árboles de decisión para analizar los eventos, y una técnica de persecución de proyección dirigida para el modelado usuario-evento. Los resultados obtenidos revelaron una fuerte asociación entre la edad de los usuarios y las activaciones, o sea, diferentes grupos de edades exhiben diferentes patrones de activación. Además, un análisis enfocado en usuarios con problemas de salud mental demostró que los usuarios más envejecidos con problemas de memoria y que viven solos tienen mayor probabilidad de cometer errores en el uso de los dispositivos.

Los patrones encontrados permiten al centro de ayuda de tele-cuidado ganar conocimiento detallado sobre sus operaciones y mejorar su efectividad. Este estudio contribuyó al análisis automático y apoyo en la toma de decisiones en la industria del tele-cuidado.

Barlow et al. [6] condujeron un estudio sistemático de tele-cuidado en casa para personas mayores frágiles y pacientes con condiciones crónicas. Los autores evaluaron varias pruebas de control de naturaleza aleatoria, las cuales examinaban los efectos del uso de tecnologías de telecomunicación para monitorizar signos vitales o la seguridad en el hogar, o proveer información y soporte. Según la evidencia revisada, las intervenciones de tele-cuidado más efectivas son la monitorización automática de signos vitales (para reducir el uso del servicio de salud) y el seguimiento

vía telefónica por enfermeros (as) (para mejorar indicadores clínicos y reducir el uso del servicio de salud).

La evidencia sugiere que el tele-cuidado ayuda a mejorar la atención a personas con condiciones crónicas y ancianos frágiles, pero la fuerza de dicha evidencia depende del tipo de aplicación de tele-cuidado.

El ritmo de adopción y desarrollo investigativo de Big Data son retos analizados por Belle et al. [28], enfocándose en tres áreas prometedoras de la investigación médica: 1) Análisis basados en imágenes, 2) Análisis de señales y 3) Las ciencias genómicas. Los autores discutieron una reciente investigación acerca de la utilización de grandes volúmenes de información médica combinando datos multimodales provenientes de fuentes heterogéneas. Los autores examinaron áreas potenciales de investigación que impactan significativamente en el cuidado de la salud. Los procesos de análisis Big Data para gestionar cantidades descomunales de información, jugarán un rol esencial en la forma en que se desarrollará el cuidado de la salud en el futuro. Análisis de imágenes médicas, procesamiento de señales de información fisiológica y la integración de información fisiológica con genómica presentan oportunidades similares en el proceso de lidiar con información dispar estructurada y no estructurada.

Nicolalde et al. [33] analizaron intentos investigativos de última generación de datos IoT. Los autores exploraron el potencial impacto de los retos en grandes cantidades de información, esfuerzos de investigación dirigidos hacia el análisis de datos IoT y varias herramientas asociadas con dicho análisis. Los autores presentaron una visión general de los conceptos Big Data, IoT y BDA como tecnologías y cómo se manifiestan en el mundo actual de la información. También se exploraron los retos actuales de Big Data, como el almacenamiento y análisis de información, descubrimiento de conocimiento y complejidades computacionales, escalabilidad, visualización y seguridad de la información, así como problemas de investigación abiertos en IoT para BDA. Los autores estudiaron siete herramientas para procesamiento Big Data: Apache Hadoop y MapReduce, Apache Spark™, Dryad, Apache Drill, Storm, Splunk,

Jaspersoft y Apache Mahout. Finalmente, sugirieron el uso de plataformas para la exploración de Big Data en varias fases y entender mejor el conocimiento que se extrae, abriendo nuevos horizontes para desarrollar soluciones basadas en tópicos y retos de investigación abiertos.

Dimitrov [34] realizó un análisis de mIoT (*Medical Internet of Things*, Internet Médico de las Cosas) y Big Data en los campos del cuidado de la salud, siendo el primero una pieza clave en la transformación digital del cuidado de la salud, ya que permite el surgimiento de nuevos modelos de negocio y posibilita cambios en procesos de trabajo, mejoras en la productividad, contención de costos y una mejor experiencia para el cliente. El autor señala la importancia del flujo de información existente como la fuerza impulsora de IoT. También se discutió el futuro de las empresas farmacéuticas en el entorno IoT, y cómo algunas de estas empresas se asociaron a entidades como Google en el desarrollo de tecnologías robóticas y de sensores. Finalmente, Dimitrov se refirió a los retos de IoT y Big Data en el contexto del cuidado de la salud.

Islam et al. [35] presentaron una revisión de la literatura sobre analítica en el cuidado de la salud usando minería de datos y Big Data. Siguiendo PRISMA (*Preferred Reporting Items for Systematic Reviews and Meta-Analyses*, Elementos de Reporte Preferidos para Revisiones Sistemáticas y Meta-Análisis), los autores condujeron una búsqueda en registros de bases de datos del período 2005 a 2016. Elementos cruciales de los estudios seleccionados—sub-áreas del cuidado de la salud, técnicas de minería de datos, tipos de analíticas, datos y fuentes de datos—se extrajeron para proveer una revisión sistemática de desarrollo en este campo y posibles direcciones futuras. Se encontró que la literatura existente examina mayormente analíticas en la toma de decisiones clínica y administrativa. Predomina el uso de datos generados por humanos, considerando la adopción masiva de EMR (*Electronic Medical Records*, Registros Médicos Electrónicos) en el cuidado clínico. Sin embargo, la analítica basada en información de sitios *Web* y medios sociales ha aumentado durante los últimos años. La falta de analítica prescriptiva en la práctica e integración de conocimiento de

expertos de dominio en el proceso de toma de decisiones enfatiza la necesidad de investigación futura.

Yacchirema et al. [36] presentaron un sistema innovador para detectar y apoyar el tratamiento de la Apnea de Sueño Obstructiva (ASO) de personas mayores con la monitorización de múltiples factores como el entorno de sueño, estado del sueño, actividades físicas y parámetros fisiológicos, así como el uso de información abierta disponible en “ciudades inteligentes”. Los autores crearon una arquitectura de sistema que realiza dos tipos de procesamiento: 1) Un pre-procesamiento basado en reglas que permite enviar notificaciones en tiempo real al responsable del cuidado del adulto mayor, y 2) Un procesamiento de datos por lotes que posibilita un análisis descriptivo que detalla estadísticamente el comportamiento de la información y un análisis predictivo para el desarrollo de servicios.

Lee y Yoon [37] presentaron un estudio sobre el estado actual de mBD (*medical Big Data*, Big Data médico) y el hecho de su relativo atraso en el contexto de Big Data al tratar de presentar conocimiento accionable, al contrario de otras disciplinas como los negocios y las finanzas en Internet. En su artículo, los autores discutieron tópicos como el concepto de mBD, sus particularidades, cómo se analiza y cuáles son sus retos. También se mencionaron áreas de BDA para el mejoramiento del cuidado de la salud, como son el modelado predictivo para riesgos y el uso de recursos, gestión de población, vigilancia de la seguridad de drogas y dispositivos médicos, la heterogeneidad de tratamientos y enfermedades, medicina de precisión y soporte en decisiones clínicas, por mencionar algunos. Los autores también expusieron formas en las que algoritmos de minería de datos se utilizan en un contexto clínico; metas analíticas de mBD como la predicción, el modelado y la inferencia se alcanzan a través de clasificación, agrupamiento y regresión. Finalmente, los autores trataron algunos de los retos para mBD como la escasez de beneficios prácticos, problemas en la consistencia, calidad y estabilidad de los datos, las limitaciones de los estudios observacionales, la validación, así como la integración y utilidad clínica.

El manejo de cambios de contexto frecuentes y dinámicos es una tarea difícil sin disponer de una plataforma de adquisición y procesamiento de eventos/información en tiempo real. Big Data e IoT se propusieron recientemente para el análisis en tiempo real de flujos de información (es decir: datos, eventos). Con base en lo anterior, en [38], se propuso una combinación de técnicas IoT con tecnologías Big Data en una plataforma holística para el monitorización y procesamiento continuo de información en tiempo real. Los autores presentaron una arquitectura para procesar datos de sensores en tiempo real, compuesta por tres capas: 1) Capa de adquisición de datos, 2) Capa de procesamiento de datos y, 3) Capa de almacenamiento y visualización de datos. Los experimentos preliminares se realizaron y los resultados demostraron que dicha plataforma es útil en una situación real.

Marjani et al. [2] estudiaron esfuerzos investigativos del estado del arte dirigidos a IoT y BDA y explicaron la relación entre BDA e IoT. Además, su estudio añadió valor al proponer una nueva arquitectura para IoT y BDA. Los autores también discutieron los tipos y métodos de analíticas de datos IoT, y tecnologías para minería Big Data, así como varios casos de uso. Se examinaron varias oportunidades generadas por la analítica de datos en el paradigma IoT. Finalmente, se presentaron retos de investigación abiertos, tales como la privacidad, minería Big Data, visualización e integración, como direcciones de investigación futuras.

En [39], se propuso, validó y evaluó, una arquitectura *Fog Data* orientada a servicios para el cómputo en “la niebla”. La pieza central de la arquitectura propuesta es una computadora embebida de bajo consumo energético que realiza minería y análisis de datos sobre información no procesada, recolectada por varios sensores en *wearables* usados en aplicaciones de tele-salud. Se construyó un prototipo funcional de la arquitectura para realizar casos de estudio en aplicaciones Big Data de tele-salud. Los autores también evaluaron cuantitativamente varias técnicas de minería de datos y técnicas de compresión estándares. Los resultados obtenidos mostraron una mejora substancial en la eficiencia del sistema al usar la arquitectura *Fog Data*.

Lee et al. [40], examinaron los retos en el diseño de algoritmos y sistemas para la analítica en el cuidado de la salud, además realizaron un estudio de soluciones relevantes. Los autores también discutieron aplicaciones, servicios y sistemas en el cuidado de la salud de nueva generación, que están relacionados con BDA en el cuidado de la salud. El artículo de los autores, se enfocó en el procesamiento de EHR (*Electronic Health Records*, Registros de Salud Electrónicos), grandes cantidades de datos médicos heterogéneos disponibles en varias organizaciones del cuidado de la salud y sensores (por ejemplo, *wearables*). Se presentaron datos provenientes de sensores y EMR como subconjuntos de EHR. La alta dimensionalidad (selección y extracción de características), irregularidad (uso de características base, transformación de datos y el uso directo de datos irregulares), datos perdidos y escasez de datos, ruido, parcialidad y base de conocimientos se identificaron como retos actuales. Lee et al. [40] también presentaron los pasos del proceso a realizar sobre EHR antes de su análisis, comenzando con la anotación de datos, siguiendo con la limpieza e integración de los datos y culminando con el modelado y la analítica de los mismos.

## 2.2 Análisis comparativo

La TABLA I muestra, de manera concreta, un resumen de cada artículo mencionado en la sección anterior, con el fin de realizar un análisis comparativo.

Artículo	Problema	Contribución	Tecnologías	Arquitectura	Resultados	Estatus
Palanisamy y Thirunavukarasu [29]	Análisis comparativo de <i>frameworks</i> para el procesamiento y análisis Big Data.	Presentar varios escenarios analíticos para el sistema de cuidado de la salud centrado en pacientes, desde la perspectiva de varias partes interesadas. Revisar varios <i>frameworks</i> Big Data con respecto a sus fuentes de datos subyacentes, capacidad analítica y áreas de aplicación. Presentar las implicaciones de herramientas Big Data en el desarrollo de eco	No se mencionan.	No	Las soluciones basadas en <i>frameworks</i> siempre aportan a los requerimientos comprensivos de las partes interesadas involucradas en el cuidado de la salud.	Terminado.

Artículo	Problema	Contribución	Tecnologías	Arquitectura	Resultados	Estatus
		sistemas de cuidado de la salud.				
Dinesen et al. [30]	Necesidad de un enfoque global en la implementación de un programa de investigación para la tele-salud personalizada.	Describir los retos de promover la implementación de la tele-salud y avanzar su adopción. Presentar un programa de investigación global para la tele-salud personalizada en la gestión de enfermedades crónicas. Un programa de investigación de 12 puntos para aplicaciones de tele-salud futuras con gestión de enfermedades crónicas.	No se mencionan.	No	Facilitación de evaluaciones comparativas de soluciones de tele-salud desde un nivel individual hasta nivel de sistema, usando una variedad de dispositivos y tecnologías, y en contextos y configuraciones disímiles.	Terminado.
Ahmed et al. [31]	La gestión de Big Data en	Exploración de avances recientes en BDA	No se mencionan.	No	Los sistemas IoT tienen el potencial de resolver	Terminado.

Artículo	Problema	Contribución	Tecnologías	Arquitectura	Resultados	Estatus
	redes de continuo crecimiento propicia el surgimiento de asuntos no triviales relacionados con la recolección eficiente de información, su procesamiento, análisis y seguridad.	para sistemas IoT, así como los requerimientos clave para la gestión de Big Data y para habilitar el análisis en un entorno IoT. Taxonomización de la literatura basada en parámetros importantes. Identificación de oportunidades a partir de la convergencia de Big Data, los procesos analíticos e IoT.			muchos problemas, pero aún quedan numerosos retos que enfrentar. Las soluciones a muchos de estos retos provienen de BDA, mientras que otros requieren de esfuerzos concentrados de la comunidad IoT, fabricantes de plataformas y <i>hardware</i> , gobiernos y hacedores de políticas.	
Angelova et al. [32]	Estudio de patrones en la activación de dispositivos y sensores en	Posibilitar al centro de tele-cuidado obtener información detallada sobre sus operaciones y mejorar su efectividad.	No se mencionan.	No	Fuerte asociación entre la edad de los usuarios y las activaciones de dispositivos. Usuarios	Terminado.

Artículo	Problema	Contribución	Tecnologías	Arquitectura	Resultados	Estatus
	el área de tele-cuidado utilizando análisis estadístico y aprendizaje automático.	Aportar análisis automático y soporte a la toma de decisiones en la industria del tele-cuidado. Uso e instalación efectivos del equipamiento. Aumentar la eficiencia de visitas de cuidadores y reparadores.			de mayor edad con problemas de memoria que viven solos tienen mayor probabilidad de cometer errores en el uso de los dispositivos.	
Barlow et al. [6]	Revisión sistemática de los beneficios del tele-cuidado en el hogar.	Se incluyeron noventa y ocho estudios de naturaleza aleatoria en la revisión. Se obtuvieron resultados a nivel individual y de sistema en las áreas de monitorización de signos vita-	No se mencionan.	No	La evidencia sugiere que el tele-cuidado mejora la atención a personas con condiciones crónicas y a ancianos frágiles, en mayor o menor medida en dependencia de la aplicación del tele-cuidado.	Terminado.

Artículo	Problema	Contribución	Tecnologías	Arquitectura	Resultados	Estatus
		les, seguridad y servicios de información y soporte.			Monitorización de signos vitales: la evidencia sobre los efectos de la monitorización a distancia en resultados clínicos fue inconsistente. Monitorización de seguridad: existe insuficiente evidencia rigurosa acerca de los efectos de sistemas de alerta como detectores de caídas y alarmas comunitarias. Provisión de información: el apoyo proactivo por parte de facultativos por vía telefónica mejora los resultados clínicos. El seguimiento por	

Artículo	Problema	Contribución	Tecnologías	Arquitectura	Resultados	Estatus
					teléfono luego de la salida del hospital también estaba asociado con una reducción en el uso de servicios de salud.	
Belle et al. [28]	El ritmo de aceptación y el desarrollo investigativo en BDA aún es obstaculizado por problemas fundamentales del paradigma Big Data.	Una revisión de la información producida por técnicas de imagen y la aplicación de imágenes médicas, así como los métodos usados para aprovechar los beneficios de la naturaleza multi-modelo de la información. Un panorama general de los retos y enfoques	No se mencionan.	No	Se necesitan nuevos <i>frameworks</i> y métodos para analizar imágenes médicas con diferentes modalidades en un contexto clínico. Así se aumentaría la exactitud del diagnóstico utilizando fuentes dispares para reducir costos y optimizar el mejoramiento, registro y segmentación de imáge-	Terminado.

Artículo	Problema	Contribución	Tecnologías	Arquitectura	Resultados	Estatus
		<p>existentes en el desarrollo de sistemas de monitorización de información en forma de onda de alta fidelidad e información discreta de fuentes no continuas: adquisición, almacenamiento y agregación de información, y analítica de señales usando Big Data.</p> <p>Un estudio de <i>Pathway Analysis</i> y <i>Reconstruction of Regulatory Networks</i> como ejemplos de aplicaciones de Big Data en la genómica.</p>			<p>nes médicas, entregando mejores recomendaciones a nivel clínico.</p> <p>El estado actual de la competencia y estructura no estandarizada de la información permite al procesamiento de señales de datos fisiológicos, ayudar en las comunidades de práctica e investigación en el cuidado de la salud.</p> <p>La regulación de genes, los efectos de diferentes arquitecturas de</p>	

Artículo	Problema	Contribución	Tecnologías	Arquitectura	Resultados	Estatus
					<p>redes y los efectos evolutivos en redes a escala de genomas aún están bajo análisis. La combinación de diseños minuciosos de experimentos y desarrollo de modelos para la reconstrucción de estas redes resultará en un proceso de comprensión más eficiente de las mismas.</p>	
Nicolalde et al. [33]	El uso de BDA en los retos de IoT.	Se discutió la relación entre BDA e IoT. Se examinaron varios temas de investigación y oportunidades generadas por el análisis de	No se mencionan.	No	A partir de esta investigación, se entiende que cada plataforma Big Data presenta un enfoque individual; algunas están diseñadas para el procesamiento	Terminado.

Artículo	Problema	Contribución	Tecnologías	Arquitectura	Resultados	Estatus
		información en el paradigma IoT.			por lotes, mientras que otras son mejores para el análisis en tiempo real. Las diferentes técnicas usadas en el estudio incluyen el análisis estadístico, el aprendizaje automático, minería de datos, análisis inteligente, cómputo en la nube y procesamiento de flujo de datos. Es deseable que los investigadores presten mayor atención a estas técnicas en el futuro, para resolver problemas de grandes	

Artículo	Problema	Contribución	Tecnologías	Arquitectura	Resultados	Estatus
					cantidades de información efectiva y eficientemente.	
Dimitrov [34]	Este artículo revisa mIoT y Big Data en los campos del cuidado de la salud.	Las aplicaciones para dispositivos móviles y los <i>wearables</i> apoyan en la educación de la salud, el seguimiento de síntomas, la gestión colaborativa de enfermedades y la coordinación del cuidado. Todas estas plataformas de análisis aumentan la relevancia de la interpretación de datos, reduciendo la demora comunicación de información al usuario. El conocimiento obtenido por	No se mencionan.	No	Surgirá una nueva categoría de “mentores personalizados de salud preventiva” (Consejeros de Salud Digital). Estos trabajadores tendrán la habilidad de entender e interpretar datos de salud y bienestar. Ayudarán a sus clientes a evitar males crónicos y relacionados con la dieta, mejorar su salud mental y alcanzar mejores estilos de vida. En la misma manera en	Terminado.

