

**Universidad Internacional de La Rioja
Máster universitario en Ingeniería de Software y
Sistemas Informáticos**

Sistema de registro de entrada de empleados por medio de detección facial

Trabajo Fin de Máster

Presentado por: Prieto Rodriguez, Carlos Arturo

Director/a: De Olea Moreta, Cinthia

Resumen

El objetivo principal del trabajo es poder diseñar y construir un sistema de reconocimiento facial que apoye a las organizaciones en poder recolectar la información del registro de entradas y salidas de los empleados a sus oficinas o plantas de trabajo. Además por medio del reconocimiento de sentimientos se podrá dar una valiosa asistencia las organizaciones en la recolección de información sobre el clima laboral para llevar estadísticas que permitan implementar acciones de mejora.

Para la realización de este sistema se implementó la metodología scrum la cual permite avanzar de manera rápida gracias a sus prácticas ágiles de desarrollo de software asegurando la calidad y fiabilidad del mismo.

Como conclusión se determinó que el desarrollo de los sistemas de aprendizaje de máquina ya tienen un nivel suficiente de madurez con el cual podremos involucrarlos en casos reales y con ello enriquecer los sistemas dando apoyo a nuestras organizaciones.

Palabras Clave: Reconocimiento facial, aprendizaje de máquina, redes neuronales, aplicación práctica

Abstract

The main objective of the work was designed and construct a facial recognition system that supports organizations in the purpose of collecting information related with entrances and exits of employees to its offices or workplaces. In addition, through feelings recognition, organizations can be given valuable assistance in collection of information on the work climate to carry statistics that allow the implementation of improvement actions.

For the implementation of this system, Scrum methodology was used, which allows rapid progress thanks to its agile software development practices, ensuring its quality and reliability.

In conclusion, it was determined that the development of machine learning systems already has a sufficient level of maturity to which we can involve them in real cases and thereby enrich the systems giving support to our organizations.

Keywords: Facial recognition, machine learning, neural networks, practical application.

Índice de contenidos

RESUMEN	1
ABSTRACT	2
ÍNDICE DE CONTENIDOS	3
ÍNDICE DE TABLAS	6
ÍNDICE DE FIGURAS.....	7
1. INTRODUCCIÓN	9
1.1 JUSTIFICACIÓN	10
1.2 PLANTEAMIENTO DEL TRABAJO.....	11
1.3 ESTRUCTURA DE LA MEMORIA.....	12
2. CONTEXTO Y ESTADO DEL ARTE	13
2.1 BREVE HISTORIA DE LA INTELIGENCIA ARTIFICIAL	13
2.2 APRENDIZAJE DE MÁQUINA.....	16
2.3 REDES NEURONALES ARTIFICIALES.....	17
2.4 MODELOS DE ANÁLISIS VISUAL	19
2.4.1 <i>Detección de objetos.....</i>	<i>19</i>
2.4.2 <i>Detección y reconocimiento facial.....</i>	<i>21</i>
2.6 LIBRERÍAS OPEN SOURCE PARA RECONOCIMIENTO FACIAL	25
2.6.1 <i>Pytorch.....</i>	<i>25</i>
2.6.2 <i>Tensor Flow.....</i>	<i>26</i>
2.6.3 <i>OpenCV.....</i>	<i>26</i>
2.6.4 <i>Open Face.....</i>	<i>26</i>
	3

2.7 APIs DE INTELIGENCIA ARTIFICIAL DE GOOGLE.....	26
3. OBJETIVOS CONCRETOS Y METODOLOGÍA DE TRABAJO	29
3.1. OBJETIVO GENERAL	29
3.2. OBJETIVOS ESPECÍFICOS	29
3.2. METODOLOGÍA DEL TRABAJO.....	30
3.2.1 Reuniones.....	31
3.2.2 Artefactos.....	31
3.2.3 Acuerdos	34
3.3.4 Elementos excluidos.....	36
4. HISTORIAS DE USUARIO.....	37
4.1 REGISTRO DE TIEMPO CON RECONOCIMIENTO FACIAL.....	37
4.2 DETECCIÓN DE ESTADO DE ÁNIMO	39
4.3 ALTA DE NUEVOS COLABORADORES	40
4.4 REPORTE DE INGRESO	43
4.5 REPORTE DE ESTADO DE ÁNIMO	45
5. REQUERIMIENTOS NO FUNCIONALES Y DEFINICIÓN DE ARQUITECTURA	46
5.1 REQUERIMIENTOS NO FUNCIONALES.....	46
5.2 VISTA LÓGICA.....	48
5.2.1 Modelo de datos	48
5.2 VISTA DE DESPLIEGUE	50
5.2.1 App Colaboradores.....	50
5.2.2 App de Administración	52
5.3 VISTA DE IMPLEMENTACIÓN	53
6. DESCRIPCIÓN DEL SISTEMA.....	54
6.1 ENTORNO DE DESARROLLO BACKEND	54

6.2 ENTORNO DE DESARROLLO FRONTEND.....	54
6.3 CAPTURAS DE LA APLICACIÓN.....	55
7. EVALUACIÓN.....	60
8. CONCLUSIONES Y LÍNEAS DE TRABAJO FUTURO.....	64
8.1 CONCLUSIONES.	64
8.2 LÍNEAS FUTURAS.....	65
9. BIBLIOGRAFÍA.....	66
ANEXOS	70
ARTÍCULO.....	70

Índice de tablas

<i>Tabla 1: Figura de descripcion de caso de uso.....</i>	<i>32</i>
<i>Tabla 2: Historia de usuario Registro de tiempo con reonomiento facial.....</i>	<i>37</i>
<i>Tabla 3: Historia de usuario Deteccion de estado de ánimo.....</i>	<i>39</i>
<i>Tabla 4: Historia de usuario de nuevos colaboradores.....</i>	<i>40</i>
<i>Tabla 5: Historia de usuario reporte de ingreso.....</i>	<i>43</i>
<i>Tabla 6: Historia de usuario Reporte de estado de ánimo.....</i>	<i>45</i>

Índice de Figuras

<i>Figura 1: Ejemplo de red neuronal artificial (Proyecto Perceptron Simple (Código C++), 2018)</i>	18
<i>Figura 2: Seis reglas para delinear el contorno (Object Detection by Contour Segment Networks)</i>	20
<i>Figura 3: Imagen de botella con contorno delimitado para su posible detencion. (Object Detection by Contour Segment Networks)</i>	20
<i>Figura 4: Función de evaluación del candidato (Object Detection by Contour Segment Networks)</i>	20
<i>Figura 5: Función para la consideración de la localización del objeto (Object Detection by Contour Segment Networks)</i>	21
<i>Figura 6:Proceso de deteccion de rostro (OpenFace, s.f.)</i>	21
<i>Figura 7: Mapeo de pixeles (A Tour of Machine Learning Algorithms, s.f.)</i>	23
<i>Figura 8: Foto de un ojo interpretada por una computadora (A Tour of Machine Learning Algorithms, s.f.)</i>	23
<i>Figura 9: Mapeo de características del rostro (A Tour of Machine Learning Algorithms, s.f.)</i>	24
<i>Figura 10: Puntos clave del rostro (A Tour of Machine Learning Algorithms, s.f.)</i>	24
<i>Figura 11: Rostro encontrado (A Tour of Machine Learning Algorithms, s.f.)</i>	25
<i>Figura 12: La metodología de trabajo a utilizar estará basada en el framework SCRUM (Proyectos agiles.org, s.f.),</i>	30
<i>Figura 13: Ley de pareto (Gestion de la Comunicación en la empresa, 2013)</i>	31
<i>Figura 14: Interfaz bienvenido</i>	38
<i>Figura 15: Interfaz Adios</i>	39
<i>Figura 16: Interfaz listado de usuarios</i>	42
<i>Figura 17: Interfaz Crear nuevo usuario</i>	42
<i>Figura 18: Interfaz registrar fotos</i>	43
<i>Figura 19: Interfaz reporte de ingresos y salidas por usuario</i>	44

<i>Figura 20: Interfaz gráfico de estado de animo</i>	46
<i>Figura 21: Diagrama de datos usuarios</i>	48
<i>Figura 22: Diagrama de datos eventos</i>	49
<i>Figura 23: Diagrama de despliegue de la app para colaborado</i>	50
<i>Figura 24: Diagrama de despliegue App de Administración</i>	52
<i>Figura 25: Diagrama de Componentes</i>	53
<i>Figura 26: Listado de usuarios</i>	55
<i>Figura 27: Boton nuevo usuario</i>	56
<i>Figura 28: Creación de usuario</i>	56
<i>Figura 29: Opciones de menú de usuario</i>	57
<i>Figura 30: Carga de imagenes</i>	57
<i>Figura 31: Cambio de idioma</i>	58
<i>Figura 32: Ingreso de usuario</i>	58
<i>Figura 33: Salida de usuario</i>	59
<i>Figura 34: Estadísticas pregunta No. 01</i>	61
<i>Figura 35: Estadísticas pregunta No. 02</i>	61
<i>Figura 36: Estadísticas pregunta No. 03</i>	62
<i>Figura 37: Estadísticas pregunta No. 04</i>	62
<i>Figura 38: Estadísticas pregunta No. 05</i>	63
<i>Figura 39: Estadísticas pregunta No. 06</i>	63

1. Introducción

El aprendizaje de máquina es una de las tecnologías que más ha evolucionado en el campo de la Inteligencia artificial. Gracias a esto se ha comenzado a dar grandes avances en algunos sectores como lo son el reconocimiento de fraudes en tarjetas de crédito, la conducción de carros autónomos, comprensión del lenguaje natural, y traducciones entre diferentes idiomas.