Artículo	Problema	Contribución	Tecnologías	Arquitectura	Resultados	Estatus
		BDA impulsará la disrupción digital del mundo del cuidado de la salud, los procesos de negocios y toma de decisiones en tiempo real.			que la población mundial envejece, estos roles son cada vez más importantes.	
Islam et al. [35]	Existe una falta de narrativa comprensiva y sistemática en el tema de aplicaciones de minería de datos en el cuidado de la salud.	En lo que a los autores respecta, este artículo es el primero en tomar un enfoque de revisión abarcadora y ofrecer una visión holística del cuidado de la salud en la literatura del cuidado de la salud y la minería de datos. El exhaustivo y metodológicamente riguroso enfoque adoptado	La metodología usada siguió las pautas propuestas por PRISMA. Se evaluó la calidad de los artículos seleccionados usando la Lista de Evaluación	No	La creciente industria del cuidado de la salud genera un gran volumen de información útil: la demográfica de pacientes, planes de tratamiento, pagos y seguros médicos, atrayendo la atención de clínicos y científicos por igual. Entre las recomendaciones de investigaciones futuras se	Terminado.

Artículo	Problema	Contribución	Tecnologías	Arquitectura	Resultados	Estatus
		<p>tado, abarca la perspectiva teórica y de aplicación de analítica y minería de datos en el cuidado de la salud. Este enfoque sistemático, comenzando con el proceso de revisión y categorizando el resultado como analítico o teórico, provee a los lectores con una revisión más diseminada con referencia a campos específicos.</p>	<p>Crítica de JBI (<i>Joanna Briggs Institute</i>, Instituto Joanna Briggs) para estudios analíticos transversales y las pautas de investigación cualitativa de CASP (<i>Critical Appraisal Skills Programme</i>, Programa de Habilida-</p>		<p>encuentran la integración del conocimiento de expertos del dominio, enfoques para disminuir errores de predicción, y la integración de modelos predictivos en entornos de trabajo reales.</p>	

Artículo	Problema	Contribución	Tecnologías	Arquitectura	Resultados	Estatus
			des de Evaluación Crítica).			
Yacchirema et al. [36]	ASO posee un impacto negativo directo en la calidad de vida; el deterioro intelectual, rendimiento psicomotor disminuido y desórdenes de comportamiento y personalidad son algunas consecuencias. Por lo tanto, la	Una arquitectura de sistema basada en IoT y Big Data que se enfoca en la detección de ASO, así como en guiar el tratamiento de este síndrome, con base en el procesamiento de información abierta disponible en “ciudades inteligentes” y en múltiples factores que afectan directamente la ASO tales como el entorno de sueño, estado de sueño, actividades	Sensores de temperatura y humedad. Sensores de movimiento (acelerómetro y giroscopio). Monitor de ritmo cardíaco. Módulo de sensor de sonido de tres clavijas para detectar la intensidad	Una arquitectura de un sistema de monitorización de apnea de dueño basada en IoT y Big Data, otra de procesamiento en “la niebla” y una	Los experimentos realizados muestran un 93.3% de efectividad en la predicción del Índice de Calidad de Aire para guiar el tratamiento de ASO. Se evaluó el rendimiento del sistema en términos de latencia. Los resultados demuestran que el pre-procesamiento de los datos al borde de la red mejora la eficiencia del sistema.	Terminado.

Artículo	Problema	Contribución	Tecnologías	Arquitectura	Resultados	Estatus
	<p>monitorización en tiempo real de este desorden es una necesidad crítica en soluciones de cuidado de la salud.</p>	<p>físicas, y parámetros fisiológicos.</p>	<p>de los ruidos inalámbricos de bajo consumo energético tales como Bluetooth 4.0, Zigbee y 6LowPAN (Protocolo de Internet v6 sobre redes inalámbricas personales de bajo consumo energético). Los</p>	<p>de procesamiento en “la nube”.</p>		

Artículo	Problema	Contribución	Tecnologías	Arquitectura	Resultados	Estatus
			datos históricos se almacenan en un <i>cluster</i> de Sistema de Archivos Distribuidos con Hadoop, y explotados usando la plataforma Apache Spark™.			
Lee y Yoon [37]	Progreso retardado de mBD en el sector del cui-	Un estudio sobre mBD, incluyendo una comparación entre el análisis mBD y el análisis clásico estadístico, y una descripción de minería	No se mencionan.	No	Big Data es necesario, pero no suficiente, la simple acumulación de datos no tiene valor si no permiten ser analizados de forma tal que	Terminado

Artículo	Problema	Contribución	Tecnologías	Arquitectura	Resultados	Estatus
	dado de la salud; retos futuros.	de datos aplicable en un contexto clínico.			generen futuros conocimientos accionables.	
Malek et al. [38]	La monitorización en tiempo real y el procesamiento de cambios de contexto frecuentes y dinámicos.	Una plataforma holística obtenida de la combinación de herramientas IoT, procesamiento de eventos y tecnologías Big Data.	Kaa, una plataforma para construir, gestionar e integrar dispositivos IoT. Apache Storm, un <i>framework</i> distribuido para procesamiento de flujos, programado en su mayoría	Una plataforma conformada por tres capas: adquisición de datos (Kaa), procesamiento de datos (Apache Storm) y almacenamiento y visualiza-	Un primer prototipo de la aplicación se desarrolló y probó en un contexto de cuidado de la salud a pequeña escala, con el objetivo de mostrar la utilidad de la plataforma holística para desarrollar aplicaciones para un contexto específico.	En desarrollo.

Artículo	Problema	Contribución	Tecnologías	Arquitectura	Resultados	Estatus
			en Clojure. MongoDB, programa de base de datos NoSQL orientado a documentos.	ción (aplicación <i>web</i> ) de datos.		
Marjani et al. [2]	Recolectar y procesar grandes cantidades de información como resultado del desarrollo de IoT.	Una nueva arquitectura que integra IoT con BDA.	RFID ( <i>radio-frequency identification</i> , identificación de frecuencia de radio), Wi-Fi, banda ultra ancha, ZigBee y Bluetooth.	Una capa contenedora de dispositivos sensores interconectados por red inalámbrica, una puerta de enlace	No se mencionan.	Terminado.

Artículo	Problema	Contribución	Tecnologías	Arquitectura	Resultados	Estatus
				IoT y una capa BDA.		
Dubey et al. [39]	La cantidad de datos multi-modales y heterogéneos recolectados a través de sensores.	Un prototipo funcional de una arquitectura orientada a servicios para cómputo en “la niebla”.	Sistema en Chip Intel® Edison que incluye una unidad de procesamiento central Atom™ con doble núcleo a 500 MHz de frecuencia y un microcontrolador Intel® Quark™ a 100 MHz.	Wi-Fi, 3G/4G, Bluetooth2.0. Servicios de cómputo en “la niebla”: <i>Pipeline</i> para procesamiento de señales, extracción de atributos, bases de datos <i>in situ</i> , control de comunica-	Se validó el uso de la arquitectura <i>Fog Data</i> para dos importantes problemas del cuidado de la salud: la dificultad en el habla y los electrocardiogramas. Las soluciones basadas en la arquitectura <i>Fog Data</i> tienen el potencial para reducir requisitos de logística para aplicaciones de tele-salud y requerimientos de almacenamiento en “la nube” y de transmisión	Terminado.

Artículo	Problema	Contribución	Tecnologías	Arquitectura	Resultados	Estatus
			Una memoria LPDDR3 de 1GB y un dispositivo integrado eMMC de almacenamiento <i>flash</i> . Distribución de Linux embebida Ubuntu. Python v2.7.2	Sistemas operativos embebidos. Procesador embebido Intel® Edison.	de corriente en dispositivos al borde de la red.	
Lee et al. [40]	Retos en el diseño de algoritmos y sistemas para analíticas y aplicaciones	Un estudio de los retos, pasos claves de procesamiento y aplicaciones de BDA sobre EHR en el cuidado de la salud.	No se mencionan.	No	A partir del examen de varios tipos de analíticas del cuidado de la salud en EHR, el flujo de analítica de datos es aún la base de la mayoría de aplicaciones en	Terminado.

Artículo	Problema	Contribución	Tecnologías	Arquitectura	Resultados	Estatus
	del cuidado de la salud.				el cuidado de la salud. Sin embargo, es necesario que los algoritmos específicos se ajusten modelando las características únicas de los rasgos médicos.	

*TABLA I*  
*Análisis comparativo de los artículos estudiados*

La literatura estudiada enfatiza el potencial de los fenómenos Big Data e IoT como tecnologías clave para el futuro de las telecomunicaciones y la información en Internet. Los autores abarcan desde problemas generales, como la necesidad de acelerar la adopción de Big Data y los retos que esto conlleva [6], [33]–[35], [37], hasta soluciones específicas enfocadas en el procesamiento y análisis eficiente de información para convertirla en conocimiento accionable en campos como el tele-cuidado [2], [36], [38], [39].

Este proyecto de tesis propone el uso de técnicas de BDA, para el desarrollo de un sistema de tele-cuidado de nueva generación, con capacidad para generar recomendaciones a partir del descubrimiento de hallazgos y la realización de predicciones y detecciones en datos procedentes de dispositivos conectados a IoT, datos que generalmente no se explotan en este tipo de sistemas de tele-salud, aunque sí en sistemas de tele-medicina. La meta es permitir a los principales actores involucrados en el tele-cuidado de adultos mayores – pacientes, cuidadores o médicos – tomar mejores decisiones más rápidamente con base en dichas recomendaciones.

### **2.3 Propuesta de solución**

A continuación, se presenta una descripción general de la solución al problema planteado en el capítulo uno, asimismo se describen los enfoques y tecnologías contempladas en el proceso de selección de la solución.

Considerando la combinación de eficiencia y facilidad de integración que brinda OpenCV con Python, así como la compatibilidad inherente entre Cassandra™ y Spark™, la posibilidad de desarrollar ambas aplicaciones de forma paralela que brinda Angular, y la libertad de decisión propuesta por Scrum, se propone como solución la alternativa (3) descrita en la TABLA II.

No.	Dispositivo ponible	Dispositivo de VPC	Biblioteca para VPC	Lenguaje para VPC	Repositorio de datos	Analítica Big Data	Desarrollo de aplicaciones	Metodología
1	Fitbit	RPi3B+ integrada a cámara de videovigilancia	OpenCV	C++	Apache Cassandra™	Apache Spark™	Spring	Scrum
2	Fitbit	Arduino Uno integrada a cámara de videovigilancia	Scikit-image	Python	Apache HBase	MapReduce	Laravel	XP
3	Fitbit	RPi3B+ integrada a cámara de videovigilancia	OpenCV	Python	Apache Cassandra™	Apache Spark™	Angular/Node.js	Scrum

*TABLA II*  
*Alternativas de solución*

### 2.3.1 Justificación de la solución seleccionada

#### 2.3.1.1 Hardware para monitorizar actividades y adquisición de datos

La placa RPi3B+ posee un procesador con una frecuencia de 1.4 GHz, conexión Gigabit Ethernet o Wi-Fi doble banda a 2.4/5 GHz (100 Mbit/s). Este dispositivo se conecta fácilmente a la cámara que se usará en el proyecto, y soporta la instalación de un SO, en este caso se usará Raspbian, un SO basado en la distribución de Linux Debian, optimizado para el *hardware* de la placa Raspberry Pi. La posibilidad de interactuar con un SO, facilita la instalación de paquetes y *software* en general durante el desarrollo e interacción con la placa.

#### 2.3.1.2 Biblioteca para aplicación de técnicas de VPC

OpenCV es una biblioteca creada fundamentalmente para proyectos de VPC, lo cual la convierte en una herramienta esencial el desarrollo del módulo de recopilación de datos por medio del equipo de videovigilancia. Es completamente compatible con Python y, a pesar de estar escrita en C/C++, brinda *scripts* para generar automáticamente las funciones equivalentes en Python. OpenCV brinda un vasto conjunto de algoritmos clásicos y de última generación sobre VPC y aprendizaje automático. Estos algoritmos permiten, al menos, identificar objetos, clasificar gestos humanos en video y seguir objetos en movimiento. Gracias a su diseño enfocado en aplicaciones en tiempo real, OpenCV presenta la mejor opción para el desarrollo de este proyecto.

#### 2.3.1.3 Lenguaje para utilizar biblioteca de VPC en dispositivos de videovigilancia

Ambos, Python y C++, presentan una opción deseable cuando se pretende interactuar con OpenCV en tareas de VPC. Para este proyecto específico se decidió adoptar el primero para estas funciones, tomando en consideración la facilidad que brinda Python a través de bibliotecas que apoyan directamente los procesos de VPC, tales como NumPy, SciPy y matplotlib, por mencionar algunas.

#### 2.3.1.4 Repositorio de datos

La arquitectura de Cassandra™, basada en una arquitectura sin nodo “maestro”, brinda alta resistencia a fallos, ya que no posee un punto de fallo principal, a diferencia de HBase. Dicha arquitectura también soporta ambos, gestión y almacenamiento de datos, mientras que HBase se apoya en HDFS para el segundo.

#### 2.3.1.5 Herramientas para el procesamiento y analítica Big Data

Apache Spark™ es un sistema de cómputo en *clusters* agnóstico a la capa de almacenamiento subyacente, lo cual significa que es utilizable tanto en un Sistema Gestor de Base de Datos (SGBD), relacional o no, como sobre un sistema de archivos. Su abstracción principal (RDD) y el modelo de almacenamiento de Cassandra™ son inherentemente compatibles: por un lado, el RDD en Spark™ es distribuido en su *cluster* (y potencialmente en el almacén de datos utilizado) y replicado en sus nodos para lograr tolerancia a fallos; Cassandra™, por su parte, posee un modelo con particiones distribuidas en sus nodos y replicadas para lograr también la tolerancia a fallos [41].

#### 2.3.1.6 Marcos de trabajo para el desarrollo de la aplicación web

Con una arquitectura basada en componentes, Angular asegura un código reusable y desacoplado, lo cual facilita las pruebas y el mantenimiento. El uso de Angular como principal herramienta de desarrollo para las aplicaciones *web* y móvil asegura como resultado exactamente el mismo código base (JavaScript) para ambas aplicaciones. Al integrar Angular con Node.js como entorno de ejecución y desarrollo, significa que se usará el mismo lenguaje tanto en el cliente como el servidor. El modelo dirigido a eventos de Node.js lo convierten en una alternativa asíncrona ligera y eficiente. El modelo de Node.js presenta dos características principales: un ciclo de eventos y un único hilo de ejecución. Dicho ciclo “escucha” constantemente al puerto designado y envía cada petición a ser procesada por “trabajadores” de trasfondo, quedando listo para recibir las peticiones siguientes. Esto representa la principal diferencia entre Node.js y los servidores basados en tecnologías como IIS (*Internet Information*

*Services*, Servicios de Información de Internet) o Apache: por cada petición, Node.js no crea un nuevo hilo de ejecución (de ahí: hilo único de ejecución), sino que aprovecha la naturaleza asíncrona y sin bloqueo de sus APIs de operaciones de escritura/lectura.

#### 2.3.1.7 Metodologías de desarrollo

Se selecciona Scrum como metodología de desarrollo teniendo en cuenta su entorno ágil, el tamaño del equipo y el tiempo de desarrollo disponible. Esta metodología establece una línea de trabajo a seguir, al mismo tiempo que brinda libertad de toma de decisiones al equipo de desarrollo ante situaciones inesperadas, como el cambio o surgimiento de nuevos requisitos.

## Capítulo 3 Aplicación de la Metodología

Durante el presente capítulo se describen las partes más relevantes del diseño y desarrollo del proyecto de tesis, aplicando la metodología propuesta en la sección anterior, de igual forma se presenta la arquitectura de la aplicación, sus capas, componentes y flujo de trabajo.

### 3.1 Sprint 0

Este primer *sprint* no aporta ningún valor de negocio, es decir, este paso no tiene como objetivo producir una o alguna parte del producto “tangible” y funcional. Al contrario, busca plasmar y cimentar el análisis que sustenta el producto de software esperado. Se inicia con actividades básicas pero fundamentales, tales como: análisis de requerimientos, diseños previos, modelo conceptual, entre otros.

En esta sección tanto el diseño como la arquitectura creada se realizan de manera minimalista dando cabida a cambios durante el tiempo de vida del desarrollo del producto de software.

La arquitectura tecnológica diseñada permite la integración de los dos módulos desarrollados y la aplicación *web*. Como fase inicial de dicho diseño y sirviendo de base a la arquitectura, se elaboró un concepto de solución, el cual se ilustra en la Figura 1. Dicho concepto muestra, a gran escala, los elementos que forman parte del sistema; la recopilación de datos a través del DVP y el dispositivo de videovigilancia, el almacenamiento de datos en un repositorio, el procesamiento y analítica en conjunto con un modelo predictivo para detectar patrones de comportamiento, y los canales de presentación de datos al usuario final de las aplicaciones.

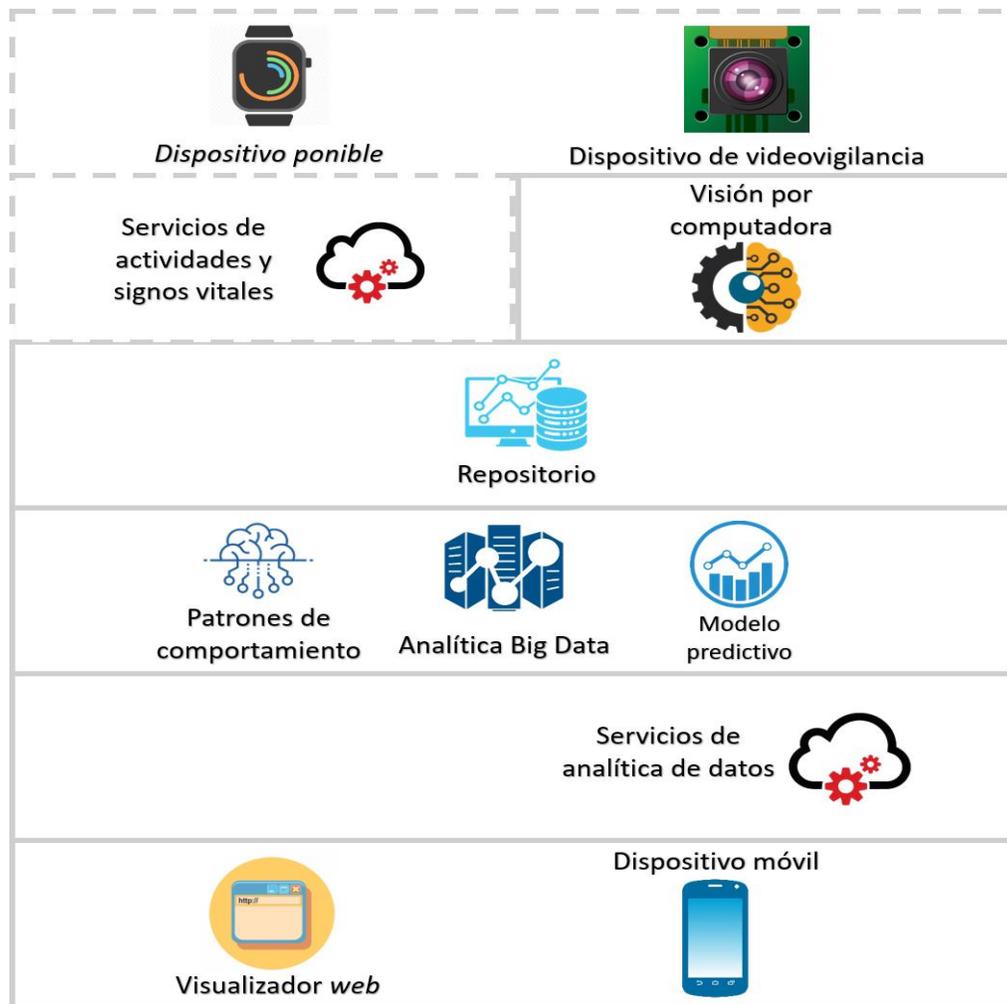


Figura 1. Diseño conceptual de la solución

### 3.1.1 Arquitectura de la solución

Se diseñó una arquitectura en capas, agrupando de manera lógica sus componentes. La Figura 2 muestra dicha arquitectura, sus capas y componentes, los cuales se explican en detalle a continuación.

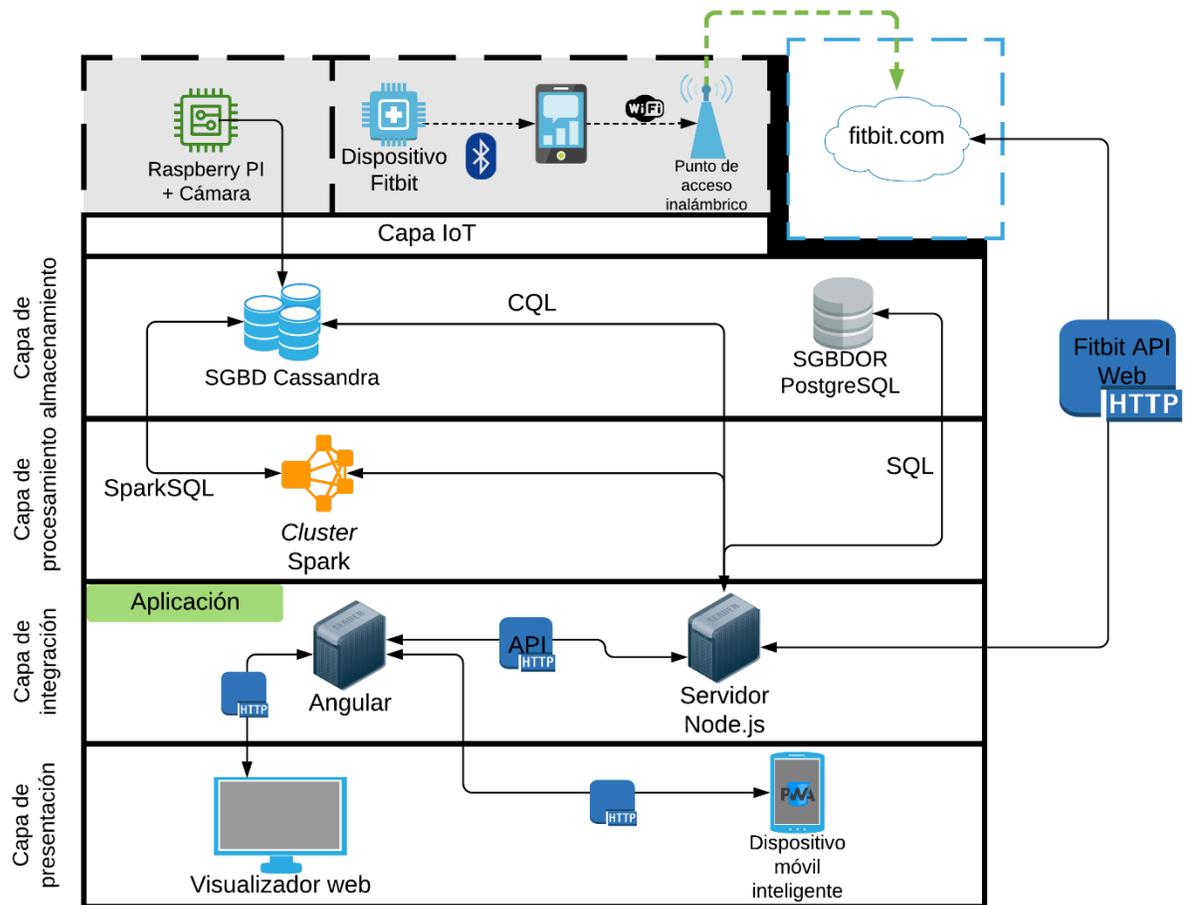


Figura 2. Arquitectura del sistema

Con respecto a la arquitectura del sistema, se considera que tanto el DVP como la cámara integrada a la RPi3B+ conforman elementos que no se desarrollaron o fabricaron como parte del presente proyecto, pero se incluyen en la propuesta de arquitectura ya que representan los orígenes de datos que “alimentan” los procesos de analítica. Cabe mencionar también que la transmisión de datos (su frecuencia y volumen) desde el DVP hacia el dominio del sitio del fabricante ocurre fuera del control de nuestro sistema, y que la única forma que existe en este momento de acceder a los mismos es a través de la API que brinda dicho dominio.

#### 3.1.1.1 Capa IoT

Esta capa contiene las fuentes de datos, a saber, un DVP y un dispositivo de videovigilancia integrado a una placa RPi3B+. Los datos de la primera fuente se obtienen a través de la API Web provista por el dominio. Los datos provenientes de la placa RPi3B+ se transmiten hacia nuestro servidor de aplicación y de ahí se almacenan en nuestra base de datos no relacional.

**DVP.** Permite la monitorización de parámetros como la frecuencia cardíaca, los pasos dados, los pisos subidos y bajados, y las horas transcurridas en cada estado de sueño. Estos parámetros brindan los datos necesarios para inferir el estado de salud físico del usuario y su nivel de cansancio corporal, elementos que se utilizan en los procesos de prevención de posibles accidentes. Durante el desarrollo del sistema se utilizó un dispositivo perteneciente al fabricante Fitbit, Inc.

**Dispositivo de videovigilancia integrado a una placa RPi3B+.** Se enfoca en la monitorización de actividades del usuario, así como del análisis de su entorno, específicamente los objetos y/o desniveles que lo conforman y representen posibles causas de caídas. En esta placa se ejecuta el *software* de VPC para detectar las posiciones de objetos y del usuario en la habitación.

#### 3.1.1.2 Capa de almacenamiento

Esta capa contiene los SGBD utilizados en el sistema: Apache Cassandra™ y PostgreSQL.

**SGBD no relacional Apache Cassandra™.** Almacena los datos procedentes del DVP y del dispositivo de videovigilancia de acuerdo al modelo definido en la sección 3.1.4. Estos datos conforman la materia prima con que se realizan los procesos de analítica Big Data en el *cluster* de Apache Spark™.

**Sistema Gestor de Base de Datos Objeto Relacional PostgreSQL.** Contiene la información relacionada con los usuarios del sistema y sus permisos, las notificaciones generadas, y los datos agregados para mostrar en reportes.

#### 3.1.1.3 Capa de procesamiento

**Apache Spark™.** Es el motor analítico que se utiliza en el proyecto para el procesamiento de datos a gran escala, para la detección de anomalías en las actividades y parámetros. A causa de la forma en que Spark™ distribuye los datos para su analítica en el *cluster*, de acuerdo a su RDD, y la forma en que Cassandra™ almacena los datos siguiendo su modelo distribuido, estas dos herramientas son altamente compatibles, por lo que son perfectas para trabajar en conjunto.

#### 3.1.1.4 Capa de integración

**Servidor de aplicación Node.js.** Nuestro servidor de aplicaciones se desarrolló sobre el entorno de ejecución de JavaScript, Node.js. En el mismo se implementó una API que provee el acceso a datos en ambos SGBD, la cual se consume desde el *front-end* de la aplicación. También se implementó la lógica de peticiones HTTP a la API Web que brinda el dominio del proveedor de DVP, cuyos datos se almacenan periódicamente en Cassandra™.

**Angular.** Para el desarrollo de la aplicación *web* se utilizó el marco de trabajo Angular en su versión 8. Este componente contiene la estructura de las interfaces de usuario, así como los puntos de comunicación con la API del *backend* en Node.js.

#### 3.1.1.5 Capa de presentación

Los usuarios finales interactúan con el sistema a través de la aplicación *web* desarrollada en Angular.

### 3.1.2 Flujo de trabajo

Las relaciones entre los componentes de la arquitectura mostrada en la Figura 2 definen el flujo de trabajo para el proceso de análisis de datos, desde la obtención de los mismos hasta su presentación en forma de alertas o reportes generados por el usuario de la aplicación. A continuación, se describe dicho flujo.

➤ El DVP transmite de forma periódica los parámetros fisiológicos del adulto mayor hacia el dominio del proveedor de DVP. Desde ahí, el servidor de aplicación

consume estos datos a través de la API Web que brinda dicho dominio, y los almacena en Cassandra™ utilizando consultas CQL (*Cassandra Query Language*, Lenguaje de Consultas de Cassandra). Paralelamente, a partir de las imágenes de video procesadas en la RPiB3+, se monitoriza la posición del adulto mayor con respecto a los objetos a su alrededor, y se detecta y registra cualquier situación de peligro potencial, notificando al cuidador. Tanto los datos del DVP como los de la placa RPi3B+ se almacenan siguiendo un patrón de series de tiempo.

➤ Los datos almacenados en Cassandra™ se utilizan en el RDD de Spark™ para realizar la analítica de detección de patrones y determinar comportamientos anómalos en los parámetros fisiológicos y movimientos del adulto mayor en el local en que se encuentra el módulo de VPC, de acuerdo al modelo predictivo diseñado, con el objetivo de enviar alertas a los usuarios de las aplicaciones para evitar un posible accidente o reaccionar de manera expedita ante un accidente ya ocurrido.

➤ Una vez que se envía una alerta al servidor, este notifica al usuario encargado del cuidado del adulto mayor en cuestión, a través del menú de notificaciones de las aplicaciones y/o un correo electrónico, a la vez que se almacena en PostgreSQL para futuras consultas. En este momento, el usuario actúa en consecuencia ante la posible situación de peligro, verificando su veracidad. Posteriormente, una vez que se determina si el incidente detectado representaba en realidad una situación de peligro o fue un falso positivo, el usuario refleja dicha información en Cassandra™ y el mismo pasa a formar parte del conjunto de datos de entrenamiento disponibles a Spark™ para sus procesos de analítica. Finalmente, el usuario tiene la posibilidad de generar un reporte del incidente el cual se almacena en un historial en PostgreSQL.

El diagrama de paquetes, en la Figura 3, muestra a gran escala, en notación UML, las dependencias entre los componentes del sistema.

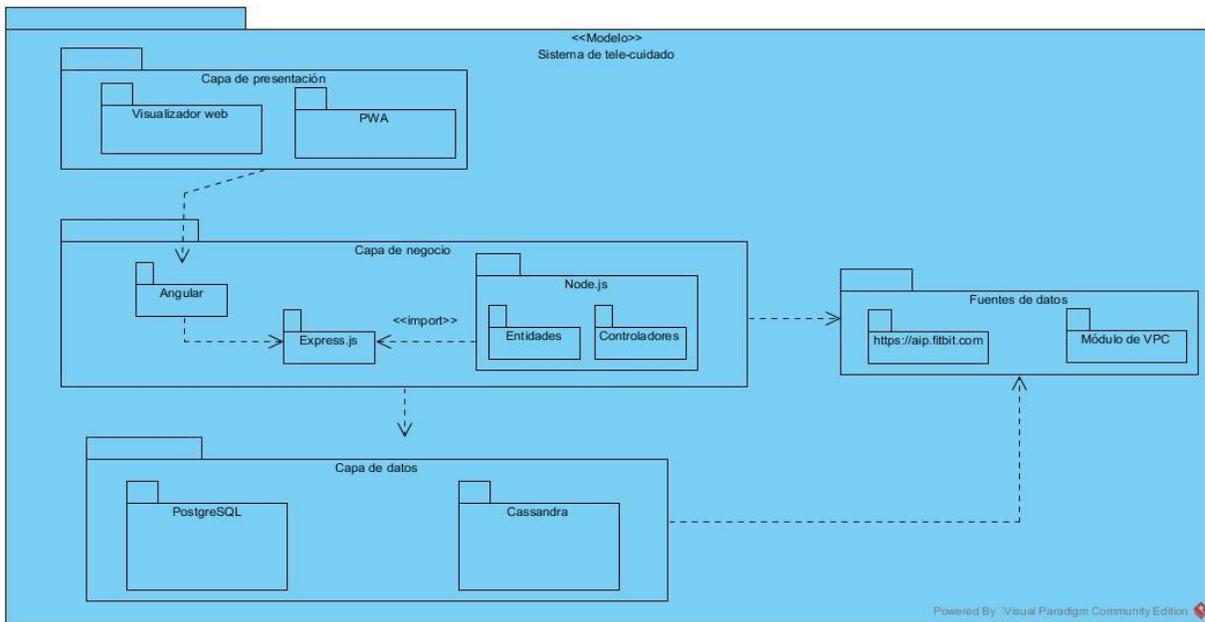


Figura 3. Diagrama de paquetes del sistema

### 3.1.3 Especificación de requisitos de la aplicación

A continuación, se describen cada uno de los requisitos que cumplirá la aplicación, posteriormente se representan dichos requerimientos en un diagrama de casos de uso para su mejor comprensión.

#### 3.1.3.1 Identificación de requerimientos

El sistema es capaz de recopilar datos procedentes del DVP, y el dispositivo doméstico de videovigilancia para monitorizar signos vitales y actividades de adultos mayores. Asimismo, el sistema detecta anomalías en signos vitales y actividades de adultos mayores, para la generación de alertas basadas en Analítica Big Data. A través de las aplicaciones, los usuarios recibirán notificaciones vía correo electrónico, o a través del panel de notificaciones; el sistema también tendrá capacidades de generación de reportes y gráficas con los datos históricos del adulto mayor, en diversos formatos.

El sistema no presenta un proceso de registro de usuarios libre, por lo que es necesario crear un administrador inicialmente, teniendo este los permisos para registrar nuevos usuarios. Cada usuario inicia la interacción con la aplicación ingresando su nombre de

usuario y contraseña. Una vez autenticado y autorizado, dependiendo del tipo de usuario, se permiten realizar diferentes acciones.

- **Administrador:** Tiene acceso al panel de usuarios y adultos mayores, desde donde gestiona su información, permite o cancela la monitorización de un adulto mayor por un usuario (vincular/desvincular), y renueva o revoca los *tokens* de acceso de cada cuenta asociada a un adulto mayor.
- **Cuidador:** Tiene acceso al panel de notificaciones y alertas, así como la visualización y generación de reportes históricos y gráficas con los datos de los adultos mayores que están a su cuidado (vinculados a su cuenta de usuario).

### 3.1.4 Modelos de bases de datos

#### Diagrama entidad-relación

La Figura 4 ilustra las entidades que conforman el sistema, sus atributos y las relaciones entre ellas.

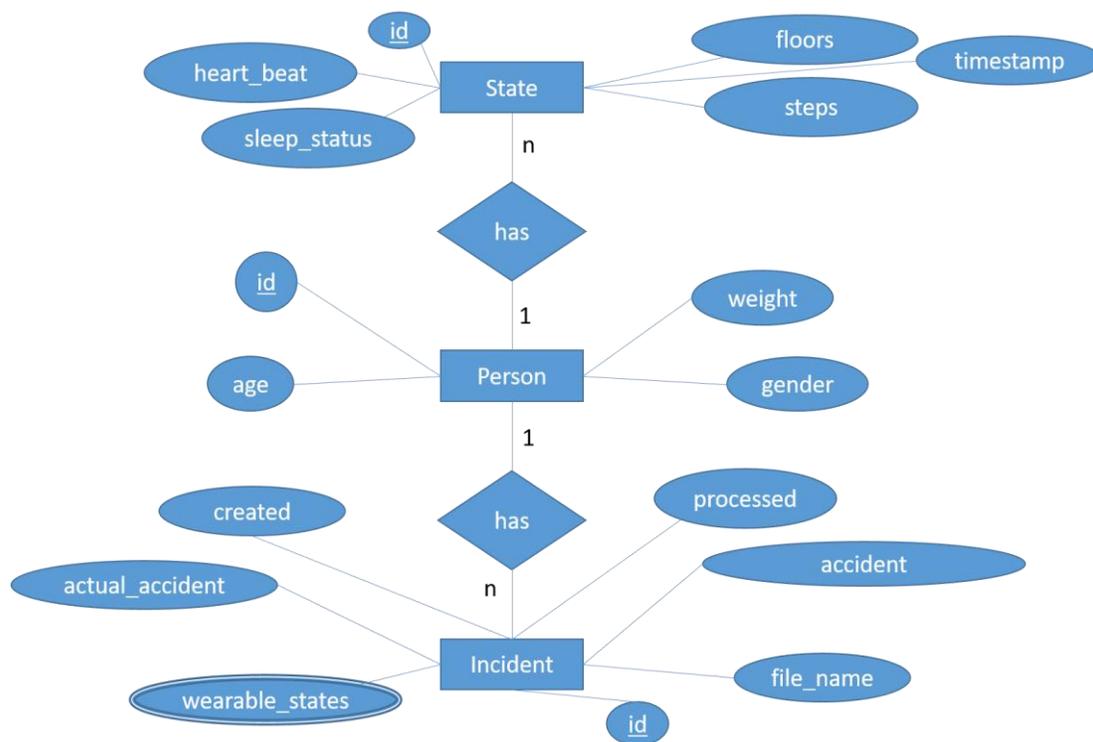


Figura 4. Diagrama entidad-relación

### 3.1.4.1 Modelo de base de datos objeto-relacional.

La Figura 5 muestra el modelo de las relaciones implementadas en PostgreSQL para el almacenamiento y acceso de la información relacionada con los usuarios del sistema y sus permisos, las notificaciones generadas, y los datos agregados para mostrar en reportes.

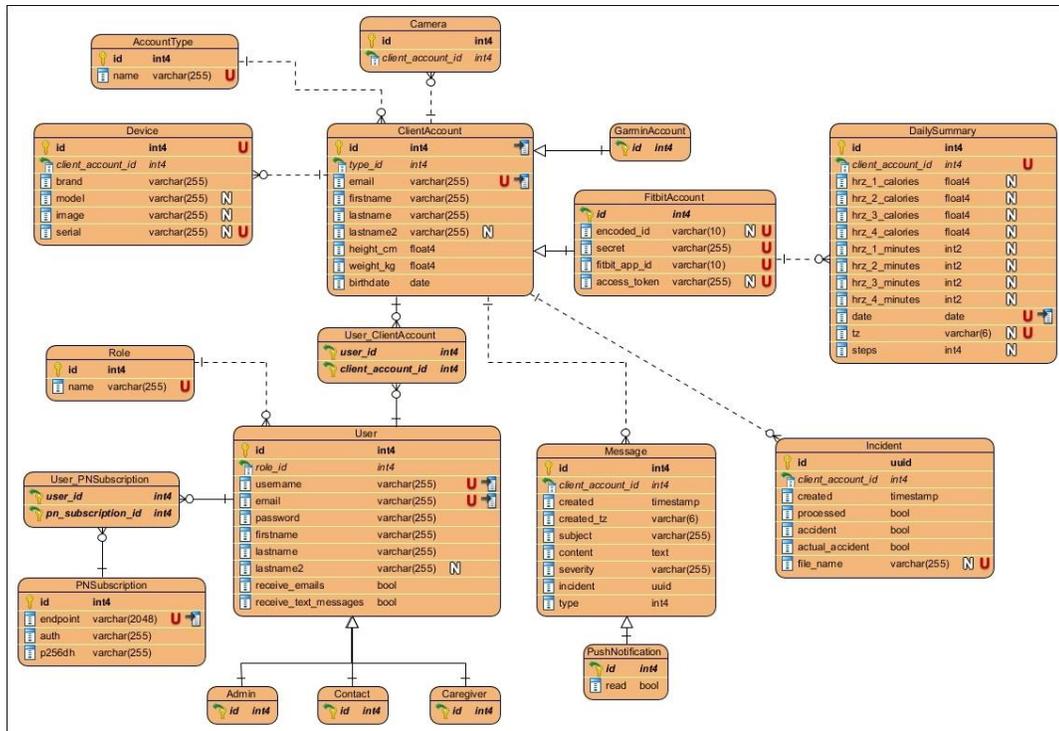


Figura 5. Modelo de base de datos objeto-relacional

### 3.1.4.2 Modelo de base de datos no relacional.

El modelado Big Data es significativamente diferente a aquel utilizado en los modelos relacionales; mientras que el segundo se enfoca en las entidades que actúan sobre el sistema, y sus relaciones, el primero se basa en las consultas sobre los datos, que se desean satisfacer, a veces sacrificando consistencia e introduciendo redundancia en dichos datos, para obtener mayor velocidad de acceso y escritura. A continuación, se muestran los diagramas lógico y físico, del modelo implementado en Cassandra™, así como las sentencias CQL obtenidas, para almacenar los datos procedentes del DVP y del dispositivo de videovigilancia.