A pesar de todos los avances que se han hecho y de tener múltiples librerías, frameworks y diferentes tecnologías tanto de código libre como propietarias que facilitan el alcance del aprendizaje de máquina, aún es un concepto que pocas personas entienden, que otras tantas ven como poco práctico y que no se puede emplear en el día a día. Por otro lado hay gente que por su falta de conocimiento piensa que la inteligencia artificial es futurista y que solo se puede ver en la ciencia ficción.

El propósito de este trabajo es llevar esta tecnología a un caso práctico de la vida real, el cual puede ayudar a las compañías a mejorar y conocer el ambiente laboral de una manera mas practica que la elaboración de complejas encuestas en las cuales no se demuestra el verdadero estado de ánimo de los empleados.

Para concretar este caso práctico se creó una aplicación que aprovecha las APIs de inteligencia artificial ofrecidas por Google y utiliza una arquitectura serverless para reducir la carga administrativa de la solución.

1.1 Justificación

El problema que se va a tratar dentro del trabajo, es la complicación que tienen algunas empresas para llevar un correcto registro de las entradas y salidas de personal. En algunas de las Industrias manufactureras que tienen gran volumen de personal hay problemas en las entradas y en las salidas debido a que deben de llevar un registro manual haciendo muy demorado este proceso y teniendo retrasos en los comienzos de las labores de los trabajadores.

Para poder solucionar esta problemática es relevante poder realizar de manera automatizada un proceso con el cual poder saber la entrada y la salida de cada una de las personas de manera rápida y eficiente con lo cual se pueda llevar a cabo un proceso mucho más fácil quitando la complejidad de las mismas y solucionando problemas de retrasos en el comienzo de las jornadas laborales.

Por otro lado se pretende dar una herramienta adicional a las áreas de recursos humanos para poder hacer análisis de clima laboral. Debido a que en algunos casos los empleados no dicen su verdadero estado de ánimo dentro de la compañía por temor a represalias o porque las encuestas que tiene como fin entender el clima laboral de la industria, son muy engorrosas y se tiene la percepción de pérdida de tiempo cuando se llenan.

Por último se pretende llevar un caso concreto con el cual se puedan demostrar las capacidades del aprendizaje de máquina en las empresas y que se pueda dar a entender que la tecnología ya está suficientemente madura como para ser utilizada.

1.2 Planteamiento del trabajo

Para poder solucionar el inconveniente la propuesta que se está indicando en este trabajo es crear un sistema que se apoya en las nuevas tecnologías de redes neuronales artificiales y de aprendizaje de máquina para entrenar modelos de reconocimiento facial y con ello poder agilizar las entradas de los empleados en las fábricas. Adicionalmente como beneficio de lo que se está proponiendo se podrá tener un indicador de felicidad en las empresas a través de la utilización de dichos modelos de Inteligencia artificial con lo cual se tendrá estadísticas del clima laboral para poder tomar decisiones de mejorar el ambiente de los trabajadores y con ello tener un mejor rendimiento en sus actividades productivas.

Lo que se está proponiendo es un sistema automatizado en el cual las personas cuando vayan entrando a su trabajo se les tomara una foto la cual será procesada y a través de reconocimiento facial se indicará cuál es el trabajador que está entrando, además se podrá hacer una evaluación de sentimientos indicando si el trabajador está feliz triste o enojado. Todo esto se guardarán una base de datos para después poder realizar análisis a través de sistemas informáticos con los cuales podremos sacar estadísticas del clima laboral sabiendo a través de esta información cómo se sienten las personas y si están a gusto con su trabajo o no.

Por último se utilizaran servicios serverless (sin servidores), que permitan administrar la herramienta sin necesidad de preocuparse de la carga que representa el hardware, sistemas operativos entre otros. Dando al sistema capacidades de responder de manera elástica a un creciente número de peticiones y pudiendo enfocar los esfuerzos en el desarrollo del software.

1.3 Estructura de la memoria

En los siguientes capítulos comenzaremos a ver cómo a través de sistemas de Inteligencia artificial y redes neuronales vamos a poder solucionar el problema concreto que estamos proponiendo. Veremos un contexto general indicando el estado del arte de dichas herramientas. Para ello veremos cómo a través de estas tecnologías vamos logrando un modelo certero para reconocimiento facial y cómo se desarrollará este sistema. Indicaremos la arquitectura que se utilizará viéndola desde el punto de vista de procesos, despliegue y un punto de vista lógico. También veremos cómo va a funcionar a través de las historias de usuario y describiremos su funcionamiento a través de mockups con lo cual se mostrará la utilización de todos los sistemas relacionados y las herramientas de desarrollo, mostrando al final el estado en el cual queda el desarrollo y las posibles evoluciones para un ambiente productivo.

2. Contexto y estado del arte

Como parte de las tecnologías propuestas para el desarrollo de este trabajo se evalúa la utilización de Inteligencia artificial como medio de detección de rostros y de sentimientos. Por ello es necesario entender de manera clara algunos de los conceptos que están alrededor como lo son el aprendizaje de máquina, aprendizaje profundo y redes neuronales artificiales para llevarlos a un medio concreto y poderlos aplicar en la arquitectura que se propondrá.

En este trabajo no se pretende describir de manera detallada cada uno de los cálculos, ecuaciones y pasos en los algoritmos, sino que trata de mostrar de una manera general la complejidad para dar un contexto más cercano al lector.

2.1 Breve historia de la inteligencia artificial

Desde épocas remotas la humanidad siempre ha pensado en la automatización de procesos para la facilitación de sus labores cotidianas y con ello se ha dado paso a pensar en máquinas que resuelvan problemas por sí mismas llegando a visualizar a máquinas inteligentes. Algunos de estos casos son el coloso de Rodas (El coloso de Rodas, 2017), El autómatas que supuestamente jugó con Napoleón al ajedrez (Guijarro & Gonzalez, 2010) e incluso monstruos mágicos como el golem de praga (Golem: Jewish Magical and Mystical Traditions on the Artificial Anthropoid, 1990).

También algunos filósofos tales como Descartes (Gonzalez, 2011) han coqueteado con la idea de la inteligencia humana como un mecanismo. Pero no fue hasta el siglo 20 que se comenzó a hablar concretamente de este concepto en concreto.

Gracias a Alan turing y su “Artículo Maquinaria Computacional e Inteligencia” (Alan Turing - Maquina Computacional e Inteligencia) se comenzó a dar el debate acerca de la idea de máquinas inteligentes que puedan resolver tareas e incluso pueden tener conversaciones con personas. Actualmente en este artículo Alan turing quiso responder la pregunta ¿Las máquinas pueden pensar?. Otros aportes fundamentales de Alan turing fue el test de turing El cual proponía una metodología sencilla para Descubrir si una máquina tenía pensamiento o no. Este test consistía en que una persona hablará con una máquina a través de una Línea de comandos emulando un chat y si la máquina podía engañar a la persona haciéndola creer que estaba hablando Con otro ser humano esto era una indicación de que la máquina tenía pensamiento.

Otra persona que contribuyó a los inicios de la Inteligencia artificial fue John McCarthy (Villatoro, 2011) El cual acuñó la La expresión en 1956 y le dio una definición concreta la cual es “...la ciencia e ingenio de hacer máquinas inteligentes, especialmente programas de cómputo inteligentes” (Basic Questions, s.f.)

Otra de las mentes pensantes fundamentales para los inicios de la Inteligencia artificial fue John Von Neumann El cual tomó la estafeta después de la muerte de Alan turing y gracias al cual se comenzó a desarrollar sistemas computacionales con conceptos de la anatomía del cerebro Como por ejemplo la memoria o la analogía entre los sensores y los sentidos.

Tras Algunos años de avances lentos y decepciones en los años ochentas se comenzó a estipular un concepto japonés el cual se denominaba quinta generación de computadoras. (Fifth Generation Computer Systems, s.f.) Este concepto se fundamenta va en un desarrollo de ordenadores con un enfoque de Inteligencia artificial, utilizando lenguajes de programación tales como Prolog para escribir los programas. Algunas de las ideas sobre esta generación indicaban que iba a ser muy sencillo de realizar traducciones automáticas entre lenguajes tales como japonés e inglés. Después de 11 años de producción se abandonó este proyecto debido a que no alcanzaron los resultados esperados por problemas en la paralelización de tareas y bajas en el desempeño y rendimiento.

En los años 90's se establecen los fundamentos de los agentes inteligentes y comienzan a surgir los primeros. Pero no es hasta el siglo 21 que comienzan a dar frutos todas estas investigaciones con los primeros programas informáticos que pueden ganar en algunos de los más complejos juegos de la humanidad tales como Jeopardy (IBM Watson: The inside story of how Jeopardy-winning supercomputer was born, and what it wants to do next, s.f.) y más recientemente Google ganando el milenario juego llamado Go (The Story of AlphaGo so Far, s.f.).

En estos últimos años se ha dado un auge de la Inteligencia artificial con la salida de los asistentes de voz (Prieto, s.f.) la liberación como software libre Del algoritmo de redes neuronales de Google llamado tensorflow (An open source machine learning library for research and production , s.f.) y las múltiples librerías que ofrecen los diferentes actores en el mundo de la computación en la nube con lo cual se está viendo un despliegue de esta tecnología comenzando a verse aplicado en soluciones del mundo real.

2.2 Aprendizaje de máquina

Aprendizaje automático o aprendizaje de máquina es una rama de la inteligencia artificial que pretende que los programas computacionales puedan aprender con lo cual podrán generar conocimiento y con este aprendizaje resolver problemas.

Como resultado del aprendizaje de máquina se obtienen modelos los cuales se utilizan posteriormente para el análisis de información algunos de estos modelos son modelos geométricos, modelos probabilísticos o modelos lógicos.

Los algoritmos utilizados para el aprendizaje de máquina, Están clasificados en los siguientes métodos (A Tour of Machine Learning Algorithms, s.f.):

- **Aprendizaje supervisado** (Aprendizaje Supervisado y no Supervisado, s.f.): Este tipo de aprendizaje consiste en que los datos del entrenamiento Están ordenados por pares y la salida es una función que permite clasificar o etiquetar objetos que están relacionados a las entradas.
- **Aprendizaje no supervisado:** En este tipo de aprendizaje No saquen una organización por etiquetas Por lo cual el sistema debe estar en la capacidad de poder entender los patrones para generar dichas etiquetas de manera automática.

- **Aprendizaje semi supervisado:** La combinación de los dos métodos anteriores con lo cual se logra obtener información a través de los datos que están marcados y los que no están marcados.
- **Aprendizaje por refuerzo:** Es un algoritmo que aprende a través de ensayo de obteniendo su información A través de la observación del mundo que lo rodea.

2.3 Redes neuronales artificiales

En 1943 Warren McCulloch (Un genio vagabundo amante de la lógica, el padre incomprendido de la cibernética, s.f.) y Walter Pitts (Geftter, 2015) Crearon un modelo matemático para computadora el cual se basaba en las redes neuronales para resolver problemas complejos. Estos modelos fueron la base de una de las disciplinas de la Inteligencia artificial con mayor repercusión y que ha dado mayores éxitos la cual es denominada redes neuronales artificiales.

Las redes neuronales artificiales son un conjunto de modelos matemáticos los cuales están divididos en Unidades neuronales sin las cuales emulan el funcionamiento de una neurona Comportándose de una manera parecida a sus análogas biológicas conectándose por medio de axones. Cada unidad está conectada a muchas otras de las mismas y ejecuta operaciones básicas que conjunto terminan siendo un modelo más complejo el cual puede representar conocimiento. Estas redes transmiten información de adelante hacia atrás logrando de esta manera poder procesar datos por medio de varias capas Como se muestra en el siguiente gráfico.

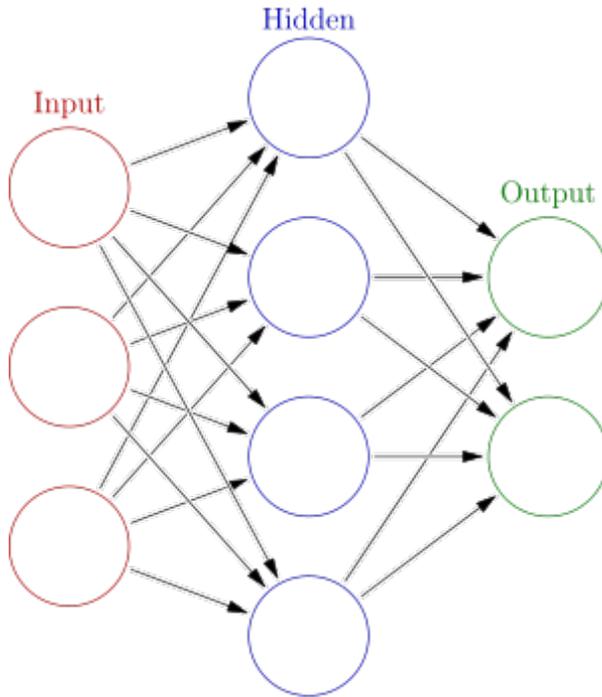


Figura 1: Ejemplo de red neuronal artificial (Proyecto Perceptron Simple (Código C++), 2018)

El objetivo de una red neuronal es poder resolver los problemas de una manera muy similar a como lo hace el cerebro humano salvando las distancias debido a la complejidad de las redes biológicas intrínsecas en el cerebro de la gente.

Gracias al avance de los últimos años en estas tecnologías se ha echado a lograr sistemas de reconocimiento de voz, lenguaje natural e incluso reconocimiento de imágenes. El aspecto más interesante a resaltar es su capacidad de autoaprendizaje y de reconocimiento de patrones por el cual se Está realizando aplicaciones a nivel estadístico y de análisis de datos.

El funcionamiento básico (Artificial Neural Networks as Models of Neural Information Processing, s.f.) de una red está dado por sus conexiones entre las diferentes neuronas un modelo básico podría estar dado por conexiones de tres niveles En los cuales el primer nivel Sería el nivel de entrada Los cuales envían la información a la segunda capa a través de la sinapsis Y luego a través de la sinapsis se enviar a la tercera capa que sería la capa de salida. Los sistemas más complejos tendrán una mayor cantidad de capas que permiten un mejor procesamiento de la información.

2.4 Modelos de análisis visual

Antes de realizar el reconocimientos de rostros hay que realizar una serie de pasos anteriores los cuales constituyen la ruta para generar una modelo aprendizaje de máquina el cual pueda ayudarnos con nuestra labor.

2.4.1 Detección de objetos

El primero de ellos es la detección de objetos (Object Detection by Contour Segment Networks) la cual consiste en que nuestro modelo simplifique las imágenes en una sucesión de líneas que son las que delinean el contorno de los diferentes objetos. Para este propósito es necesario poder verificar el contraste de las imágenes y ver los cambios de colores que nos permitirán generar una sucesión de líneas con las cuales se podrá generar el contorno.

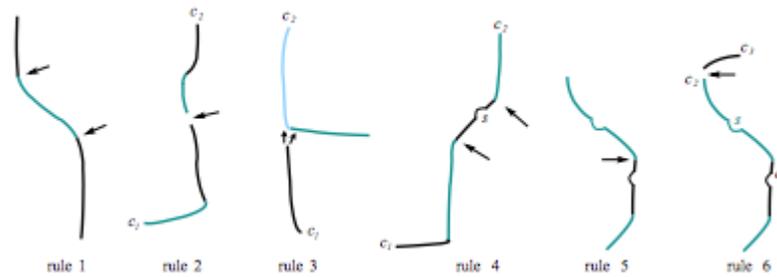


Figura 2: Seis reglas para delinear el contorno (Object Detection by Contour Segment Networks)



Figura 3: Imagen de botella con contorno delimitado para su posible detencion. (Object Detection by Contour Segment Networks)

Para poder realizar las posibles detecciones es fundamental aplicar un algoritmo basado en llamado “The basic matching algorithm”, el cual realiza cálculos complejos matemáticos para poder generar los contornos. Algunos de los cálculos matemáticos que se emplean son:

$$q_c = q(m, c, \mathcal{P}) = w_{l_a} D_{l_a}(m, c, \mathcal{P}) + w_{l_d} D_{l_d}(m, c, \mathcal{P}) + w_{\theta} D_{\theta}(m, c)$$

Figura 4: Función de evaluación del candidato (Object Detection by Contour Segment Networks)

$$D_{ta}(m, c, \mathcal{P}) = \frac{1}{|\mathcal{P}|} \sum_{\{m_i, t_i\} \in \mathcal{P}} D_{\theta}(mm_i, \vec{ct}_i)$$

Figura 5: Función para la consideración de la localización del objeto (Object Detection by Contour Segment Networks)

2.4.2 Detección y reconocimiento facial

Un caso específico de la detección de objetos es la detección de rostros la cual es una tarea en la que específicamente se buscan rostros humanos. Este algoritmo se centra en la detección frontal de rostros por medio de los puntos del contorno de la cara, ojos cejas y nariz.