### Diagrama lógico

Dado que el modelado para Cassandra™ es un modelado orientado a consultas, se consideran las consultas que soporta el sistema durante la confección del diagrama lógico, el cual se ilustra en la Figura 6 en formato Chebotko<sup>1</sup>.

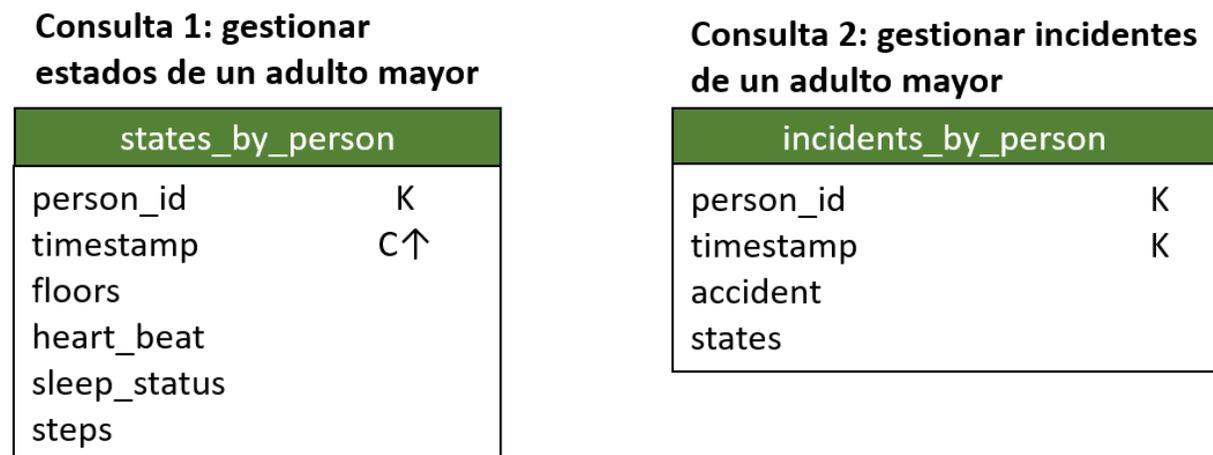


Figura 6. Diagrama lógico

En el formato de diagramas Chebotko se establecen los nombres de tablas, sus columnas, las columnas que corresponden a la llave de partición (K) y las columnas de agrupamiento (C), así como la dirección del agrupamiento: ascendente (↑) o descendente (↓). Las columnas denotadas con K y C conforman, en su conjunto, la llave primaria de la tabla.

### Diagrama físico

En el modelado para Cassandra™, durante la transición del diagrama lógico al físico, se realizan dos cambios principales: La asignación de tipos de datos y las optimizaciones que se estimen necesarias. La Figura 7 muestra el diagrama físico de la base de datos no relacional.

<sup>1</sup> El Diagrama Chebotko es una técnica de visualización novedosa que presenta un diseño de esquema de base de datos, como una combinación de esquemas individuales de tablas y transiciones de flujo de trabajo de aplicaciones basadas en consultas [17]

**Consulta 1: gestionar estados de un adulto mayor**

states_by_person_by_week		
person_id	int	K
week	timestamp	K
second	int	C↓
floors	int	
heart_beat	int	
sleep_status	float	
steps	int	

**Consulta 2: gestionar incidentes de un adulto mayor**

incidents_by_person		
person_id	int	K
id	uuid	K
accident	boolean	
[states]	wearable_state	

*wearable_state*	
hbpm	int
sleep_status	float
steps	int
floors	int

Figura 7. Diagrama físico

En el diagrama lógico, la tabla **states\_by\_person** posee un problema potencial referente al tamaño de sus particiones, ya que, en ese estado, tiene la capacidad de almacenar más de 31 millones de valores cada una, con lo cual, en teoría, Cassandra™ opera sin inconvenientes, pero en la práctica no es recomendable. Para mitigar esta situación, se decide “dividir” la columna **timestamp**, generando las columnas **week** y **second**. La columna **week** pasa a ser parte de la llave de partición, lo cual disminuye a un máximo de 604800 valores su límite (cantidad de segundos en una semana), ya que al inicio de cada semana se creará una nueva partición, lo cual es mucho más manejable en términos de tamaño de partición. La columna **second** pasa a ser de agrupamiento (descendente), lo cual identifica inequívocamente cada tupla en la tabla y las ordena desde la más reciente a la más antigua, dentro de cada semana.

Finalmente, y como consecuencia de esta optimización, la tabla se renombró **states\_by\_person\_by\_week**.

En la tabla **incidents\_by\_person** no se realizó ninguna optimización, solamente se necesario notar el uso de un Tipo de Dato Definido por el Usuario (UDT por sus siglas en inglés), el cual se identifica usando la notación **\*wearable\_state\***.

### *Sentencias CQL*

Para culminar la implementación del modelo de la base de datos no relacional en Cassandra™, se definió el espacio de claves *masters*, que contendrá las definiciones de tipos y tablas utilizadas por nuestro sistema. En el Listado 1 se muestran las sentencias CQL utilizadas.

```
CREATE TYPE wearable_state (
    hbpm          INT,
    sleep_status  FLOAT,
    steps         INT,
    floors        INT,
);
CREATE TABLE incidents_by_person (
    person_id     INT,
    id            UUID,
    accident      BOOLEAN,
    wearable_states LIST<FROZEN <WEARABLE_STATE>>,
    PRIMARY KEY ((person_id, id))
);
CREATE TABLE wearable_states_by_person_by_week (
    person_id     INT,
    week          TIMESTAMP,
    second        INT,
    floors        INT,
    heart_beat    INT,
    sleep_status  FLOAT,
    steps         INT,
    PRIMARY KEY ((person_id, week), second)
) WITH CLUSTERING ORDER BY (second DESC);
```

Listado 1. Sentencias CQL utilizadas en la creación del esquema de base de datos para Cassandra™

## 3.2 Sprint 1

### 3.2.1 Registro de la aplicación en el dominio <https://dev.fitbit.com>

Durante el desarrollo y pruebas, se utilizó un DVP marca Fitbit, modelo Blaze, el cual se usa alrededor de la muñeca como un reloj. Este dispositivo detecta constantemente datos referentes al estado fisiológico del usuario – ritmo cardíaco, calorías quemadas – así como los relacionados con su actividad física – pasos recorridos, pisos subidos, horas dormidas – y transmite dichos datos hacia un dispositivo móvil o celular, en el cual previamente se instaló la aplicación Fitbit (disponible de forma gratuita en la *Play Store* y *App Store* para dispositivos con SO Android e iOS respectivamente), y vinculado a través de una conexión Bluetooth con el DVP. Finalmente, el celular reenvía de manera periódica en intervalos de 15 minutos (y cada vez que el usuario abra la aplicación Fitbit) los datos hacia los servidores de Fitbit a través su red de datos móviles o conexión Wi-Fi disponible.

Una vez los datos se encuentran en los servidores de Fitbit, se consumen a través de su API Web, por las aplicaciones registradas en el dominio <https://dev.fitbit.com> que poseen los permisos requeridos. Entre los datos necesarios para el registro de la aplicación, se encuentra la dirección de dominio de la misma (<http://localhost.com> por el momento) y una URL (*Uniform Resource Locator*, Localizador de Recurso Uniforme) de *retorno* (<http://localhost.com/admin>) hacia la cual se redirige el usuario administrador una vez culmine su autenticación en el flujo OAuth<sup>1</sup> provisto por Fitbit. Como parte del registro de la aplicación, se generan automáticamente un identificador de cliente y un *secreto*, ambos en forma de cadena de caracteres, que se utilizan en la generación, renovación y revocación de *tokens* de acceso, así como durante las peticiones a los recursos de la API.

---

<sup>1</sup> OAuth 2.0 es el protocolo de autenticación estándar para la industria.

Durante la instalación y configuración de la placa Raspberry Pi 3B+, se utilizó como dispositivo de almacenamiento una tarjeta de memoria micro-SD, donde se instaló el SO Raspbian<sup>1</sup>, utilizando la herramienta BalenaEtcher.

### 3.3 Sprint 2

Hasta el momento se consideran dos tipos de usuarios para el uso del sistema, Administrador y Cuidador. Solamente los administradores de la aplicación están autorizados para crear nuevos usuarios, registrar nuevas cuentas de adultos mayores, así como generar, renovar y revocar *tokens* de acceso. Los cuidadores, por su parte, se encargan de la monitorización y seguimiento del estado de adultos mayores, haciendo uso del panel de notificaciones y la vista de gráficas y reportes agregados.

#### 3.3.1 Autenticación de usuarios

El proceso de autenticación se basa en el uso de JWTs (*JSON Web Token*). En la Figura 8 se muestra el diagrama de secuencia correspondiente.

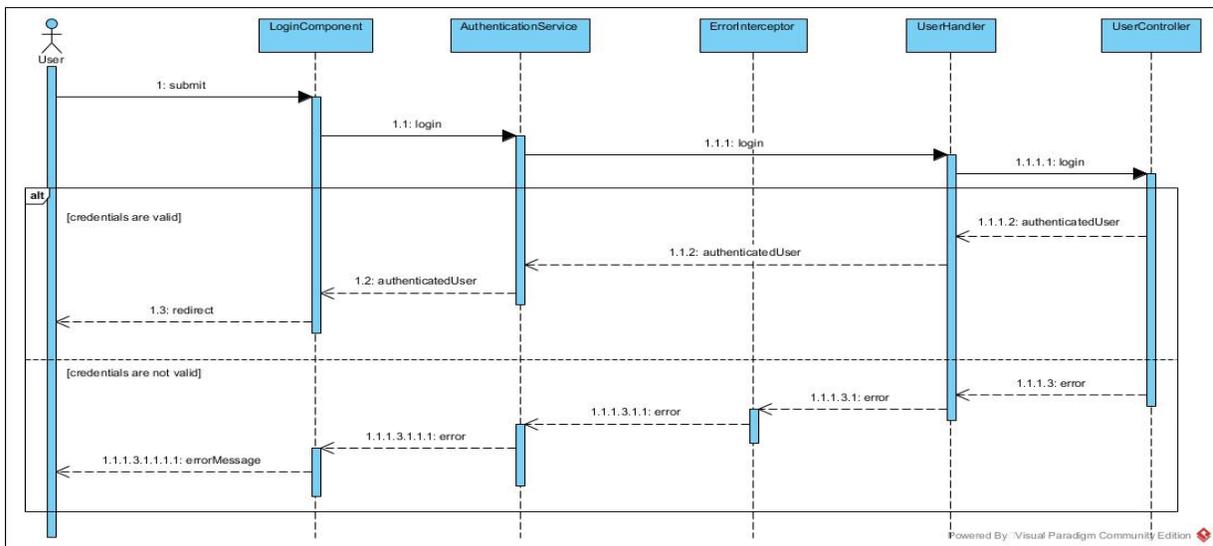


Figura 8. Autenticación – diagrama de secuencia

<sup>1</sup> Raspbian es un SO basado en Debian – distribución de Linux – de libre adquisición y uso, optimizado para el *hardware* de la Raspberry Pi. Raspbian es un proyecto de la comunidad bajo desarrollo activo y contiene más de 35 mil paquetes de *software* pre-compilado [42].

### 3.3.2 Gestión de usuarios

En adición a los procesos de altas, bajas, modificaciones y consultas, y como parte de la gestión de usuarios, los administradores de la aplicación brindan (o deniegan) a los cuidadores, acceso a las notificaciones y datos de varios adultos mayores; este proceso se define como **Vinculación (o desvinculación) de cuenta de adulto mayor**, y se ilustra en la Figura 9.

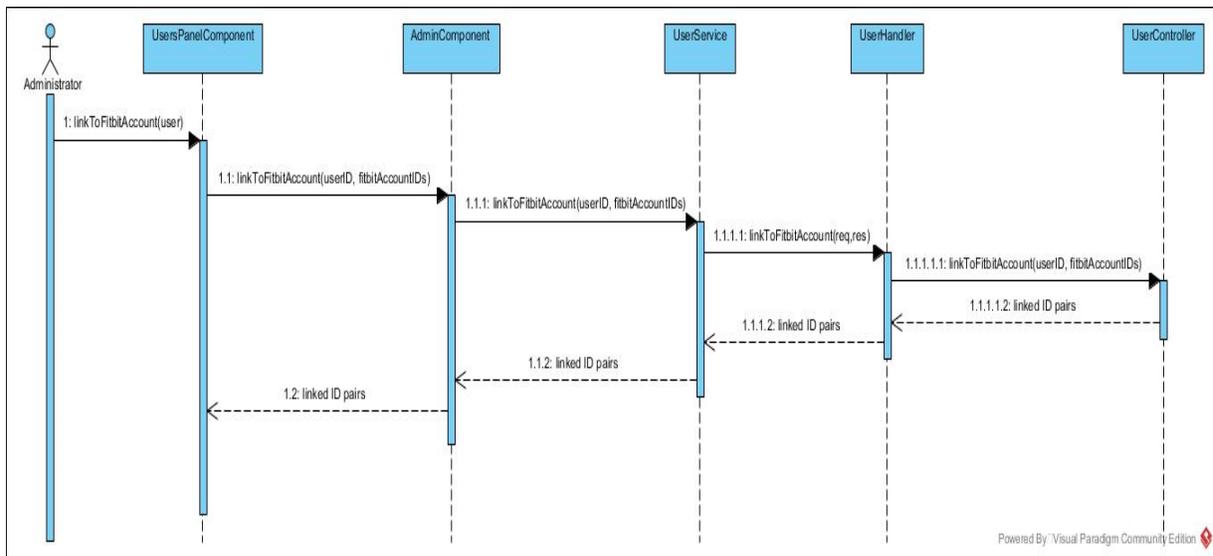


Figura 9. Vinculación de cuenta de adulto mayor – diagrama de secuencia

### 3.3.3 Gestión de cuentas de adultos mayores

De manera análoga, en adición a los procesos de crear, editar, eliminar y consultar cuentas de adultos mayores en la aplicación, se permite generar, renovar y revocar los *tokens* de acceso a dichas cuentas. Esto último también lo realizan los administradores, y es la única interacción directa entre algún usuario de la aplicación y los recursos que brinda la API Web de Fitbit, porque los flujos OAuth que implementa el dominio <https://www.fitbit.com> para la gestión de dichos *tokens*, requieren la inserción manual de credenciales, por lo que no es posible su total automatización.

Para solicitar un *token* de acceso, el administrador selecciona una cuenta de adulto mayor en el panel correspondiente, y hace clic sobre la opción *Renovar token de acceso* – Figura 10.

## Cuentas



Figura 10. Renovar token de acceso

Esta acción re-direcciona al usuario a la interfaz para autenticación por OAuth del dominio <https://www.fitbit.com> donde se introducen los datos del usuario y su contraseña – Figura 11.



Figura 11. Interfaz del dominio <https://www.fitbit.com> para autenticación por OAuth

Seguidamente, el administrador asigna los permisos necesarios – pasos, ritmo cardíaco, estatus de sueño, pisos, entre otros –Figura 12.



Proto de none solicita permiso para acceder a los siguientes datos en tu cuenta de Fitbit. durante 30 días

¡Advertencia! Esta aplicación no utiliza HTTPS para obtener tu permiso de forma segura.

- Permitir todo
  - actividad y ejercicio
  - peso
  - Dispositivos y configuraciones de Fitbit
  - perfil
  - sueño
  - ritmo cardíaco

Si solo permites algunos de estos datos, puede que Proto no funcione según lo previsto. Obtén más información sobre estos permisos [aquí](#).

Negar
Permitir

Los datos que compartas con Proto se registrarán por la [Política de privacidad](#) y las [Términos de servicio](#) de none. Podrás revocar dicho consentimiento en cualquier momento en tu [configuración de la cuenta](#) de Fitbit.

Sesión iniciada como kazy28@hotmail.com  
¿No eres tú?

Figura 12. Interfaz de asignación de permisos

Finalmente, el flujo re-direcciona al administrador de vuelta hacia la aplicación a través de la URL de retorno. A partir de este punto, el sistema extrae los datos de acceso de dicha URL y los vincula a la cuenta de adulto mayor seleccionada inicialmente; estos se utilizan para realizar peticiones a la API Web de Fitbit. La Figura 13 ilustra la secuencia de solicitud de *token* de acceso para una cuenta de adulto mayor.

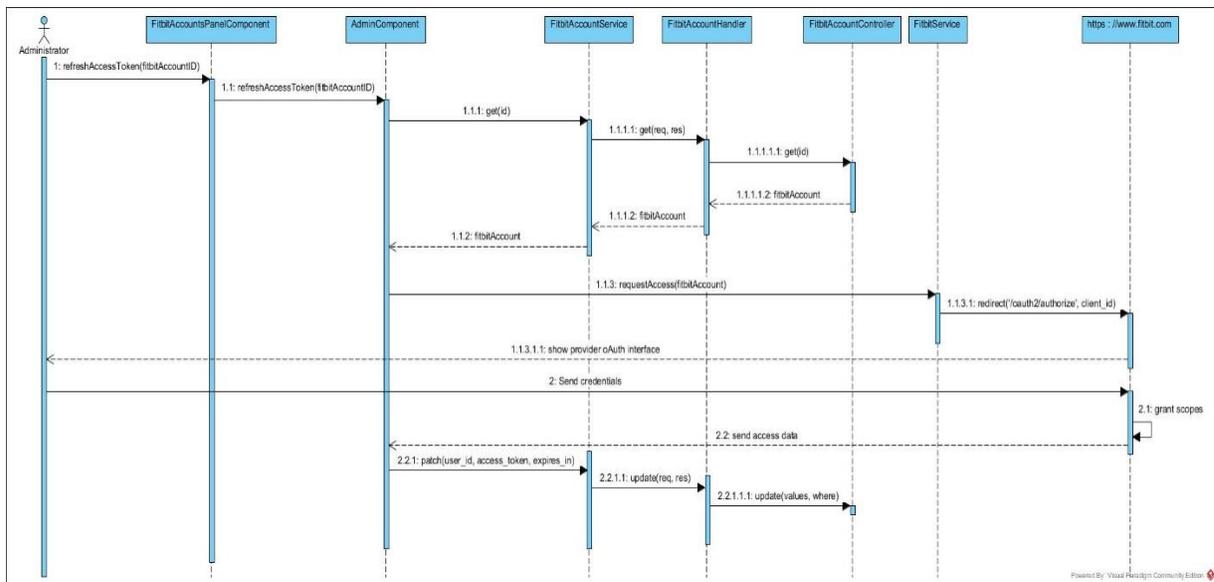


Figura 13. Solicitud de token de acceso – diagrama de secuencia

## 3.4 Sprint 3

### 3.4.1 Módulo de recopilación de datos provenientes del DVP

El proceso de recopilación de datos provenientes del DVP se realiza automática y periódicamente en el servidor, específicamente en el archivo *DataFetcher.js*, donde se coordina la persistencia de dichos datos en las bases de datos en Cassandra™ y PostgreSQL según corresponda.