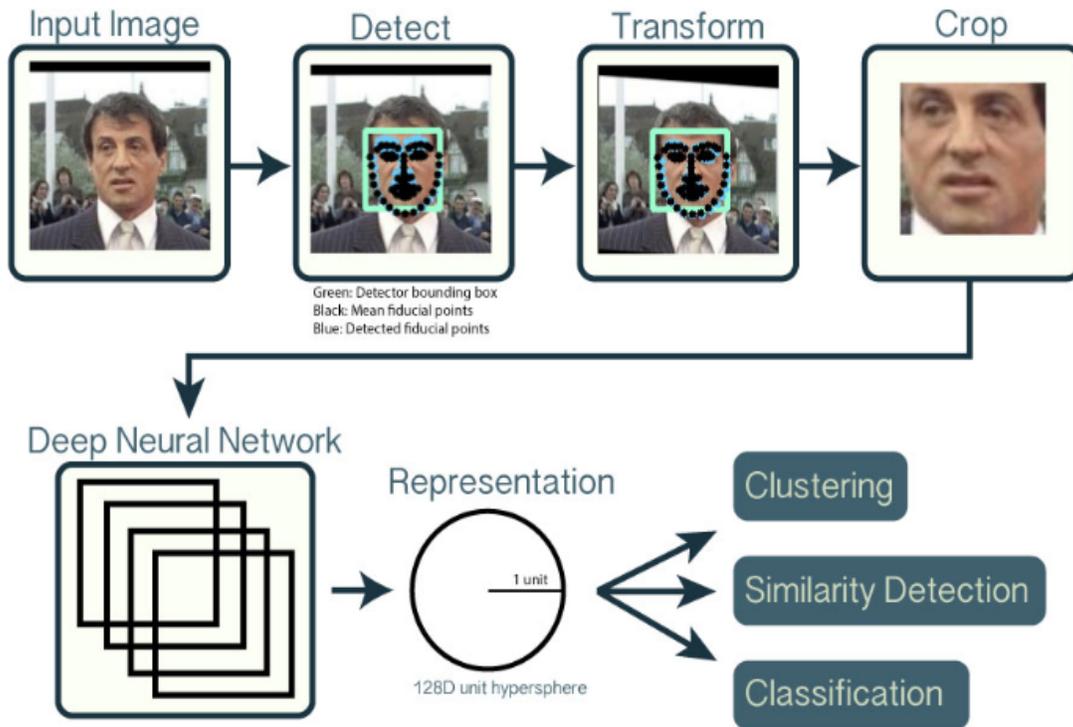


Figura 6: Proceso de detección de rostro (OpenFace, s.f.)

Pero la detección de rostros es solo el principio de la solución que se quiere implementar. Adicionalmente hay que realizar el reconocimiento lo cual permite identificar individuos por sus características únicas para poder realizar la diferenciación entre múltiples personas. Algunos de los problemas más comunes son (Geitgey, 2016):

1. Encontrar en la foto los rostros
2. Entender que si la persona está con la cara en otra dirección o incluso hay una mala iluminación en la foto aun sigue siendo la misma persona.
3. Ser capaz de diferenciar esas características únicas con las cuales se puede diferenciar un rostro.
4. Finalmente poder comparar esas características con una base de datos de los rostros almacenados para poder realizar la detección adecuada.

Por esto es que se necesita un gran poder de cómputo y el apoyo de los redes neuronales artificiales para poder realizar esta tareas.

Es necesario poder entender pixel por pixel de la foto para poder realizar un mapeo de la foto en números y con ello realizar la detección de las características

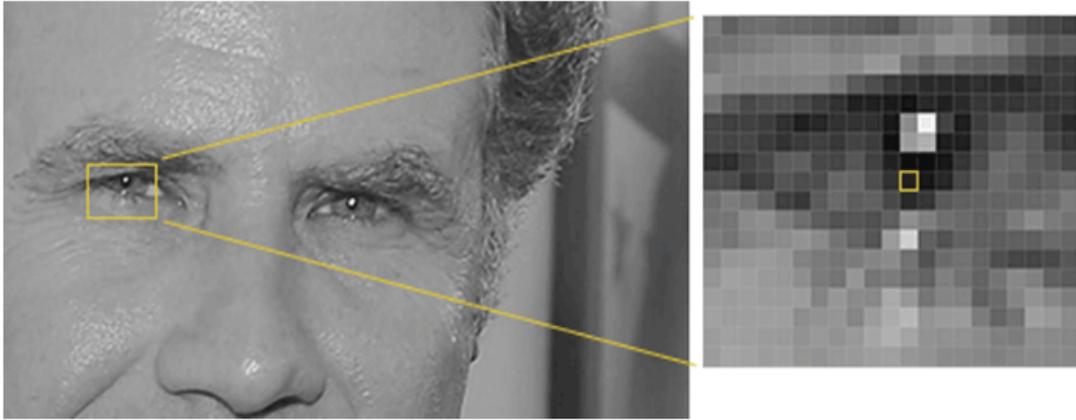


Figura 7: Mapeo de pixeles (A Tour of Machine Learning Algorithms, s.f.)

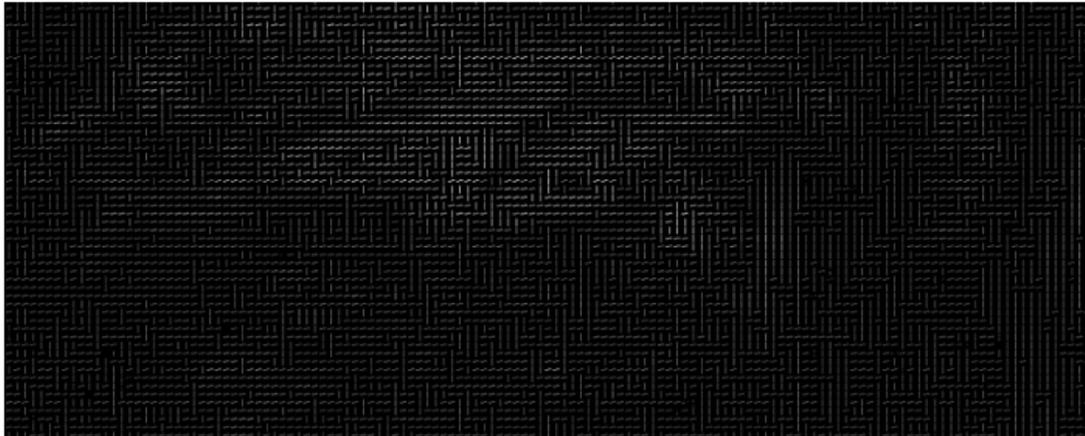


Figura 8: Foto de un ojo interpretada por una computadora (A Tour of Machine Learning Algorithms, s.f.)



Figura 9: Mapeo de características del rostro (A Tour of Machine Learning Algorithms, s.f.)

Una vez realizado el mapeo el algoritmo enumera cada uno de los puntos relevantes en el rostro y hace un análisis de características claves como la separación de las cejas, distancia entre la nariz y los ojos entre otras.

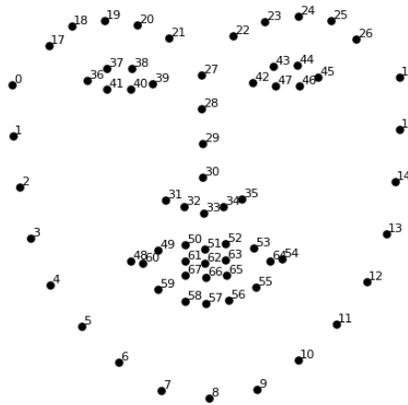


Figura 10: Puntos clave del rostro (A Tour of Machine Learning Algorithms, s.f.)

Con estos datos se puede realizar una comparación matemática para que estadísticamente se pueda dar una probabilidad de semejanza con los rostros almacenados en el modelo pre entrenado.

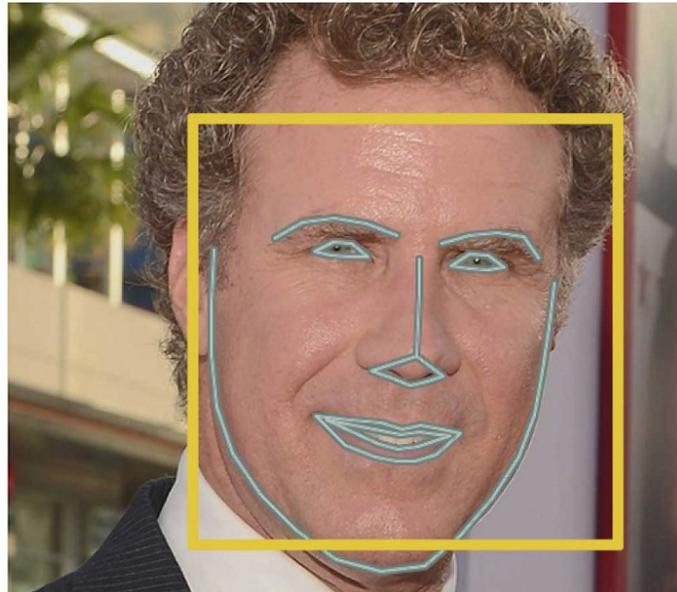


Figura 11: Rostro encontrado (A Tour of Machine Learning Algorithms, s.f.)

2.6 Librerías open source para reconocimiento facial

Para realizar las tareas de reconocimiento facial existen multiples librerias open source que permiten realizar aprendizaje de máquina de manera genérica como las cuales se mostraran a continuación:

2.6.1 Pytorch

Librería de python que permite realizar entrenamiento por medio de algoritmos de aprendizaje de máquina gracias a la aceleración por medio de GPU. [\(Pytorch, s.f.\)](#)

2.6.2 Tensor Flow

La importancia de esta librería radica en que es una herramienta para aprendizaje de máquinas pasado en redes neuronales artificiales la cual fue desarrollada por Google y liberada como software libre en el año 2015.

Esta librería es la que está detrás de la inteligencia de las búsquedas de Google, el asistente que está embebido en los dispositivos Android, la clasificación de imágenes, los sistemas de reconocimiento en YouTube e incluso en el sistema que ganó al campeón mundial de Go. ([TensorFlow](#), s.f.)

2.6.3 OpenCV

Librería desarrollada en C++ que permite realizar computación visual y que está enfocada en el reconocimiento visual de patrones, formas y objetos. Es altamente utilizada en robótica para dar habilidad de reconocimiento. ([OpenCV](#), s.f.)

2.6.4 Open Face

Librería para python que se apoya en OpenCV para realizar detección de rostros y reconocimiento facial. ([OpenFace](#), s.f.)

2.7 APIs de inteligencia artificial de Google

Dentro del catálogo de servicios que tiene Google en su plataforma de nube se encuentra una

categoría enfocada a temas de aprendizaje de máquina los cuales están contemplados para personas con diferentes habilidades que van desde los programadores hasta los científicos de datos con experiencia en inteligencia artificial.

En el campo de las ciencias de datos con herramientas especializadas Google nos ofrece la posibilidad de utilizar un servicio llamado Machine Learning Engine con el cual podemos entrenar y realizar predicciones con modelos basados en Tensorflow sin la necesidad de crear y administrar un cluster ya que es un servicio de tipo PAAS (SearchCloudComputing, s.f.) el cual da la posibilidad de utilizar los nuevos chips denominados TPUs (Armasu, 2016) que están diseñados como una fusión entre GPU y CPU y permite crear redes neuronales de una manera más sencilla.

Del lado de las APIs Google tiene toda una gama de servicios los cuales son accesibles de manera programática y brindan el potencial de dotar con capacidades de inteligencia artificial a los programas sin la necesidad de ser un experto en ciencias de datos. Entre estas APIs se encuentran:

- **Cloud Speech** que permite convertir un audio en texto.
- **Cloud Text-to-Speech** que tiene la capacidad de convertir un texto en voz.
- **Cloud Natural Language** descubre la estructura y el significado del texto mediante potentes modelos de aprendizaje automático en una API REST fácil de usar.
- **Translation** ofrece una interfaz de programación sencilla para traducir una cadena arbitraria a cualquier idioma compatible
- **Cloud Video Intelligence** facilita la búsqueda y el descubrimiento de vídeos mediante la extracción de metadatos, la identificación de palabras clave y la anotación de contenidos.

- **Cloud Vision** clasifica rápidamente imágenes en miles de categorías (por ejemplo, "barco de vela" o "torre Eiffel"), detecta objetos y caras por separado dentro de ellas e incluso encuentra y lee las palabras que contienen.

Y en el caso de necesitar una herramienta un poco más personalizada que haga provecho de los modelos pre entrenados de las APIs antes mencionadas pero que brinde la capacidad de construir y re entrenar para dar un mayor potencial en los casos que no están contemplados por estas APIs se encuentra AutoML que cuenta con 3 diferentes tipos que están enfocados en Natural Language, Translate, Vision. Con respecto a AutoML Vision permite aprovechar la tecnología de reconocimiento de imágenes propiedad de Google para preparar modelos de alta calidad gracias al aprendizaje por transferencia de última generación de Google y la tecnología de búsqueda con arquitectura neuronal.

3. Objetivos concretos y metodología de trabajo

Para este trabajo se plantearon unos objetivos concretos que ayudaron a tener una visión de dónde se llegará con el desarrollo del mismo los cuales serán planteados a continuación.

3.1. Objetivo general

Modelar e implementar una aplicación que permita llevar el control del ingreso de los empleados a sus lugares de trabajo por medio de detección facial y a demás capturar estadísticas sobre el estado de ánimo de los empleados por medio de redes neuronales artificiales que permitan la detección de sentimientos y con ello poder evaluar el clima laboral.

3.2. Objetivos específicos

- Investigar las tecnologías de aprendizaje de máquina para generar una aplicación práctica
- Indagar a fondo los nuevos servicios de análisis de imágenes basados en redes neuronales para detección de sentimientos.
- Modelar una arquitectura en la nube para la aplicación que permita tener un servicio escalable.
- Diseñar un data warehouse para poder realizar el análisis estadístico de los datos recolectados por la aplicación.
- Ofrecer una solución accesible a empresas para agilizar los procesos de ingreso a sus instalaciones.

3.2. Metodología del trabajo

Omitiendo algunos de sus elementos debido a que este marco se centran mucho en el trabajo en equipo y este proyecto se realizará de manera individual. (Oufaska, 2018)

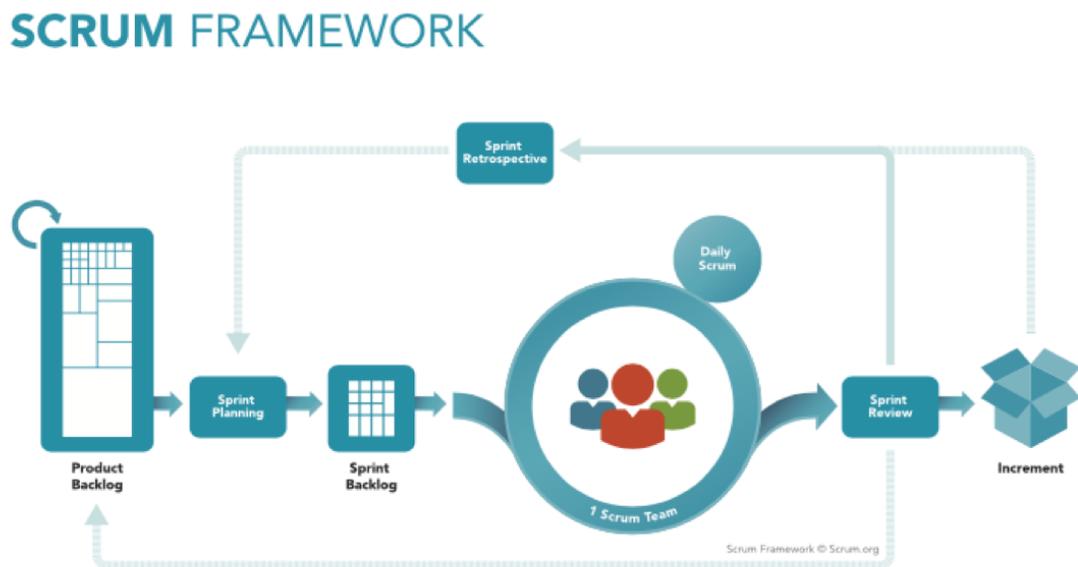


Figura 12: La metodología de trabajo a utilizar estará basada en el framework SCRUM (Proyectos agiles.org, s.f.),

3.2.1 Reuniones

Planeación: Reunión que se realiza al principio del sprint la cual se divide en dos partes, en la primera el Product Owner indica cuales son las historias de usuarios prioritarias para realizar y en la segunda parte el equipo valida los pormenores técnicos de cada historia.

Revisión de producto: Reunión que se realiza al finalizar el sprint y cuyo objetivo es la validación del incremento funcional por parte de los Stakeholders.

3.2.2 Artefactos

Product Backlog: Lista de historias de usuario que representan los requerimientos funcionales de la aplicación. Esta lista es vigilada, ordenada y priorizada por el Product Owner. Para la priorización se tiene en cuenta la ley de pareto la cual dice que con el 20% de las funcionalidades se resuelve el 80% de las necesidades de los clientes.