Para la solicitud de datos a Fitbit, se utiliza el módulo *fitbit-node*<sup>1</sup> – Listado 2 – desarrollado para Node.js, el cual se instancia haciendo uso del secreto e identificador de cliente provistos durante el registro de la aplicación en el dominio Fitbit.

```

1. import FitbitApiClient from "fitbit-node";
2. const fitbitApiClient = new FitbitApiClient({
3.     clientId: fitbit_app_id,
4.     clientSecret: secret,
5.     apiVersion: "1.2",
6. });

```

*Listado 2. Instancia del módulo *fitbit-node* utilizando los metadatos de la aplicación*

Se realizan una serie de peticiones a la API Web de Fitbit a intervalos de diez minutos, para obtener el perfil de usuario que contiene el peso y edad actuales del adulto mayor, así como los datos de frecuencia cardíaca, pasos, pisos, y el estatus de sueño de cada cuenta de adulto mayor registrada en la aplicación que corresponden al tiempo transcurrido desde la última vez que se obtuvieron datos. Por otra parte, también se obtienen datos referentes a las zonas cardíacas por las que transita el usuario y las calorías quemadas durante el día. Estos últimos se almacenan en la base de datos relacional, en PostgreSQL. Los listados 3, 4, 5, 6, 7 y 8 muestran ejemplos de las peticiones al dominio <https://api.fitbit.com> mencionadas anteriormente.

<sup>1</sup> <https://github.com/lukasolson/fitbit-node>

```

1. const profile = await fitbitApiClient.get("/profile.json",
2.   access_token, encoded_id);

```

*Listado 3. Obtener perfil de usuario*

```

1. const getHeartRateDataSet = async (startMoment, endMoment,
2.   fitbitAccount, fitbitApiClient) => {
3.
4.   //prepare API endpoint
5.   let heartRateEP = `/activities/heart/date/
6.     ${startMoment.format("YYYY-MM-DD")}/
7.     ${endMoment.format("YYYY-MM-DD")}/1sec/time/
8.     ${startMoment.format("HH:mm")}/
9.     ${endMoment.format("HH:mm")}.json`;
10.
11.  //fetch heart rate
12.  const heartRateResponse = await fitbitApiClient.get(
13.    heartRateEP,
14.    fitbitAccount.access_token,
15.    fitbitAccount.encoded_id
16.  );
17.
18.  if (verifyResponse(heartRateResponse, fitbitAccount)) {
19.    const heartRateIntraday =
20.      heartRateResponse[0]["activities-heart-intraday"];
21.    if (heartRateIntraday) {
22.      const heartRateIntradayDataSet = heartRateIntraday.dataset;
23.      if (heartRateIntradayDataSet &&
24.        heartRateIntradayDataSet.length)
25.        return heartRateIntradayDataSet;
26.    }
27.    return [];
28.  }
29.  };
30.  //...
31.  const heartRateDataSet = await getHeartRateDataSet(startMoment,
32.    endMoment, fitbitAccount, fitbitApiClient);

```

*Listado 4. Obtener datos de ritmo cardíaco*

```

1. const getStepsDataSet = async (startMoment, endMoment, fitbitAccount,
2.   fitbitApiClient) => {
3.   const stepsEP = `/activities/steps/date/

```

```

4.     ${startMoment.format("YYYY-MM-DD")}/
5.     ${endMoment.format("YYYY-MM-DD")}/1min/time/
6.     ${startMoment.format("HH:mm")}/
7.     ${endMoment.format("HH:mm")}.json`;
8.
9.     const stepsResponse = await fitbitApiClient.get(stepsEP,
10.      fitbitAccount.access_token, fitbitAccount.encoded_id);
11.
12.     if (verifyResponse(stepsResponse, fitbitAccount)) {
13.       const stepsIntraday =
14.         stepsResponse[0]["activities-steps-intraday"];
15.       if (stepsIntraday) {
16.         const stepsIntradayDataSet = stepsIntraday.dataset;
17.         if (stepsIntradayDataSet && stepsIntradayDataSet.length)
18.           return stepsIntradayDataSet;
19.       }
20.       return [];
21.     }
22.   };
23.   //...
24.   const stepsDataSet = await getStepsDataSet(startMoment, endMoment,
25.     fitbitAccount, fitbitApiClient);

```

*Listado 5. Obtener cantidad de pasos*

```

1.   const getFloorsDataSet = async (startMoment, endMoment, fitbitAccount,
2.     fitbitApiClient) => {
3.     const floorsEP = `/activities/floors/date/
4.       ${startMoment.format("YYYY-MM-DD")}/
5.       ${endMoment.format("YYYY-MM-DD")}/1min/time/
6.       ${startMoment.format("HH:mm")}/
7.       ${endMoment.format("HH:mm")}.json`;
8.
9.     const floorsResponse = await fitbitApiClient.get(floorsEP,
10.      fitbitAccount.access_token, fitbitAccount.encoded_id);
11.
12.     if (verifyResponse(floorsResponse, fitbitAccount)) {
13.       const floorsIntraday =
14.         floorsResponse[0]["activities-floors-intraday"];
15.       if (floorsIntraday) {
16.         const floorsIntradayDataSet = floorsIntraday.dataset;
17.         if (floorsIntradayDataSet && floorsIntradayDataSet.length)
18.           return floorsIntradayDataSet;

```

```

19.     }
20.     return [];
21.   }
22. };
23. //...
24. const floorsDataSet = await getFloorsDataSet(startMoment, endMoment,
25.   fitbitAccount, fitbitApiClient);

```

*Listado 6. Obtener cantidad de pisos*

```

1.  const getSleepStatusDataSet = async (startMoment, fitbitAccount,
2.    fitbitApiClient) => {
3.    const sleepStatusEP = `/sleep/date/
4.      ${startMoment.format("YYYY-MM-DD")}.json`;
5.
6.    const sleepStatusResponse = await fitbitApiClient.get(sleepStatusEP,
7.      fitbitAccount.access_token, fitbitAccount.encoded_id);
8.
9.    if (verifyResponse(sleepStatusResponse, fitbitAccount))
10.     return sleepStatusResponse;
11.  };
12.
13.  const sleepSummary = sleepStatusDataSet[0].summary, dataSet = {};
14.  const sleep_status = sleepSummary && sleepSummary.totalTimeInBed
15.    ? Math.round(
16.      parseFloat(
17.        sleepSummary.totalMinutesAsleep / sleepSummary.totalTimeInBed
18.      ) * 10000
19.    ) / 100 : 100;

```

*Listado 7. Obtener estatus de sueño*

```

1.  let summary = await fitbitApiClient.get(summaryEP, access_token,
2.    encoded_id
3.  );
4.
5.  if (!verifyResponse(summary, fitbitAccount)) return;
6.  else if (!summary[0].summary) {
7.    throw Error("UNKNOWN");
8.  }
9.
10. summary = summary[0].summary;
11. const heartRateZones = summary.heartRateZones;

```

```

12.   const values = heartRateZones.reduce((acc, element, index) => {
13.     acc[`hrz_${index + 1}_calories`] = element.caloriesOut;
14.     acc[`hrz_${index + 1}_minutes`] = element.minutes;
15.     return acc;
16.   }, {});

```

*Listado 8. Resumen de zonas cardíacas y calorías quemadas.*

Una vez integradas las respuestas, se persisten en Cassandra™ – Listado 10 – y en PostgreSQL – Listado 11. En el caso de Cassandra™, las sentencias se ejecutan a través de su controlador para Node.js, *cassandra-driver*<sup>1</sup> – Listado 9.

```

1.   import cassandra from "cassandra-driver";
2.
3.   export const cassandraClient = new cassandra.Client({
4.     contactPoints: ["127.0.0.1:9042"],
5.     localDataCenter: "datacenter1",
6.     keyspace: "masters",
7.   });

```

*Listado 9. Instancia del controlador dedicado para la ejecución de sentencias en Cassandra™*

```

1.   async persistWearableStates(personID, clientAccountTimeZone, date,
2.     dataSet) {
3.     try {
4.       const dataSetKeys = Object.keys(dataSet);
5.       for (const time of dataSetKeys) {
6.         const dataMoment = moment.tz(`${date}T${time}`, moment.ISO_8601,
7.           clientAccountTimeZone).utc();
8.
9.         //store the utc time at which the local week starts
10.        const weekMoment = dataMoment.clone().startOf("isoWeek");
11.        const second = dataMoment.diff(weekMoment, "seconds");
12.        const elementValues = Object.values(dataSet[time]);
13.
14.        await cassandraClient.execute(
15.          `UPDATE wearable_states_by_person_by_week
16.            SET ${Object.keys(dataSet[time])
17.              .map((key) => `${key}=?`)}

```

<sup>1</sup> <https://github.com/datastax/nodejs-driver>

```

18.         .join(",")}, week_offset=?
19.         WHERE person_id=? and week=? and second=?;`,
20.         elementValues.concat([clientAccountTimeZone, personID,
21.         weekMoment.valueOf(), second]),{prepare: true}
22.         );
23.     }
24. } catch (error) {
25.     throw error;
26. }
27. }

```

Listado 10. Ejemplo de persistencia de datos en Cassandra<sup>200</sup>

```

1.  async insert(values) {
2.    const keys = Object.keys(values);
3.
4.    try {
5.      return await db.query(
6.        `INSERT INTO daily_summary (${keys.join(",")})
7.        VALUES (${keys.map((e, index) => `>${index + 1}`)})
8.        RETURNING *`,
9.        Object.values(values)
10.       );
11.    } catch (error) {
12.      throw error;
13.    }
14.  },

```

Listado 11. Ejemplo de persistencia de datos en PostgreSQL

## 3.5 Sprint 4

### 3.5.1 Analítica en el servidor. Árbol de decisión

De forma paralela a la recopilación y persistencia de datos provenientes del DVP, se ejecuta un módulo de analítica a menor escala en forma de árbol de decisión clasificador (ADC), desarrollado en Python. Este sub-proceso se inicia utilizando la función *spawn*, del módulo *child\_process*<sup>1</sup>, con la cual se ejecuta el archivo *main.py* que sirve de entrada a la ejecución del código en Python. Dicho sub-proceso

<sup>1</sup> [https://nodejs.org/api/child\\_process.html](https://nodejs.org/api/child_process.html)

se coordina desde el archivo *DecisionTree.js*, donde se pre-procesan los datos y seguidamente se pasan al ADC, el cual es implementado usando la biblioteca *scikit-learn* en Python. Dicha biblioteca posee herramientas para la fácil manipulación de matrices y la implementación de algoritmos de aprendizaje automático. Específicamente, se utiliza la clase *DecisionTreeClassifier*, disponible en el paquete *sklearn.tree*. La biblioteca *scikit-learn* implementa el algoritmo CART<sup>1</sup> (*Classification and Regression Trees*, Árboles de Clasificación y Regresión). CART construye árboles binarios usando la característica y el umbral que aportan la mayor ganancia de información en cada nodo [43].

#### 3.5.1.1 Pre-procesamiento de valores de ritmo cardíaco

Los datos de ritmo cardíaco se obtienen en pares **momento-valor**, donde **momento** es una cadena de caracteres que representa una hora del día, y **valor** un número que representa la frecuencia cardíaca para ese momento. Dado que un único valor de frecuencia cardíaca irregular no implica necesariamente un peligro potencial – sólo el hecho de ponerse de pie altera momentáneamente la frecuencia cardíaca – se decide agrupar y clasificar en rangos los valores similares que se registren en instantes de tiempo consecutivos. Para definir dichos rangos se tienen en cuenta la edad y género de la persona y se construye un árbol de decisión secundario con la ayuda de un experto.

El resultado es una categoría o clasificación, de cuatro posibles (Baja, Normal, Alta, Muy Alta), donde se agrupa cierta cantidad de valores numéricos de ritmo cardíaco, y una duración en segundos. Finalmente, a causa de que se utiliza un ADC – las características de entrada son valores numéricos – se codifican las categorías con un proceso de *codificación por etiquetas*, lo cual es factible ya que dichas categorías presentan un ordenamiento natural a priori. La Figura 14 ejemplifica el proceso.

---

<sup>1</sup> *scikit-learn* utiliza una versión optimizada de CART; aunque su implementación no soporta variables categóricas por el momento.

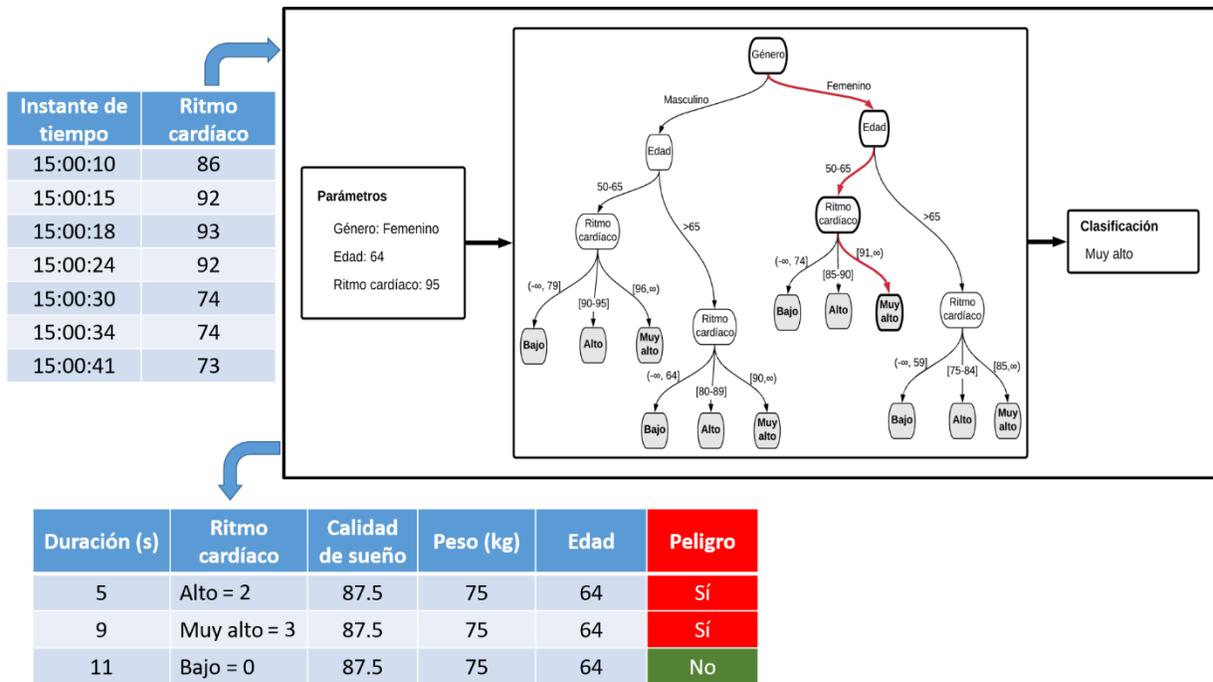


Figura 14. Pre-procesamiento de valores de ritmo cardíaco

### 3.5.1.2 Pre-procesamiento de valores de estatus de sueño

Los valores obtenidos en relación con el estatus de sueño del adulto mayor, reflejan los tiempos en minutos que la persona está acostada, tanto sin dormir como durmiendo. Según la Fundación Nacional de Sueño, una métrica de calidad de sueño aceptada es la razón que existe entre estas dos cantidades – se considera buena cuando dicha razón es igual o mayor a 0.85 [44] – por lo que fue una de las características utilizadas en el ADC.

### 3.5.2 Sistema de notificaciones

Una parte esencial del sistema de tele-cuidado es su capacidad de notificar al cuidador (o contacto registrado) del adulto mayor ante una situación de peligro potencial. Para lograr este propósito, se utiliza un mecanismo de notificaciones denominado *push notifications*, implementado en Angular para Aplicaciones Web Progresivas (PWA por sus siglas en inglés) a través del uso de *trabajadores de servicio*, y un sistema de suscripción-notificación entre el *front-end* (cliente Angular) y *backend* (servidor Node.js) de la aplicación.

Inicialmente, el cliente se suscribe para recibir notificaciones, para lo cual se generan llaves para la comunicación cliente-servidor – una llave pública compartida entre el cliente y el servidor, y una privada que se encuentra sólo en el servidor. El objeto suscripción, enviado al servidor a través de una petición POST, implementa una interfaz que provee un URL de suscripción, el cual es almacenado en la base de datos y se utiliza para enviar las notificaciones a los clientes suscritos. El proceso de envío de notificaciones se realiza a través del módulo *web-push* implementado para Node.js, el cual envía el objeto suscripción y una notificación que contiene la *carga útil*. La Figura 15 ilustra el buzón de notificaciones como es presentado al usuario de la aplicación.



Figura 15. Buzón de notificaciones

Al hacer clic sobre una notificación, el sistema re-direcciona al usuario hacia el panel de notificaciones y muestra los datos del incidente que corresponde a la notificación seleccionada.

## 3.6 Sprint 5

### 3.6.1 Módulo de VPC

A través del módulo de VPC, el sistema es capaz de monitorizar las actividades de los adultos mayores y detectar situaciones de posible peligro – por ejemplo, acercarse a un desnivel o escalera –, así como reaccionar ante los accidentes que suceden. En ambos casos se notificará a los cuidadores y contactos del adulto mayor. Este módulo

se desarrolló en Python, haciendo uso de las bibliotecas OpenCV (`cv2`) y NumPy (`numpy`) para el procesamiento en tiempo real de las imágenes en video.

La preparación inicial del módulo consiste en inicializar el flujo de video y consumir las imágenes provenientes de la cámara – Listado 12 –, para esto se utilizó el módulo de Python, `imutils`.

```

1. vs = VideoStream(src=0).start() # initialize the video stream
2. time.sleep(2.0) # allow the camera sensor to warmup
3. hsv, mask, res, marker = None, None, None, None
4. # loop over the frames from the video stream
5. while True:
6.     # grab the frame from the threaded video stream and resize it
7.     # to have a maximum width of 1000 pixels
8.     frame = vs.read()
9.     frame = imutils.resize(frame, width=1000)
    
```

*Listado 12. Se inicializa el flujo de video y se comienzan a procesar las imágenes en un ciclo `while`*

Durante el desarrollo de este módulo, se consideraron varios algoritmos de visión por computadora. A continuación, se describen dichos algoritmos, así como las ventajas y desventajas que se detectaron durante el proceso de investigación y pruebas.

#### 3.6.1.1 Seguimiento de personas en video utilizando Haar Cascade

Haar Cascade es un algoritmo de aprendizaje automático para detección de objetos en imagen o video, basado en el concepto de *características de tipo Haar*<sup>1</sup> [45].