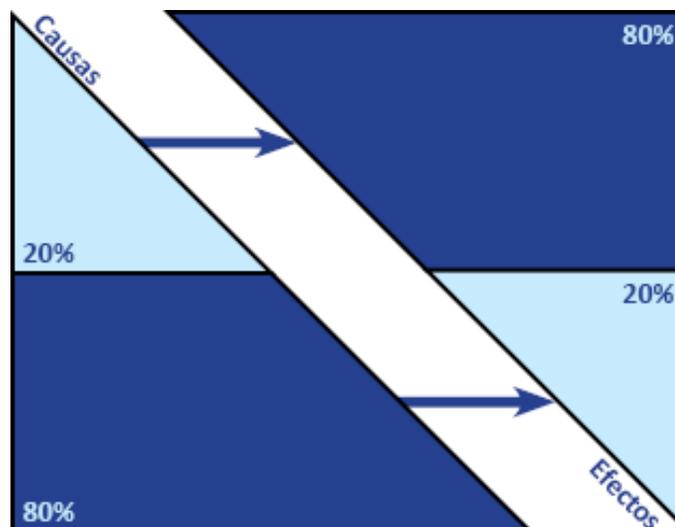


Figura 13: Ley de pareto (*Gestion de la Comunicación en la empresa, 2013*)

Sprint Backlog: Lista de tareas que se realizarán en los 15 días que comprende el sprint.

Historia de usuario: representación de un requisito escrito en una o dos frases utilizando el lenguaje común del usuario. Para la metodología implementada las historias de usuario son representación de requerimientos funcionales que además de ser descritos en una frase deben cumplir con las siglas de INVEST:

Independiente

Negociable

Valorable

Estimable

Small (Pequeña)

Testeable

Para la definición de las historias de usuario además se definió una carta o tarjeta la cual veremos a continuación:

Tabla 1: Figura de descripción de caso de uso

Actor:				
Descripción:				
Criterios de aceptación:				
#	Dado	Cuando	Entonces	Notas /

Escenario	que			Reglas de Negocio
1				
2				
n				

También se definieron una serie de acuerdos para saber cuando una historia está lista para ser trabajada o está lista para integrarse al incremento funcional, estos acuerdos están consignados en los documentos de DoR(Definición de listo) y DoD(Definición de hecho) que veremos más adelante.

Para documentar visualmente las historias de usuario se utilizan los wireframes los cuales indican una aproximación gráfica (sin llegar a ser la final) de cómo se verán las interfaces relacionadas. Para este trabajo se utilizará la metodología de Google para realizar prototipado (Google for Entrepreneurs, 2016), la cual utiliza bosquejos en papel y lápiz para poder avanzar rápidamente en la generación de ideas.

Incremento funcional: Resultado de cada sprint o iteración, este puede ser potencialmente entregable.

3.2.3 Acuerdos

DoR(Definición de listo): Describe las cosas que se deben tener una historia de usuario para estar lista para ser desarrollada.

Entendimiento del alcance: El equipo completo entiende lo más claramente posible el alcance funcional y no funcional de la historia, esto requiere que el dueño del producto haya explicado en detalle la funcionalidad esperada en la historia de usuario y sus consideraciones.

Las tareas (tickets) que componen la historia se han identificado y se han estimado: Toda Historia de Usuario contiene una serie de tareas para su desarrollo, que deben estar claramente identificadas y estimadas por el equipo, esto incluye las tareas para la generación de pruebas unitarias, de integración y de interfaz así como la actualización del documento y guía del producto.

La estimación de la historia de usuario debe ser de menos de 40 horas: Si una historia de usuario sobrepasa este número de horas es porque es demasiado grande por lo tanto debería partir en varias historias para que sean mas atómicas.

La historia de usuario tiene definidos los criterios de aceptación:La lista de los requisitos mínimos para evaluar una historia de usuarios y así saber si cumple con todas las especificaciones técnicas y de negocio, está escrita dentro la descripción.

Wireframes y modelos UML aprobados por el dueño del producto: Los prototipos de diseño de la interfaz de usuario, sean wireframes o mockups han sido revisados en detalle por el dueño del producto quién da el visto bueno o sugiere cambios y ajustes. En caso de no tener una interfaz para los usuarios finales, se puede esquematizar en un diagrama. Dichos prototipos de diseño se pueden esquematizar en una hoja de papel, en un tablero o digitalmente.

DoD(Definición de hecho): Describe las cosas que se deben tener una historia de usuario para estar finalizada.

Las tareas (tickets) que componen la historia se han marcado como cerrados: Todas las tareas de la historia de usuario, realizadas durante el hito, deben estar completadas y marcadas como cerradas, antes de poder marcar la misma historia como cerrada.

Código fuente de la historia de usuario debe estar en el repositorio: Una vez el ingeniero de desarrollo termine de escribir y probar el código, debe descargar cualquier cambio en el repositorio, realizar la integración de los cambios, probar que todo esté funcionando y finalmente subir todo al repositorio.

Contempla la actualización de datos existentes resultantes de cambios de esquema: en caso que el desarrollo de la historia tenga un cambio en el modelo de datos es necesario que se codifique la actualización de datos como se indica en la sección de actualización masiva de datos del sitio de desarrollo

Documentación en el manual: Cada funcionalidad desarrollada dentro de la historia de usuario debe contar con su correspondiente documentación completa en el manual del producto.

Changelog del sprint actualizado: En las notas del Sprint se deben escribir las funcionalidades que fueron agregadas por la historia de usuario en la última versión de la aplicación. Las funcionalidades deben ser concretas y escritas desde el punto de vista del usuario.

Cumple los criterios de aceptación definidos en la historia: Se deben ejecutar de manera manual las pruebas necesarias para hacer la verificación de la lista de criterios de aceptación tanto en ambiente de desarrollo como en ambiente de pruebas (staging). Una vez terminadas las pruebas se debe dejar un comentario en la historia de usuario para registrar si las pruebas fueron exitosas o no.

Revisión de todo lo anterior: tanto el código, las pruebas, la documentación y todo lo descrito anteriormente debe estar completado y debe tener el visto bueno de un miembro del equipo, quién es el único que puede dar la historia como completada y cerrada.

3.3.4 Elementos excluidos.

Refinamiento del Backlog: Es una reunión en la cual se verifican las historias de usuario que vendrían para los siguientes sprints para volverlas más atómicas y con ello más fácil de estimar.

Reunión diaria: Reunión que se realiza todos los días al comienzo de la jornada para ver el avance del equipo. En esta reunión cada miembro debe responder a tres preguntas. Que hizo el día de ayer?, Que va a hacer el día de hoy?, hubo algún problema u obstáculo?.

Retrospectiva: Reunión que intenta ver las cosas positivas y negativas del sprint para poder aprender y mejorar el proceso.

4. Historias de usuario

En este capítulo veremos cómo a través de las historias de usuario se definen los requerimientos funcionales los cuales servirán de base para poder desarrollar la aplicación y son una guía para las diferentes vistas del sistema.

4.1 Registro de tiempo con reconocimiento facial

Tabla 2: Historia de usuario Registro de tiempo con reconocimiento facial

Actor: Colaborador				
Descripción: Como Colaborador necesito que mi rostro sea reconocido por el sistema de autenticación para registrar la hora en la que entré y salí de las instalaciones				
Criterios de aceptación:				
# Escenario	Dado que	Cuando	Entonces	Notas / Reglas de Negocio
1	Soy Colaborador y estoy en laborando en	Ingreso a las instalaciones de la empresa	Se debe reconocer mi rostro y con ello registrar mi ingreso en el registro de la compañía.	La fecha y hora se almacenarán en UTC haciendo

	la empresa.			las conversiones necesarias en los reportes.
2	Soy Colaborador y estoy en laborando en la empresa.	Salgo de las instalaciones de la empresa	Se debe reconocer mi rostro y con ello registrar mi salida en el registro de la compañía.	La fecha y hora se almacenarán en UTC haciendo las conversiones necesarias en los reportes.