El algoritmo demostró ser eficiente en la detección de rostros en imágenes, pero al ejecutarse utilizando un clasificador pre-entrenado para la detección de cuerpos de personas en video – Listado 13 –, la detección se perdía en tiempo real, y se constató un alto número de falsos positivos debido a la alta dependencia de las condiciones de iluminación y sombras. La Figura 16 muestra un ejemplo de uno de estos casos.

---

<sup>1</sup> Las *características de tipo Haar* se representan mediante la determinación de un área rectangular de la imagen, la cual se divide en varias partes. Por lo general, se visualizan en la forma de rectángulos adyacentes de color blanco o negro.

```
1. body_cascade =
2.     cv2.CascadeClassifier('./haarcascades/haarcascade_fullbody.xml')
3. def detect_body(img):
4.     body_img = img.copy()
5.     body_rects = body_cascade.detectMultiScale(body_img)
6.     for(x, y, w, h) in body_rects:
7.         cv2.rectangle(body_img, (x, y), (x+w, y+h),
8.             (255, 255, 255), 10)
9.     return body_img
10.
11. cap = cv2.VideoCapture(0)
12.
13. while True:
14.     ret, frame = cap.read(1)
15.     frame = detect_body(frame)
16.     cv2.imshow('video body detection', frame)
```

Listado 13. Detección de personas en video usando Haar Cascade



Figura 16. Instancias de pruebas con el algoritmo Haar Cascade

### 3.6.1.2 Seguimiento de personas en video utilizando HOG

En el descriptor de características HOG (*Histogram of Oriented Gradients*, Histograma de Gradientes Orientados), la distribución (histogramas) de las direcciones de los gradientes (gradientes orientados) representan las características. Este descriptor se centra más en la extracción de información con base en la distribución de gradientes de intensidad local o la dirección de los contornos [46], que en la detección de sombras e iluminación.

Al aplicarse el algoritmo que se muestra en Listado 14 se apreció una mejoría considerable en la capacidad de seguimiento de la persona en tiempo real en el video, pero la ocurrencia de falsos positivos aún era un problema en muchas de las pruebas. La Figura 17 muestra un ejemplo de uno de estos casos.

```
1.  import cv2
2.  import imutils
3.  from imutils.video import VideoStream
4.  import numpy as np
5.  from imutils.object_detection import non_max_suppression
6.  import time
7.
8.  # initialize the HOG descriptor/person detector
9.  hog = cv2.HOGDescriptor()
10.  hog.setSVMDetector(cv2.HOGDescriptor_getDefaultPeopleDetector())
11.
12.  vs = VideoStream(src=0).start()
13.
14.  time.sleep(2)
15.
16.  while True:
17.      # load the frame and resize it to (1) reduce detection time
18.      # and (2) improve detection accuracy
19.      image = vs.read()
20.      image = imutils.resize(image, width=min(400, image.shape[1]))
21.      orig = image.copy()
22.
23.      # detect people in the image
24.      (rects, weights) = hog.detectMultiScale(
25.          image, winStride=(4, 4), padding=(8, 8), scale=1.05)
26.      # draw the original bounding boxes
```

```

27.     for (x, y, w, h) in rects:
28.         cv2.rectangle(orig, (x, y), (x + w, y + h), (0, 0, 255), 2)
29.         # apply non-maxima suppression to the bounding boxes using a
30.         # fairly large overlap threshold to try to maintain overlapping
31.         # boxes that are still people
32.     rects = np.array([[x, y, x + w, y + h] for (x, y, w, h) in rects])
33.     pick = non_max_suppression(rects, probs=None, overlapThresh=0.65)
34.     # draw the final bounding boxes
35.     for (xA, yA, xB, yB) in pick:
36.         cv2.rectangle(image, (xA, yA), (xB, yB), (0, 255, 0), 2)
37.
38.     cv2.imshow("After NMS", image)

```

Listado 14. Detección de personas en video usando descriptor HOG

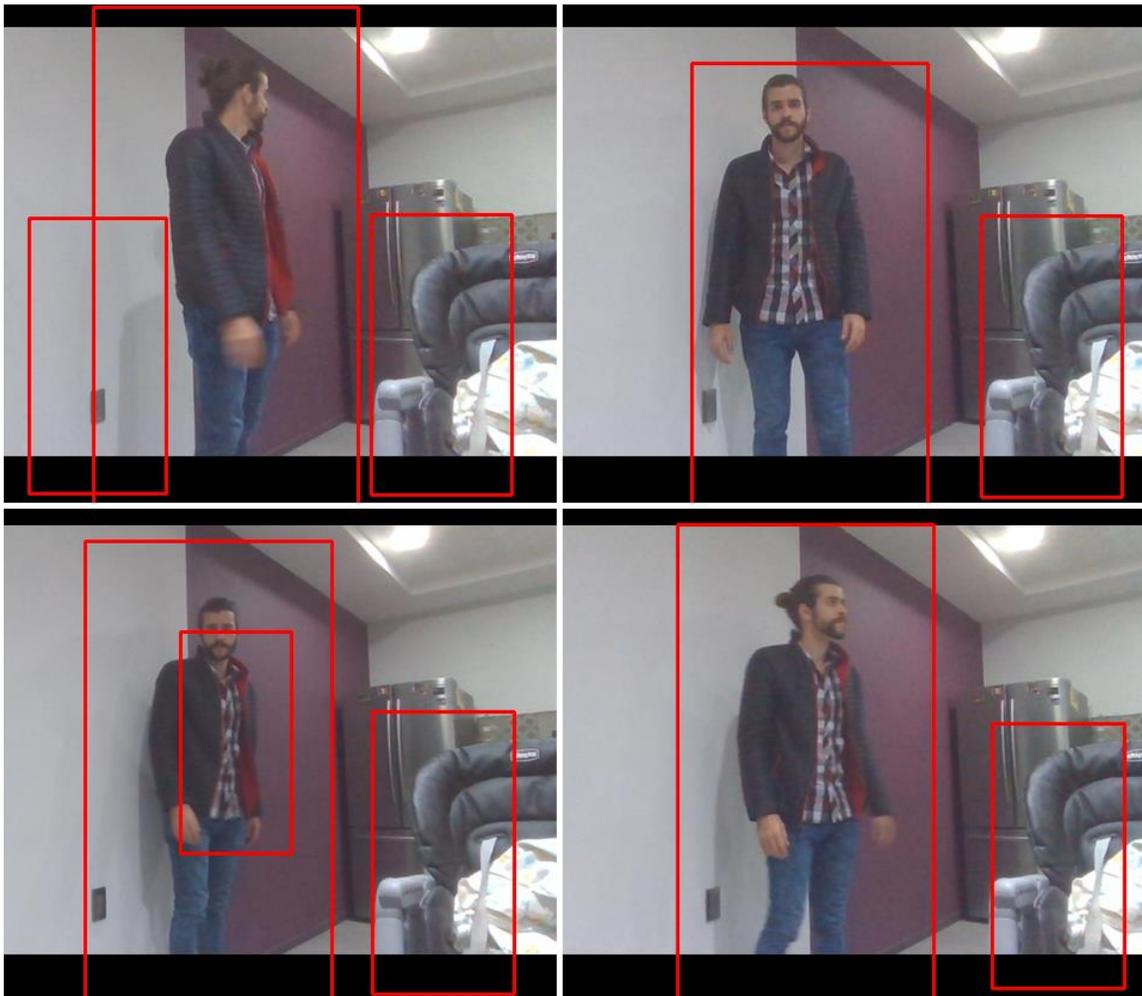


Figura 17. Instancias de pruebas con el algoritmo HOG

## 3.6.1.3 Seguimiento de personas en video utilizando dnn

Se utilizó `dnn` (*Deep Neural Network*, Red Neuronal Profunda), un módulo de aprendizaje automático incluido en OpenCV, con el cual se cargó un modelo pre-entrenado para la categorización de imágenes que contienen personas. Cada imagen (*frame*) captada en el ciclo, se suministra al modelo y se obtienen las detecciones. Finalmente, se usa `numpy` para filtrar dichas detecciones según su nivel de certeza – mayor que 0.5 – y determinar las coordenadas de los *recuadros contenedores* de cada persona detectada en la imagen – Listado 15.

```

1.  # load our serialized model from disk
2.  net = DNN(args["prototxt"] if args["prototxt"] else None,
3.           args["model"] if args["model"] else None).getNet()
4.
5.  # loop over the frames from the video stream
6.  while True:
7.      # grab the frame dimensions and convert it to a blob
8.      (h, w) = frame.shape[:2]
9.      blob = cv2.dnn.blobFromImage(frame, 0.007843, (299, 299), 127.5)
10.
11.     # pass the blob through the network and obtain the detections and
12.     # predictions
13.     net.setInput(blob)
14.     detections = net.forward()
15.     # loop over the detections
16.     box = []
17.     for i in np.arange(0, detections.shape[2]):
18.         # extract the confidence (i.e., probability) associated with
19.         # the prediction
20.         confidence = detections[0, 0, i, 2]
21.
22.         # filter out weak detections by ensuring the `confidence` is
23.         # greater than the minimum confidence
24.         if confidence > args["confidence"]:
25.             # extract the index of the class label from the
26.             # `detections`, then compute the (x, y)-coordinates of
27.             # the bounding box for the object
28.             idx = int(detections[0, 0, i, 1])
29.             if CLASSES[idx] == 'person':
30.                 box =

```

```
31.         detections[0, 0, i, 3:7] * np.array([w, h, w, h])
32.         (leftBorder, topBorder, rightBorder, bottomBorder) =
33.         box.astype("int")
```

*Listado 15. Carga del modelo pre-entrenado, filtro de detecciones y determinación de coordenadas de las personas en cada imagen*

La Figura 18 muestra un ejemplo de detección de la persona en video, utilizando el módulo `dnn` con el modelo pre-entrenado.



*Figura 18. Persona detectada por modelo pre-entrenado*

#### 3.6.1.4 Detección de posible accidente

##### *Caídas*

Una vez localizado el adulto mayor en la imagen, se hace un seguimiento constante y se registran las dimensiones del recuadro contenedor, el cual se establece como un rectángulo cuya altura es mayor a su ancho en todo momento que la persona esté de pie, tanto estática como en movimiento. Siguiendo esta premisa, se monitorizan y comparan constantemente las dimensiones del recuadro para confirmar la posición (vertical u horizontal) del adulto. Con el objetivo de disminuir los falsos positivos, se

introducen dos variaciones en el procedimiento, 1) una penalización de 0.2 al ancho del recuadro – útil para mitigar los falsos positivos generados por la posición de sentado – y 2) un período de tiempo de 5 segundos antes de clasificar la imagen como posible accidente – para contrarrestar los falsos positivos generados cuando la persona se agacha o inclina. La Figura 19 ilustra la detección de una caída, resaltando el recuadro contenedor en rojo.

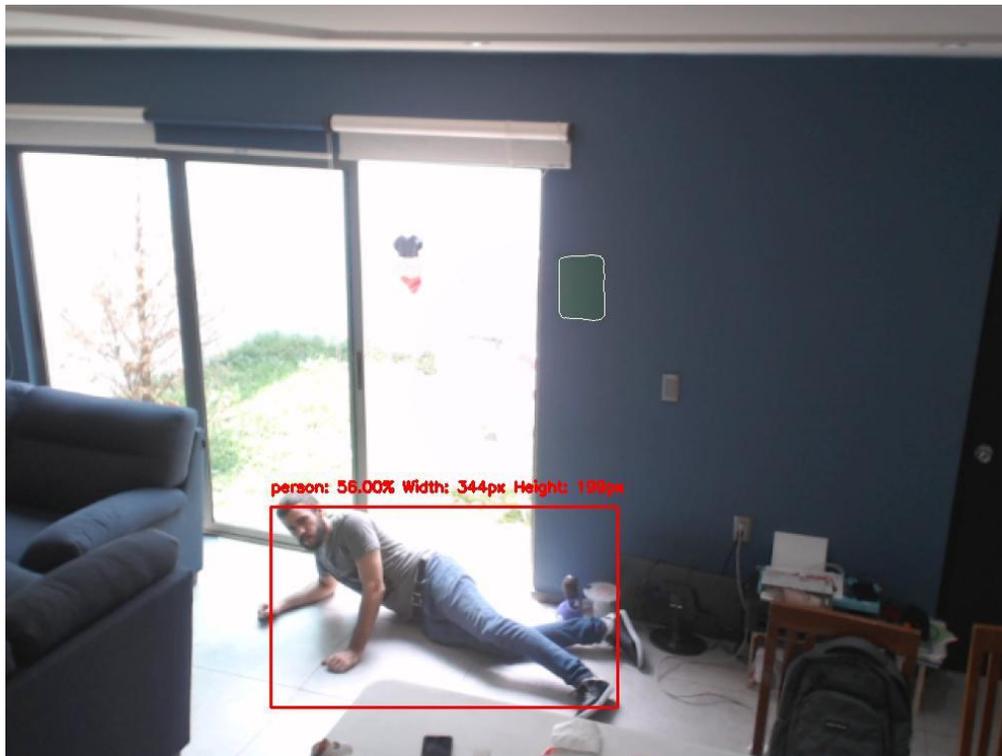


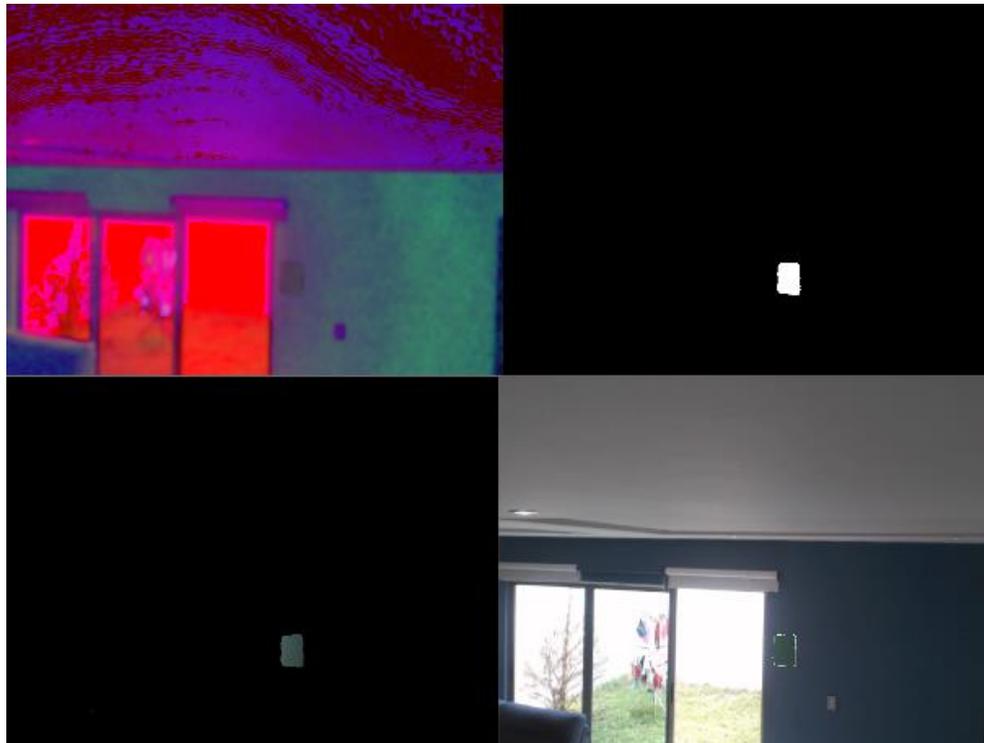
Figura 19. Detección de caída

### *Lugares de peligro potencial*

Con el objetivo de detectar si el adulto mayor se encuentra cerca de lugares peligrosos o con peligro de provocar caídas, se colocaron marcadores rectangulares de color rojo cerca de estos lugares.

A continuación, se explica el proceso de detección de los marcadores. Se difuminó la imagen para reducir el *ruido*, se transformó la imagen del espacio BGR (*Blue-Green-*

*Red, Azul-Verde-Rojo*)<sup>1</sup> al espacio HSV (*Hue-Saturation-Value, Matiz-Saturación-Valor*)<sup>2</sup>. Seguidamente, se definieron los límites superior e inferior para el color requerido y se obtuvo una máscara binaria a partir de la imagen original, en la cual, los píxeles blancos representan las áreas que coinciden con el rango de colores utilizado. Luego, se aplica una comparación bit a bit de la imagen con dicha máscara; resaltando las áreas de la imagen que intersectan con los píxeles blancos de la máscara. Finalmente, se extraen dichas detecciones, se filtran según su tamaño y forma, obteniéndose la que corresponde al marcador buscado, del cual se extrae el contorno. La Figura 20 ilustra los resultados parciales de cada una de las fases.



*Figura 20. Detección de marcador rectangular verde. Espacio HSV (superior izquierda), máscara binaria (superior derecha), intersección bit-a-bit (inferior izquierda), y resultado resaltado con contorno (inferior derecha)*

<sup>1</sup> OpenCV representa las imágenes como arreglos NumPy en orden inverso, de ahí que los componentes de colores en el rango RGB se obtengan en orden BGR.

<sup>2</sup> El canal Matiz modela el tipo de color, siendo muy útil en el procesamiento de imágenes cuando se necesita segmentar objetos según su color.

El contorno se almacena como un arreglo `numpy` de coordenadas, el cual se intersecta con el recuadro que contiene el adulto mayor para detectar acercamientos, y con ello situaciones de peligro potencial.

### 3.6.1.5 Envío de notificación con imagen

Cuando el sistema detecta una situación de peligro potencial, envía una notificación a nuestro servidor a través de una petición `http`, incluyendo una imagen del momento que generó el evento. Para esto se utilizan los módulos `requests` – para realizar la petición `HTTP` – y `PIL` – para capturar y almacenar la imagen – como se muestra en los listados 16 y 17 respectivamente.

```

1.  import io
2.  import cv2
3.  from PIL import Image
4.
5.  def takePicture(frame):
6.      try:
7.          frame_im = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
8.          img = io.BytesIO()
9.          pil_im = Image.fromarray(frame_im)
10.         pil_im.save(img, format="JPEG")
11.         img.seek(0)
12.         pictureData = img.read()
13.     except Exception as e:
14.         print(e)
15.
16.     return pictureData

```

*Listado 16. Captura de imagen del incidente o accidente*

```

1.  import requests
2.  import uuid
3.
4.  def sendNotification(params):
5.      category = 'accident' if params['accident'] else 'incident'
6.
7.      try:
8.          requests.post(
9.              "http://localhost:3000/api/v1/camera/1/incidents/new",

```

```

10.         files=[
11.             (
12.                 'file',
13.                 ('picture.jpeg', params['picture'], 'image/jpeg')
14.             )
15.         ], params={
16.             'id': uuid.uuid4(), 'accident': params['accident']
17.         }, timeout=.001)
18.
19.     except requests.exceptions.RequestException as e:
20.         print('RequestException', e)
21.     except Exception as e:
22.         print('Exception', e)

```

Listado 17. Envío de notificación desde el módulo de VPC

## 3.7 Sprint 6

### 3.7.1 Módulo de aprendizaje automático con Apache Spark™

Para el desarrollo de este módulo se utilizó el conector Spark-Cassandra 2.5.0, Apache Spark™ 2.4.5, y el lenguaje de programación Scala, versión 2.11.12. El módulo se desarrolló haciendo uso de la API para RDD de Spark™, la cual contiene clases y funciones que facilitan sobremanera la lectura, manipulación, análisis y almacenamiento de datos provenientes de Cassandra™. Para el aprendizaje automático se utilizó el módulo `org.apache.spark.mllib`. El código compilado se integró al sistema en forma de archivo `.jar`<sup>1</sup>, y su proceso de ejecución se coordina desde el servidor. Dicho proceso se inicializa de forma similar al módulo desarrollado en Python: mediante la función `spawn`, del módulo `child_process`, con la cual se ejecuta el archivo `.jar` que sirve de entrada a la ejecución del código en Scala.