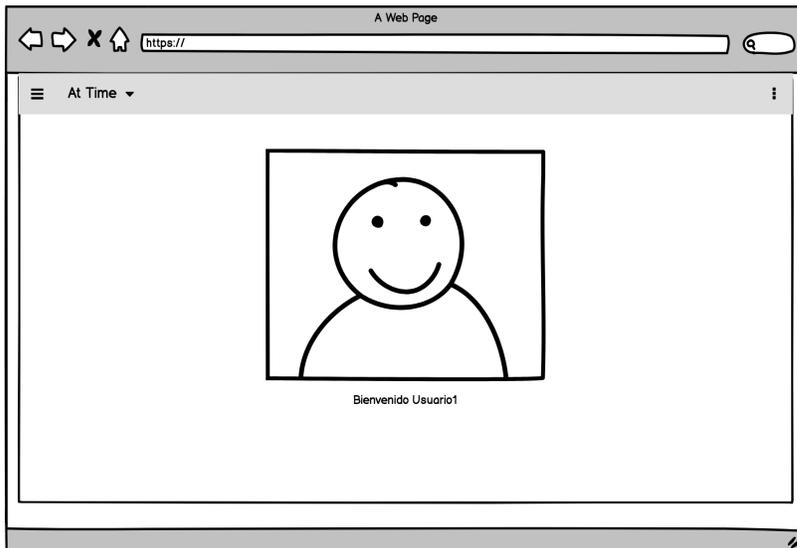


Figura 14: Interfaz bienvenido

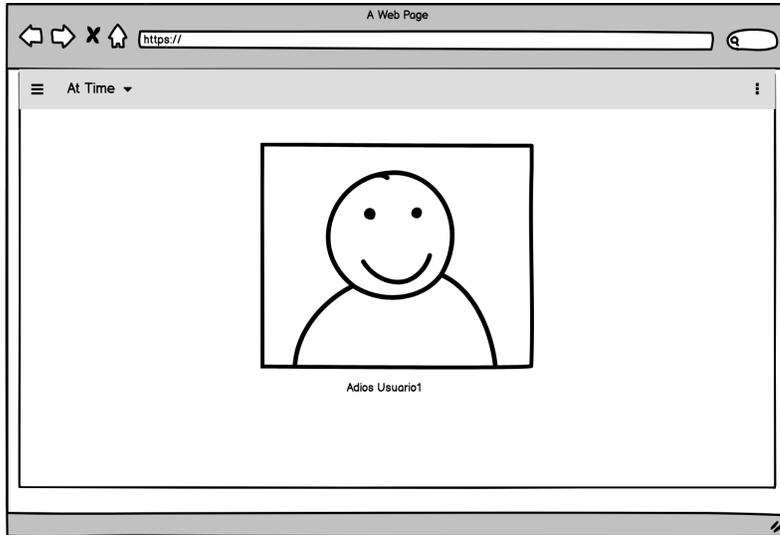


Figura 15: Interfaz Adios

4.2 Detección de estado de ánimo

Tabla 3: Historia de usuario Deteccion de estado de ánimo

Actor: Colaborador				
Descripción: Como usuario Colaborador que el sistema de autenticación detecte mi estado de ánimo para que la compañía pueda llevar estadísticas de clima laboral				
Criterios de aceptación:				
#	Dado	Cuando	Entonces	Notas /
Escenar	que			Reglas de
io				Negocio

1	Soy Colaborador y estoy en laborando en la empresa.	Ingreso o salida a las instalaciones de la empresa	Se debe registrar mi estado de ánimo en el registro de la compañía.	Los estados de ánimo son los siguientes: <ul style="list-style-type: none"> ● Enojado ● Triste ● Normal ● Feliz
---	---	--	---	---

Nota: Esta Historia de usuario no tiene interfaz gráfica

4.3 Alta de nuevos colaboradores

Tabla 4: Historia de usuario de nuevos colaboradores

Actor: Administrador				
Descripción: Como Administrador necesito poder dar de alta a un nuevo empleado en el sistema para que pueda ingresar a su lugar de trabajo				
Criterios de aceptación:				
# Escenar	Dado que	Cuando	Entonces	Notas / Reglas de

io				Negocio
1	Soy Administrador y estoy en el módulo de administración de usuarios	Selecciono la funcionalidad de agregar un nuevo usuario	Se abre una ventana en la cual se puede ingresar la información básica de registro: <ul style="list-style-type: none"> ● Primer nombre ● Apellido ● Número de identificación. ● Correo electrónico. 	
2.	Soy Administrador y estoy en el módulo de administración de usuarios y la llene los datos básicos de usuario	Hago click en registrar imagen de usuario.	Si es la primera vez pedirá acceso a la cámara del computador. <p>Se prendera la cámara del computador.</p> <p>Se tomará una serie de fotos de la persona.</p> <p>Se guardarán los datos de la persona y se</p>	

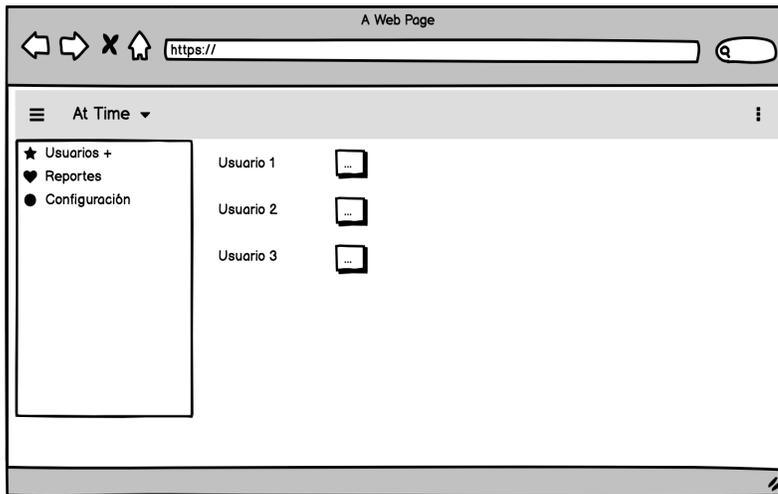
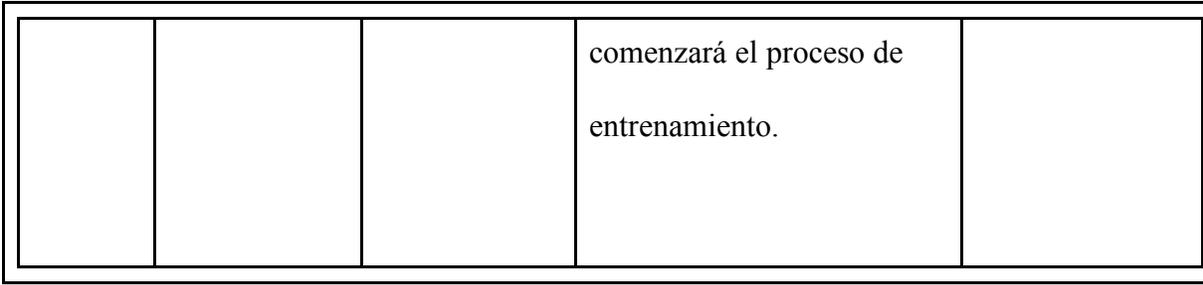


Figura 16: Interfaz listado de usuarios

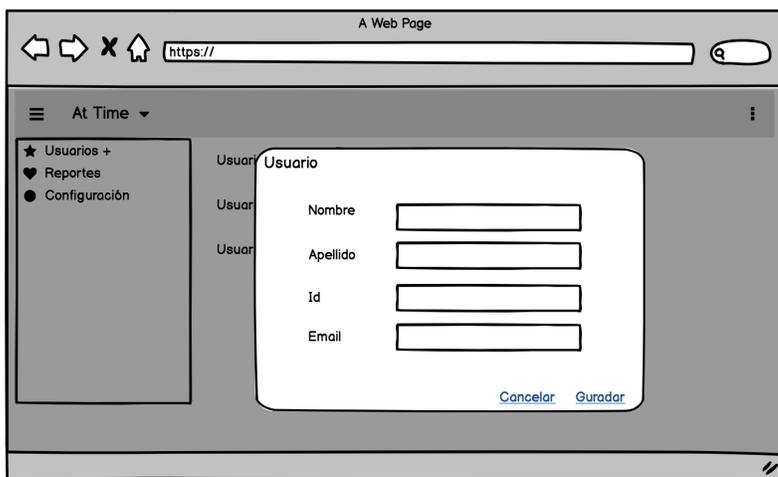


Figura 17: Interfaz Crear nuevo usuario

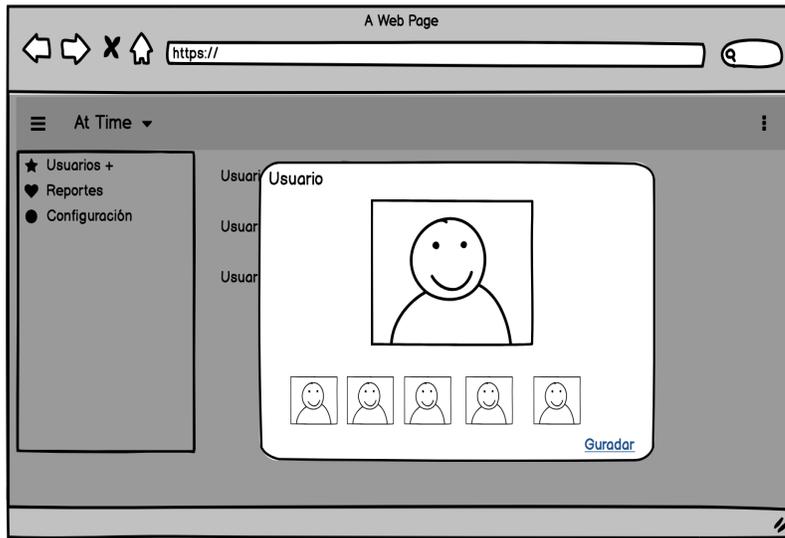


Figura 18: Interfaz registrar fotos

4.4 Reporte de ingreso

Tabla 5: Historia de usuario reporte de ingreso

Actor: Analista de recursos Humanos				
Descripción: Como Analista de recursos humanos necesito poder ver el reporte de ingreso de un usuario de un tiempo determinado para llevar la estadística del usuario.				
Criterios de aceptación:				
#	Dado	Cuando	Entonces	Notas /
Escenar	que			Reglas de

io				Negocio
1	Soy Analista de recursos Humanos	Ingreso a la Parte de reporte de ingreso y selecciono uno de los colaboradores	Muestra los registros de entradas y salidas del usuario seleccionado	