El objetivo de este módulo es clasificar los datos provenientes del DVP en posibles incidentes, utilizando los modelos entrenados. Para este problema de clasificación binaria – posible incidente o no – se entrenaron dos modelos basados en algoritmos de aprendizaje supervisado: Máquinas de Soporte Vectorial Lineales (SVM, por sus

<sup>1</sup> <https://docs.oracle.com/javase/tutorial/deployment/jar/basicsindex.html>

siglas en inglés) los cuales representan un método estándar para tareas de clasificación a gran escala [47], y Regresión Logística, método ampliamente utilizado para predecir respuestas binarias [48]. Ambos enfoques tienen soporte para el análisis de RDDs en Spark™.

### 3.7.1.1 SVM

Se utilizaron las clases `SVMWithSGD` y `SVModel`, del módulo `org.apache.spark.mllib.classification`, para entrenar y almacenar el modelo SVM, respectivamente; de forma análoga, se utilizaron las clases `LogisticRegressionModel` y `LogisticRegressionWithLBFGS` para preparar el modelo de Regresión Logística. Durante el entrenamiento, se modificaron secuencialmente los puntos de división de conjuntos de entrenamiento y prueba para ambos modelos, y el número de iteraciones de gradiente descendiente en el caso de SVM, mientras que para la Regresión Logística el número de clases se estableció en 2, siendo un problema de clasificación binaria (la cantidad de posibles resultados o clases es 2), hasta obtener la mayor área bajo la curva (AUC<sup>1</sup>, por sus siglas en inglés) ROC (*Receiver Operating Characteristic*, Característica Operativa del Receptor), la cual se utilizó como medida del poder discriminatorio del modelo [49] durante el entrenamiento. El valor máximo para el AUC es 1.0, indicando una prueba (en teoría) perfecta (i.e., 100% sensible y 100% específica) [50]. Los listados 18 y 19 contienen el proceso de entrenamiento de cada modelo.

```

1. def trainModel(data: RDD[LabeledPoint]): SVMModel = {
2.   var model: SVMModel = null
3.   var bestModel: SVMModel = null
4.   var highestAccuracy = 0d
5.   var highestAreaUnderROC = 0d
6.
7.   for (i <- Range.apply(1, 10, 1)) {
8.     for (j <- Range.apply(1, 101, 1)) {
9.       val splits = data.randomSplit(
10.        Array(i.toDouble / 10, 1 - i.toDouble / 10), seed = 11L

```

<sup>1</sup> Durante el resto del documento, el término AUC hará referencia al *área bajo la curva ROC*.

```

11.     )
12.     val training = splits(0).cache()
13.     val test = splits(1)
14.
15.     // Run training algorithm to build the model
16.     model = SVMWithSGD.train(training, j)
17.     model.setThreshold(0.5)
18.
19.     val scoreAndLabels = test.map { point =>
20.     val score = model.predict(point.features)
21.         (score, point.label)
22.     }
23.     val areaUnderROC =
24.         new BinaryClassificationMetrics(scoreAndLabels).areaUnderROC()
25.     val accuracy = new MulticlassMetrics(scoreAndLabels).accuracy
26.     if (areaUnderROC > highestAreaUnderROC ||
27.         areaUnderROC == highestAreaUnderROC &&
28.         accuracy > highestAccuracy) {
29.         highestAreaUnderROC = areaUnderROC
30.         highestAccuracy = accuracy
31.         bestModel = model
32.     }
33.     }
34.     }
35.     bestModel.save(sc, path)
36.     bestModel
37. }

```

*Listado 18. Entrenamiento del modelo SVM*

```

1.     def trainModel(data: RDD[LabeledPoint]): LogisticRegressionModel = {
2.         var model: LogisticRegressionModel = null
3.         var bestModel: LogisticRegressionModel = null
4.         var highestAccuracy = 0d
5.         var highestAreaUnderROC = 0d
6.
7.         for (i <- Range.apply(1, 10, 1)) {
8.             val splits = data.randomSplit(
9.                 Array(i.toDouble / 10, 1 - i.toDouble / 10), seed = 11L)
10.            val training = splits(0).cache()
11.            val test = splits(1)
12.
13.            // Run training algorithm to build the model

```

```

14.     model = new LogisticRegressionWithLBFGS().run(training)
15.
16.     val scoreAndLabels = test.map {
17.         case LabeledPoint(label, features) =>
18.             val prediction = model.predict(features)
19.             (prediction, label)
20.     }
21.
22.     val areaUnderROC =
23.         new BinaryClassificationMetrics(scoreAndLabels).areaUnderROC()
24.     val accuracy = new MulticlassMetrics(scoreAndLabels).accuracy
25.     if (areaUnderROC > highestAreaUnderROC ||
26.         areaUnderROC == highestAreaUnderROC &&
27.         accuracy > highestAccuracy) {
28.         highestAreaUnderROC = areaUnderROC
29.         highestAccuracy = accuracy
30.         bestModel = model
31.     }
32. }
33.     bestModel.save(sc, path)
34.     bestModel
35. }

```

*Listado 19. Entrenamiento del modelo Regresión Logística*

### 3.7.1.2 Detección de incidentes

Para el proceso de detección de incidentes, el módulo recibe los datos correspondientes al rango de tiempo que se desea clasificar, con estos se construye la sentencia CQL y se obtienen los estados desde Cassandra™. Dichos estados se pre-procesan y se convierten en RDDs de vectores numéricos, se entregan al modelo para obtener las clasificaciones. Si las clasificaciones positivas representan al menos un 25% de las totales, se persisten los datos del incidente en Cassandra™, y se envía su identificador y el de la cuenta cliente correspondiente, al servidor. El Listado 20 muestra el proceso de pre-procesamiento de datos, clasificación y persistencia de incidente.

```

1.     var statesRDD: CassandraRDD[StatesByPersonByWeek] = null
2.     if (fromWeek == toWeek) {

```

```

3.     statesRDD = sc.cassandraTable[StatesByPersonByWeek] (
4.         "masters", "wearable_states_by_person_by_week"
5.     ).where(
6.         s"person_id=$clientAccountID AND week=$fromWeek
7.         AND second>=$fromSecond AND second<$toSecond")
8. } else {
9.     statesRDD = sc.cassandraTable[StatesByPersonByWeek] (
10.        "masters", "wearable_states_by_person_by_week"
11.    ).where(
12.        s"person_id=$clientAccountID AND week=$fromWeek
13.        AND second>=$fromSecond")
14.    statesRDD = statesRDD.union(
15.        sc.cassandraTable[StatesByPersonByWeek] (
16.            "masters", "wearable_states_by_person_by_week"
17.        ).where(
18.            s"person_id=$clientAccountID AND week=$toWeek
19.            AND second<$toSecond")
20.        )
21.    ).asInstanceOf[CassandraRDD[StatesByPersonByWeek]]
22. }
23. statesRDD.cache()
24. val vectorsRDD: RDD[linalg.Vector] = StatesRDDToVectorsRDD(statesRDD)
25. vectorsRDD.cache()
26.
27. val vectorCount = vectorsRDD.count()
28. if (vectorCount > 0) {
29.     var predictions: List[Double] = null
30.     var algorithm: MLAlgorithms = null
31.
32.     if (modelType == "lr") {
33.         algorithm = new LogisticRegression(sc)
34.     } else if (modelType == "svm") {
35.         algorithm = new SVM(sc)
36.     }
37.
38.     if (optionsMap.contains("--model-dir")) {
39.         val modelDir = optionsMap("--model-dir")
40.         if (modelDir.length > 0)
41.             algorithm.setModelDirPath(modelDir)
42.     }
43.
44.     predictions = algorithm.getPredictions(vectorsRDD)
45.     val aux: Array[linalg.Vector] = vectorsRDD.collect()

```

```

46.     var positivePredictions: mutable.MutableList[WS]
47.         = mutable.MutableList[WS] ()
48.     predictions.zipWithIndex.foreach(t => {
49.         if (t._1 != 0.0) {
50.             val vector: linalg.Vector =
51.                 aux.drop(t._2).dropRight(aux.length - t._2 - 1)(0)
52.             positivePredictions += WS(
53.                 Option(vector(0).toInt),
54.                 Option(vector(1).toFloat),
55.                 Option(vector(2).toInt),
56.                 Option(vector(3).toInt)
57.             )
58.         }
59.     })
60.
61.     val positives = positivePredictions.length
62.     if (positives >= predictions.length / 4) {
63.         implicit val userFormat = Json.format[WS]
64.         val incidentID = persistIncident(sc, positivePredictions.toList)
65.         new Requests()
66.             .sendNotification(
67.                 Json.obj("incident_id" -> incidentID)
68.                     .deepMerge(Json.obj("client_account_id" -> clientAccountID))
69.             )
70.     }
71. }

```

Listado 20. Pre-procesamiento de datos, clasificación y persistencia de incidente con Spark<sup>™</sup>

## 3.8 Sprint 7

### 3.8.1 Panel de incidentes

En este panel se muestran la fecha y hora en que ocurre el incidente, el nombre del adulto mayor involucrado en el mismo, si el sistema lo clasificó inicialmente como *accidente* o como *incidente*, los datos provenientes del DVP registrados poco tiempo antes de su ocurrencia, así como información visual – en caso de existir – proveniente de la cámara – Figura 21. El usuario de la aplicación revisa dicha información y confirma manualmente la ocurrencia de un accidente real. Esta confirmación se refleja en las bases de datos y se utiliza para actualizar el modelo para clasificaciones posteriores.

## Capítulo 3 | Aplicación de la Metodología

Clasificación inicial	Fecha/hora	Nombre	Acciones
Accidente	18 de mayo de 2020 9:11	José Luis Sánchez Cervantes	Procesar incidente Accident
Incidente	17 de mayo de 2020 19:24	José Luis Sánchez Cervantes	
Incidente	16 de mayo de 2020 19:51	José Luis Sánchez Cervantes	

Figura 21. Panel de incidentes

## Capítulo 4 Resultados

Para mostrar el funcionamiento del enfoque planteado en esta investigación, se desarrolló el sistema “Sentinel”, acorde a la arquitectura propuesta en capítulos previos, bajo la modalidad de aplicación Web.

A través de esta aplicación, los usuarios son capaces de interactuar con las funcionalidades del sistema en concordancia con sus roles. Los *Cuidadores* y *Contactos* monitorizan el estado de salud del adulto mayor a través de tablas y gráficas con datos obtenidos a través del DVP, agregados, y ordenados cronológicamente, así como imágenes captadas con el módulo de VPC. Por su parte, los administradores tienen la opción de registrar nuevos usuarios, así como nuevas cuentas de adultos mayores en el sistema. De igual forma, el sistema envía notificaciones ante cualquier situación de peligro potencial detectada haciendo uso de algoritmos de aprendizaje automático y técnicas de visión por computadora, sin necesidad de que el usuario inicie la interacción.

### 4.1 Caso de estudio

#### 4.1.1 Detección de accidentes y parámetros de salud irregulares.

Para este caso de estudio, consideremos la siguiente situación.

- Un adulto mayor, de 80 años de edad, se encuentra solo en su hogar
- No durmió lo suficiente durante la noche
- Ha realizado actividades durante el día, en una medida mayor a lo acostumbrado
- Su frecuencia cardíaca ha estado fuera de los rangos normales durante el día

Esta situación genera las siguientes interrogantes:

1. ¿De qué manera un adulto mayor puede ser monitorizado para recibir atención pronta por parte de un cuidador, en caso de que ocurra algún incidente o accidente?

2. ¿De qué manera el cuidador obtendrá información sobre los signos vitales y actividades que ha realizado el adulto mayor sin que involucre la inversión de una gran cantidad de tiempo y esfuerzo?

3. ¿De qué manera el usuario cuidador se enterará sobre la ocurrencia de incidentes y accidentes en el inmueble en el que habita el adulto mayor para ofrecerle ayuda a la mayor brevedad posible?

Una alternativa de solución se presenta mediante el Internet de las Cosas, con la integración de un dispositivo vestible o ponible y un dispositivo domótico de videovigilancia. Los mismos son capaces de monitorizar los parámetros de salud del adulto mayor en cuestión, así como su nivel de actividad física y, por ende, su cansancio, durante el día. Los datos recopilados se envían a un módulo para la detección de anomalías en signos vitales y actividades de adultos mayores, donde se generan alertas basadas en Analítica Big Data.

Primeramente, los datos detectados por el sistema reflejan la baja calidad de sueño que presenta la persona en esta situación, así como el nivel de actividad física expresado en cantidad de pasos y pisos. De igual forma se observan valores irregulares en la frecuencia cardíaca del adulto mayor en cuestión. La Figura 22 muestra los datos durante el día del incidente, como se muestran al cuidador.

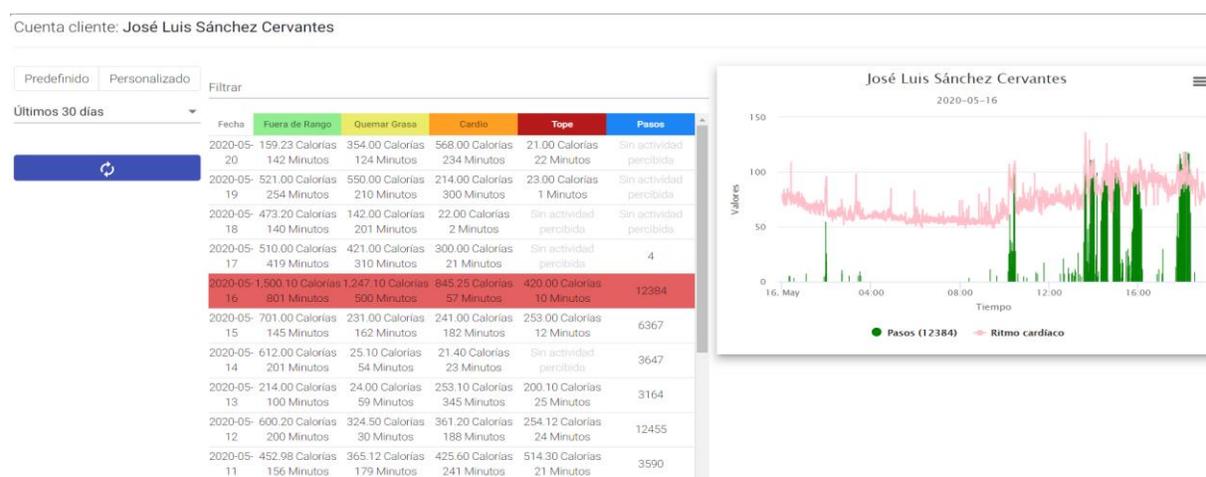


Figura 22. Datos registrados durante el día del incidente

Estos parámetros se procesan en los módulos de aprendizaje automático desarrollados, por una parte, en Python haciendo uso de un árbol de decisión clasificador, y por otra, en Spark™ utilizando SVMs y Regresión Logística. Al detectarse un conjunto de datos anómalos, se genera un incidente, el cual se persiste en las bases de datos, y se notifica al cuidador de la situación, brindando a la vez ayuda visual con los parámetros de salud previos al incidente y que probablemente fueron la causa del mismo. La Figura 23 muestra los datos del incidente, generados por dichos módulos.



*Figura 23. Datos del incidente, y parámetros de salud registrados previos al mismo*

La combinación de estos factores contribuyó a la ocurrencia de un accidente en ausencia de un acompañante o familiar, el cual se detectó a través del módulo de VPC instalado en el hogar del adulto mayor – Figura 24.

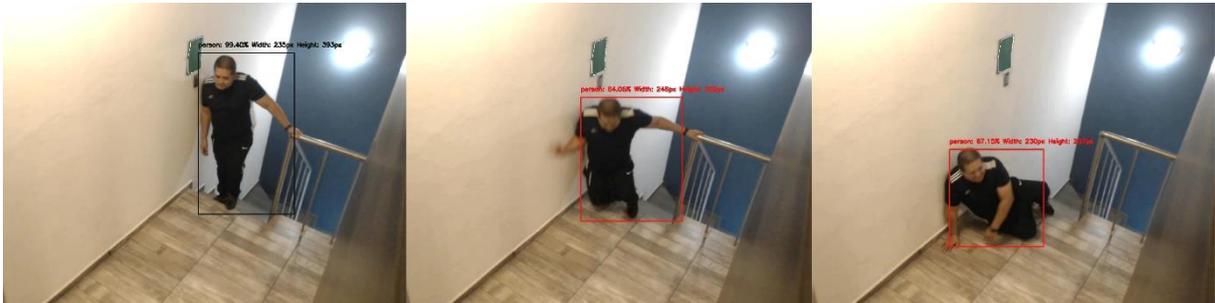


Figura 24. Accidente detectado por el módulo de VPC

El caso de estudio descrito demuestra la utilidad de Sentinel como una alternativa para el cuidado de la salud de adultos mayores mediante la monitorización de actividades y signos vitales, aprovechando las posibilidades que brinda el Internet de las Cosas, al integrar un DVP y un módulo de VPC. El resultado es un sistema capaz de alertar de manera pronta y efectiva acerca de posibles incidentes y accidentes que, de otra manera, quedarían desapercibidos, dejando al afectado sin atender durante un período de tiempo considerablemente mayor.

## 4.2 Evaluación y resultados

Para comprobar que el algoritmo de VPC seleccionado tiene la mayor efectividad dentro de los estudiados para el enfoque propuesto en esta tesis y para utilizar en el sistema Sentinel, se presenta un análisis de las métricas precisión (*precision*) y exhaustividad<sup>1</sup> (*recall*).

Dichas métricas son utilizadas en el dominio de la recuperación de información para medir qué tan bien un sistema informático realiza búsquedas, reconoce patrones y recupera información solicitada por un usuario [52]; su uso en gráficas PRC (*Precision Recall Curve*, Curva de Precisión-Exhaustividad) es recomendado para instancias de clasificación binaria en conjuntos de datos no balanceados<sup>2</sup> [53].

<sup>1</sup> También identificada como *susceptibilidad* [51].

<sup>2</sup> En los conjuntos de datos no balanceados, la cantidad de instancias negativas es muy superior a la cantidad de instancias positivas [53].

En el contexto de esta tesis, se utilizaron para medir qué tan bien cada algoritmo probado, realiza la detección de personas en video.

Reales *sin personas* (RSP) – cantidad de imágenes sin personas.

Reales *con personas* (RCP) – cantidad de imágenes con personas.

Clasificadas *sin personas* (CSP) – cantidad de imágenes clasificadas *sin personas*.

Clasificadas *con personas* (CCP) – cantidad de imágenes clasificadas *con personas*.

Verdaderos positivos (VP) – cantidad de imágenes con personas, clasificadas correctamente.

Falsos positivos (FP) – cantidad de imágenes sin personas, clasificadas incorrectamente.

Verdaderos negativos (VN) – cantidad de imágenes sin personas, clasificadas correctamente.

Falsos negativos (FN) – cantidad de imágenes con personas, clasificadas incorrectamente.

Precisión (P) – razón de imágenes con personas, clasificadas correctamente, sobre el total de imágenes clasificadas *con personas* – Ecuación 1.

$$P = \frac{VP}{(VP + FP)}$$

*Ecuación 1. Cálculo de Precisión*

Exhaustividad (E) – razón de imágenes con personas, clasificadas correctamente, sobre el total de imágenes con personas – Ecuación 2.

$$E = \frac{VP}{(VP + FN)}$$

*Ecuación 2. Cálculo de Exhaustividad*

A continuación, se muestran los resultados obtenidos para cada algoritmo de detección de personas en video, usando diferentes ángulos de visión para la cámara, así como diferentes locales y fuentes de iluminación.

#### 4.2.1 Haar Cascade

En el caso del algoritmo *Haar Cascade* se realizaron 76 pruebas. La TABLA III representa los valores de la matriz de confusión correspondiente.

	CSP	CCP
RSP	6	33
RCP	18	19

TABLA III Matriz de confusión generada a partir de las clasificaciones mediante Haar Cascade

Las ecuaciones 3 y 4 muestran la precisión y exhaustividad obtenidas por el algoritmo *Haar Cascade*, respectivamente.

$$P_{HC} = \frac{19}{(19 + 33)} \approx 0.37$$

Ecuación 3. Cálculo de la precisión del algoritmo Haar Cascade en la detección de personas en video.

$$E_{HC} = \frac{19}{(19 + 18)} \approx 0.51$$

Ecuación 4 Cálculo de la exhaustividad del algoritmo Haar Cascade en la detección de personas en video.

#### 4.2.2 Histograma de Gradientes Orientados (HOG)

En el caso del algoritmo HOG se realizaron 59 pruebas. La TABLA IV representa los valores de la matriz de confusión correspondiente.

	CSP	CCP
RSP	3	25
RCP	11	20

TABLA IV Matriz de confusión generada a partir de las clasificaciones mediante HOG

Las ecuaciones 5 y 6 muestran la precisión y exhaustividad obtenidas por el algoritmo HOG, respectivamente.

$$P_{HOG} = \frac{20}{(20 + 25)} \approx 0.44$$

*Ecuación 5. Cálculo de la precisión del algoritmo HOG en la detección de personas en video.*

$$E_{HOG} = \frac{20}{(20 + 11)} \approx 0.65$$

*Ecuación 6 Cálculo de la exhaustividad del algoritmo HOG en la detección de personas en video.*

### 4.2.3 Red Neuronal Profunda (DNN)

En el caso del uso de DNN, se realizaron 70 pruebas. La TABLA V representa los valores de la matriz de confusión correspondiente.

	CSP	CCP
RSP	28	2
RCP	1	39

*TABLA V Matriz de confusión generada a partir de las clasificaciones mediante DNN*

Las ecuaciones 7 y 8 muestran la precisión y exhaustividad obtenidas con DNN, respectivamente.

$$P_{DNN} = \frac{39}{(39 + 2)} \approx 0.95$$

*Ecuación 7. Cálculo de la precisión del enfoque usado con DNN en la detección de personas en video.*

$$E_{DNN} = \frac{39}{(39 + 1)} \approx 0.97$$

*Ecuación 8 Cálculo de la exhaustividad del enfoque usado con DNN en la detección de personas en video.*

El análisis de las métricas aplicadas, y los resultados obtenidos con cada algoritmo, justifican la selección del enfoque que utiliza DNN para la detección y seguimiento de personas en video. El mismo se aplicó con diferentes ángulos de visión, así como condiciones de iluminación detectando correctamente las personas en video.

## Capítulo 5 Conclusiones y recomendaciones

### 5.1 Conclusiones

El gran volumen de datos recopilado y almacenado en Cassandra™, es una fuente de información viable para ser utilizada en la compilación de un expediente clínico digital, útil para consultas médicas posteriores. De igual forma, servirá como base para el análisis histórico de datos que ayude a identificar patrones de salud para la detección temprana de enfermedades crónico-degenerativas y cardiovasculares tanto en el adulto mayor como en personas jóvenes.

### 5.2 Recomendaciones

Con el objetivo de mejorar los tiempos de respuesta ante un accidente, así como la proactividad del sistema, se recomienda como trabajo futuro, la implementación e integración de un módulo para el envío de alertas de manera automática o semi-automática (previa validación del cuidador) a servicios de asistencia médica, a través de mensajes de texto.

También se recomienda mejorar el módulo de VPC, para contrarrestar los efectos de parámetros ambientales como el nivel de iluminación, natural o artificial, que causan variaciones constantes en los tonos de los marcadores, siendo necesario recalibrar el detector de marcadores para ajustarse a cada cambio.

En caso de que el adulto mayor abandone su lugar de residencia, es recomendable incluir un sistema de trazabilidad para continuar la monitorización en los casos en que la persona se encuentra fuera del rango de visión de las cámaras.

Finalmente, se sugiere la utilización de un DVP más moderno y con más sensores; un oxímetro de pulso, por ejemplo, permite monitorizar los niveles de saturación de oxígeno en la sangre de manera que apoye con la detección de embolias o enfermedades cardiovasculares, las cuales no se obtienen utilizando el *hardware* usado en este proyecto de tesis de maestría.

# Anexos y apéndices

ISSN 1870-4069

## An Architecture for an IoT-based Telecare System for the Elderly Using Big Data Analytics

Jesús Miguel Echevarría Díaz<sup>1</sup>, José Luis Sánchez Cervantes<sup>2</sup>,  
Luis Omar Colombo Mendoza<sup>1</sup>, Giner Alor Hernández<sup>1</sup>, Ignacio López Martínez<sup>1</sup>

<sup>1</sup> Tecnológico Nacional de México,  
Instituto Tecnológico Orizaba,  
Mexico

<sup>2</sup> CONACYT,  
Instituto Tecnológico de Orizaba,  
Mexico

{jmechevarria2015, nachocero}@gmail.com, jlsanchez@conacyt.mx,  
{lcolombom, galor}@ito-depi.edu.mx

**Abstract.** The accelerated aging of the population is a real and constant situation worldwide. Long-lived population is the most affected demographic sector in terms of health and home safety. The advent of Big Data, in conjunction with the Internet of Things (IoT) omnipresence, have made telecare-based systems an increasingly desirable alternative when dealing with the treatment and remote care of the elderly and frail, allowing them to go on with their lives in a normal, almost independent manner. This situation has spawned an ever increasing demand for innovative telecare solutions. In order to present an alternative solution to the problem at hand, this paper proposes a telecare system architecture based on IoT and Big Data Analytics (BDA) to monitor the user's vital signs and activities, send alerts to the caregivers whenever necessary, with the goal of detecting and preventing accidents in the home.

**Keywords:** big data analytics, internet of things, telecare, wearable, decision tree.

### 1 Introduction

The impact of BDA techniques in healthcare organizations shows promising research directions, especially when implementing novel use cases for potential healthcare frameworks. Over 10 million people in Mexico are 60 years of age or more, frequently spending long periods of time alone at home; which increases the probabilities of going through a medical emergency while on their own.

pp. 53-59; rec. 2019-09-04; acc. 2019-10-07 53 *Research in Computing Science* 149(5), 2020

*Figura 25. Echevarría-Díaz, J. M., Sánchez-Cervantes, J. L., Colombo-Mendoza, L. O., Alor-Hernández, G. & Martínez-López I. (2020). An Architecture for an IoT-based Telecare System for the Elderly Using Big Data Analytics. Research in Computing Science, 149, 53-59.*

## Anexos y apéndices



Oficio No.: 0031  
Asunto: Carta de uso de sistema  
Orizaba, Ver. a 10 de julio del 2020

**Dr. José Luis Sánchez Cervantes**  
Catedrático CONACYT  
Instituto Tecnológico de Orizaba

Por medio de la presente me permito enviar a usted un cordial saludo, así mismo me permito informarle del interés y compromiso de la empresa RODAS COMPUTACIÓN S.A. de C. V. por brindar el apoyo institucional necesario para llevar a cabo el proyecto "Sistema de Tele-cuidado de Adultos Mayores basado en Analítica Big Data para el Internet de las Cosas".

El desarrollo de este proyecto es llevado a cabo por el ICI. Jesús Miguel Echevarría Díaz como parte de su tesis de Maestría en Sistemas Computacionales, así como por investigadores del Instituto Tecnológico de Orizaba en vinculación con RODAS S.A. de C.V.

Esperando obtener los beneficios que emanen tras la colaboración entre el Instituto Tecnológico de Orizaba y RODAS S.A. de C.V. me despido de usted quedando a sus finas atenciones.

ATENTAMENTE  
Dirección de Innovación y Desarrollo Tecnológico

Mtra. Carolina Bringas Centeno  
RESPONSABLE DEL AREA DE PROYECTOS  
TECNOLÓGICOS  
RODAS COMPUTACIÓN S.A. DE C.V.



Sur 7 No. 18 Col. Centro, Orizaba, Ver ☎ 272 72 59911 - 272 72 64480 🌐 [www.rodascomputacion.com.mx](http://www.rodascomputacion.com.mx)

*La mejor tecnología de información, con el trato más humano*

Figura 26. Carta de uso de sistema – Empresa Rodas®

## Bibliografía

- [1] F. C. Nicolalde, F. Silva, B. Herrera, and A. Pereira, "Big Data Analytics in IOT: Challenges, Open Research Issues and Tools," in *World Conference on Information Systems and Technologies*, 2018, pp. 776–777.
- [2] M. Marjani *et al.*, "Big IoT data analytics: architecture, opportunities, and open research challenges," *IEEE Access*, vol. 5, pp. 5247–5261, 2017.
- [3] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010, doi: 10.1016/j.comnet.2010.05.010.
- [4] R. Mital, J. Coughlin, and M. Canaday, "Using big data technologies and analytics to predict sensor anomalies," 2015.
- [5] N. Golchha, "Big data-the information revolution," *Int. J. Adv. Res*, vol. 1, no. 12, pp. 791–794, 2015.
- [6] J. Barlow, D. Singh, S. Bayer, and R. Curry, "A systematic review of the benefits of home telecare for frail elderly people and those with long-term conditions," *Journal of Telemedicine and Telecare*, vol. 13, no. 4, pp. 172–179, 2007, doi: 10.1258/135763307780908058.
- [7] D. L. Schriger, "7 - Approach to the Patient with Abnormal Vital Signs," in *Goldman's Cecil Medicine (Twenty Fourth Edition)*, L. Goldman and A. I. Schafer, Eds. Philadelphia: W.B. Saunders, 2012, pp. 27–30.
- [8] "Machine Learning: What it is and why it matters | SAS." [https://www.sas.com/en\\_us/insights/analytics/machine-learning.html](https://www.sas.com/en_us/insights/analytics/machine-learning.html) (accessed Apr. 14, 2019).
- [9] "What is Machine Learning? | Types of Machine Learning," *Edureka*, Mar. 06, 2018. <https://www.edureka.co/blog/what-is-machine-learning/> (accessed Jun. 24, 2019).

- [10] “What is Computer Vision? - Definition from Techopedia.” <https://www.techopedia.com/definition/32309/computer-vision> (accessed Apr. 14, 2019).
- [11] R. Klette, *Concise computer vision*. Springer, 2014.
- [12] “Wearable Technology and Wearable Devices: Everything You Need to Know,” *Wearable Devices*, Mar. 26, 2014. <http://www.wearabledevices.com/what-is-a-wearable-device/> (accessed Apr. 23, 2019).
- [13] “What is a web service?,” Feb. 09, 2019. undefined (accessed Apr. 14, 2019).
- [14] D. M. Rathod, M. S. Dahiya, and S. M. Parikh, “Towards composition of RESTful web services,” in *2015 6th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, Jul. 2015, pp. 1–6, doi: 10.1109/ICCCNT.2015.7395237.
- [15] A. Silberschatz, H. F. Korth, S. Sudarshan, and others, *Relational Databases*, 6th ed., vol. 4. McGraw-Hill New York, 1997.
- [16] A. Silberschatz, H. F. Korth, S. Sudarshan, and others, “Introduction to the Relational Model,” in *Database system concepts*, 6th ed., vol. 4, McGraw-Hill New York, 1997, p. 39.
- [17] A. Chebotko, A. Kashlev, and S. Lu, “A Big Data Modeling Methodology for Apache Cassandra,” in *2015 IEEE International Congress on Big Data*, Jun. 2015, pp. 238–245, doi: 10.1109/BigDataCongress.2015.41.
- [18] “What is Apache Cassandra,” *DataStax Academy: Free Cassandra Tutorials and Training*, Apr. 11, 2019. <https://academy.datastax.com/planet-cassandra/what-is-apache-cassandra> (accessed Apr. 10, 2019).
- [19] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, “Spark: Cluster computing with working sets.,” *HotCloud*, vol. 10, no. 10–10, p. 95, 2010.
- [20] “OpenCV library.” <https://opencv.org/> (accessed Apr. 14, 2019).

- [21] “The Java Language Environment.” <https://www.oracle.com/technetwork/java/intro-141325.html> (accessed Apr. 14, 2019).
- [22] “Introduction,” *Scala Documentation*. </tour/tour-of-scala.html> (accessed May 13, 2020).
- [23] “The Python Tutorial — Python 3.7.3 documentation.” <https://docs.python.org/3/tutorial/> (accessed Apr. 14, 2019).
- [24] “Node.js Introduction.” [https://www.w3schools.com/nodejs/nodejs\\_intro.asp](https://www.w3schools.com/nodejs/nodejs_intro.asp) (accessed Apr. 14, 2019).
- [25] “Fitbit Inc (FIT.N) Company Profile | Reuters.com.” <https://www.reuters.com/finance/stocks/company-profile/FIT.N> (accessed Apr. 14, 2019).
- [26] “Acerca de nosotros.” <http://orizaba.tecnm.mx/index.php/acerca-de-nosotros/49-acerca-de-nosotros> (accessed Apr. 14, 2019).
- [27] “Instituto Tecnológico de Orizaba.” <http://orizaba.tecnm.mx/> (accessed Apr. 14, 2019).
- [28] A. Belle, R. Thiagarajan, S. M. R. Soroushmehr, F. Navidi, D. A. Beard, and K. Najarian, “Big Data Analytics in Healthcare,” *BioMed Research International*, vol. 2015, p. 16, 2015.
- [29] V. Palanisamy and R. Thirunavukarasu, “Implications of big data analytics in developing healthcare frameworks – A review,” *Journal of King Saud University - Computer and Information Sciences*, Dec. 2017, doi: 10.1016/j.jksuci.2017.12.007.
- [30] B. Dinesen *et al.*, “Personalized Telehealth in the Future: A Global Research Agenda,” *J Med Internet Res*, vol. 18, no. 3, p. e53, Mar. 2016, doi: 10.2196/jmir.5257.
- [31] E. Ahmed *et al.*, “The role of big data analytics in Internet of Things,” *Computer Networks*, vol. 129, pp. 459–471, 2017, doi: <https://doi.org/10.1016/j.comnet.2017.06.013>.

- [32] M. Angelova, J. Ellman, H. Gibson, P. Oman, S. Rajasegarar, and Y. Zhu, "User activity pattern analysis in Telecare Data," *IEEE access*, vol. 6, pp. 33306–33317, 2018.
- [33] F. C. Nicolalde, F. Silva, B. Herrera, and A. Pereira, "Big Data Analytics in IOT: Challenges, Open Research Issues and Tools," in *World Conference on Information Systems and Technologies*, 2018, pp. 775–788.
- [34] D. V. Dimitrov, "Medical Internet of Things and Big Data in Healthcare," *Healthcare informatics research*, vol. 22, no. 3, pp. 156–163, Jul. 2016, doi: 10.4258/hir.2016.22.3.156.
- [35] M. Islam, M. Hasan, X. Wang, H. Germack, and others, "A systematic review on healthcare analytics: Application and theoretical perspective of data mining," in *Healthcare*, 2018, vol. 6, p. 54.
- [36] D. C. Yacchirema, D. Sarabia-Jácome, C. E. Palau, and M. Esteve, "A Smart System for Sleep Monitoring by Integrating IoT With Big Data Analytics," *IEEE Access*, vol. 6, pp. 35988–36001, 2018, doi: 10.1109/ACCESS.2018.2849822.
- [37] C. H. Lee and H.-J. Yoon, "Medical big data: promise and challenges," *Kidney research and clinical practice*, vol. 36, no. 1, p. 3, 2017.
- [38] Y. N. Malek *et al.*, "On the use of IoT and Big Data Technologies for Real-time Monitoring and Data Processing," *Procedia Computer Science*, vol. 113, pp. 429–434, Jan. 2017, doi: 10.1016/j.procs.2017.08.281.
- [39] H. Dubey, J. Yang, N. Constant, A. M. Amiri, Q. Yang, and K. Makodiya, "Fog data: Enhancing telehealth big data through fog computing," in *Proceedings of the ASE bigdata & socialinformatics 2015*, 2015, p. 14.
- [40] C. Lee *et al.*, "Big Healthcare Data Analytics: Challenges and Applications," in *Handbook of Large-Scale Distributed Computing in Smart Healthcare*, S. U. Khan, A.

Y. Zomaya, and A. Abbas, Eds. Cham: Springer International Publishing, 2017, pp. 11–41.

[41] D. Borsos, “Data Analytics using Cassandra and Spark,” *OpenCredo*, Mar. 23, 2017. <https://opencredo.com/blogs/data-analytics-using-cassandra-and-spark/> (accessed Apr. 16, 2019).

[42] “Raspbian - Raspberry Pi Documentation.” <https://www.raspberrypi.org/documentation/raspbian/> (accessed Dec. 08, 2019).

[43] “1.10. Decision Trees — scikit-learn 0.22.2 documentation.” <https://scikit-learn.org/stable/modules/tree.html#tree-algorithms-id3-c4-5-c5-0-and-cart> (accessed May 06, 2020).

[44] “Good, Fair, or Poor: How Well Do You Sleep? - National Sleep Foundation.” <https://www.sleepfoundation.org/articles/good-fair-or-poor-how-well-do-you-sleep-0> (accessed Nov. 27, 2019).

[45] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, 2001, vol. 1, p. I–I.

[46] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 2005, vol. 1, pp. 886–893 vol. 1.

[47] “Linear Methods - RDD-based API - Spark 2.4.5 Documentation.” <http://spark.apache.org/docs/latest/mllib-linear-methods.html#linear-support-vector-machines-svms> (accessed May 15, 2020).

[48] “Linear Methods - RDD-based API - Spark 2.4.5 Documentation.” <http://spark.apache.org/docs/latest/mllib-linear-methods.html#logistic-regression> (accessed May 15, 2020).

- [49] D. Faraggi and B. Reiser, “Estimation of the area under the ROC curve,” *Statistics in Medicine*, vol. 21, no. 20, pp. 3093–3106, 2002, doi: 10.1002/sim.1228.
- [50] J. Fan, S. Upadhye, and A. Worster, “Understanding receiver operating characteristic (ROC) curves,” *Canadian Journal of Emergency Medicine*, vol. 8, no. 1, pp. 19–20, 2006, doi: 10.1017/S1481803500013336.
- [51] “Precision-recall curves – what are they and how are they used?” <https://acutecaretesting.org/en/articles/precision-recall-curves-what-are-they-and-how-are-they-used> (accessed Sep. 23, 2020).
- [52] C. Sammut and G. I. Webb, “Precision and Recall,” in *Encyclopedia of machine learning*, Springer Science & Business Media, 2011, p. 781.
- [53] T. Saito and M. Rehmsmeier, “The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets,” *PloS one*, vol. 10, no. 3, p. e0118432, 2015.