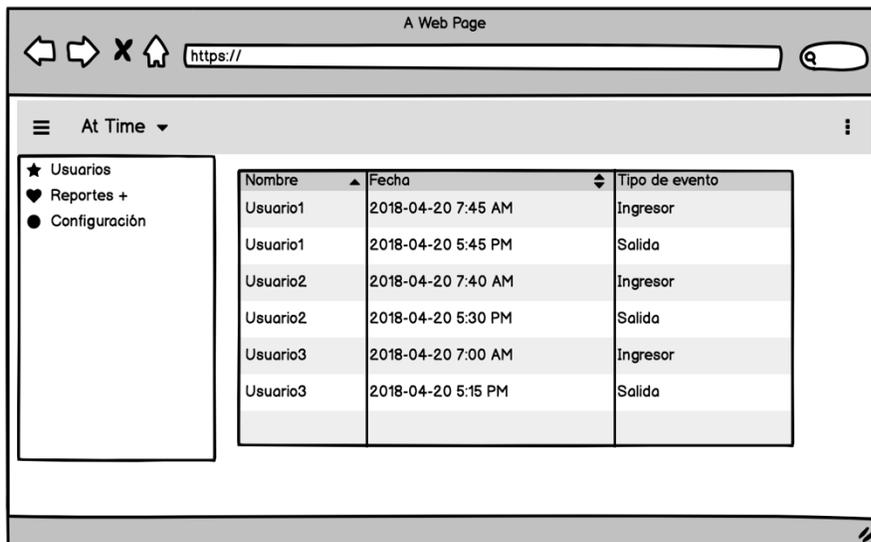


Figura 19: Interfaz reporte de ingresos y salidas por usuario

4.5 Reporte de estado de ánimo

Tabla 6: Historia de usuario Reporte de estado de ánimo

Actor: Analista de recursos humanos				
Descripción: Como Analista de recursos humanos necesito poder ver los gráficos de estado de ánimo de un tiempo determinado para verificar el clima laboral de la compañía.				
Criterios de aceptación:				
# Escenario	Dado que	Cuando	Entonces	Notas / Reglas de Negocio
1	Soy Analista de recursos Humanos	Ingreso a la Parte de reporte de estado de ánimo y selecciono las fechas en la que quiero ver el reporte	La aplicación muestra una gráfica de torta en la que se muestran los porcentajes cada una de las categorías de estado de ánimo.	

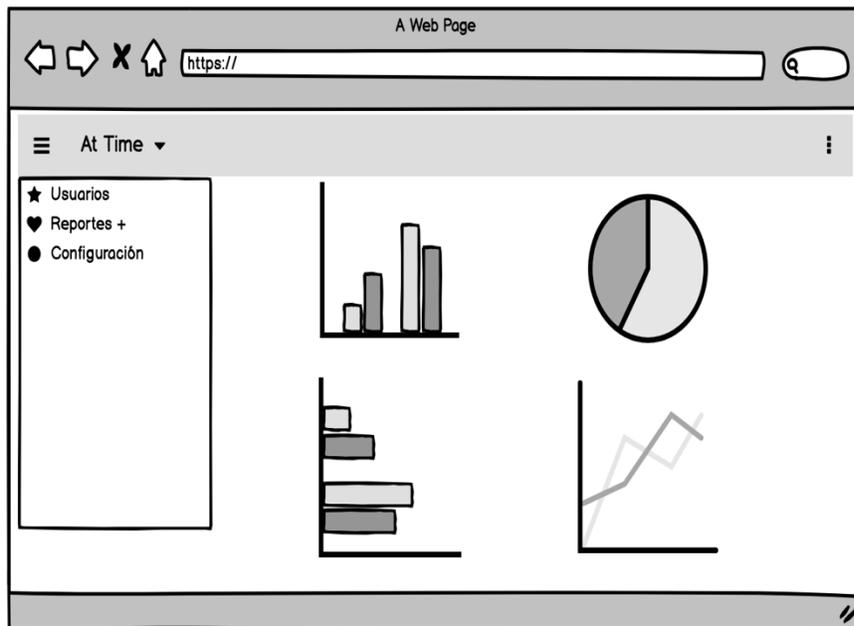


Figura 20: Interfaz gráfico de estado de animo

5. Requerimientos no funcionales y definición de arquitectura

En este capítulo veremos cómo a través de los requerimientos no funcionales y las historias de usuario se desarrollara la arquitectura que se implementara para poder satisfacer las necesidades técnicas de la aplicación.

5.1 Requerimientos no funcionales

- **Seguridad:** Teniendo en cuenta que la aplicación almacena información sensible, se requiere proteger la integridad y privacidad de está de una manera cuidadosa. Por lo tanto

la seguridad debe estar regidas por la guía de OWASP (Open Web Application Security Project) y cumplir como mínimo el TOP 10(Top 10-2017, s.f.) de recomendaciones de esta organización.

- **Usabilidad:** La aplicación debe componerse de una interfaz sencilla e intuitiva, lo cual permitirá que el usuario se pueda adaptar satisfactoriamente al uso de la misma. Además debe tener guías de usuario de fácil acceso donde se explique el funcionamiento de la aplicación.
- **Desempeño.** El sistema debe estar en la capacidad de dar respuesta a las solicitudes de los usuarios con tiempos de respuesta aceptables (alrededor de 5 segundos) y uniformes sin importar el número de usuarios concurrentes que estén en la aplicación.
- **Disponibilidad:** El sistema debe tener una disponibilidad de 100% o muy cercano las 24 horas del día para poder brindar el servicio a países con diferentes zonas horarias.
- **Escalabilidad y Concurrencia:** Se requiere que la aplicación permite ser utilizada por altos volúmenes de usuarios concurrentes por lo tanto debe ser desarrollada en las herramientas y servicios de Google Cloud Platform y su arquitectura debe tener en consideración que la base de datos debe soportar este volumen de transacciones.
- **Facilidad para pruebas:** El sistema debe contar con facilidades para la identificación de la localización de los errores durante la etapa de pruebas y de operación posterior.
- **Multi tenancy:** La aplicación debe ser desarrollada con soporte multi tenancy por lo cual debe poder mantener múltiples clientes teniendo separados los datos de cada uno, debido a que son información sensible y no deben ser expuestos a otros clientes.

5.2 Vista lógica

5.2.1 Modelo de datos

La base de datos a utilizar será del tipo no relacional o NoSQL (Adam & Jakob, 2010) la cual no cuenta con relaciones fuertemente establecidas, pero permite una mejor escalabilidad. Para tener una mejor comprensión del esquema de datos se realizará una estructuración de los datos en formato de clases de UML y con ello se tendrá un entendimiento de estos y cómo se verán en la aplicación.

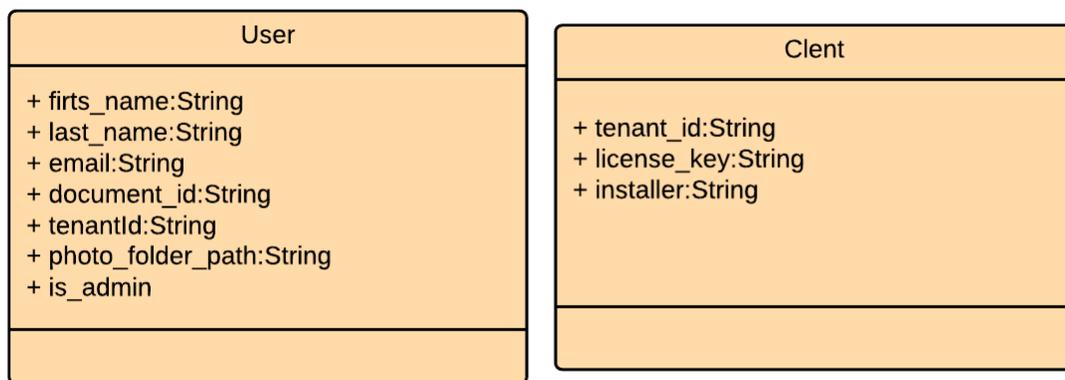


Figura 21: Diagrama de datos usuarios

- **User:** entidad donde se guarda la información básica de los usuarios así como la ruta donde se almacenan las fotos para procesar el modelo.
- **Client:** entidad donde se almacena información del cliente como el dominio para el que se instaló y el usuario instalador.

Adicionalmente para el almacenamiento del log de eventos se utilizará una solución de data warehouse que permita tener las entradas y salidas almacenadas y fácilmente consultables además de permitir el almacenamiento y consulta de grandes volúmenes de información

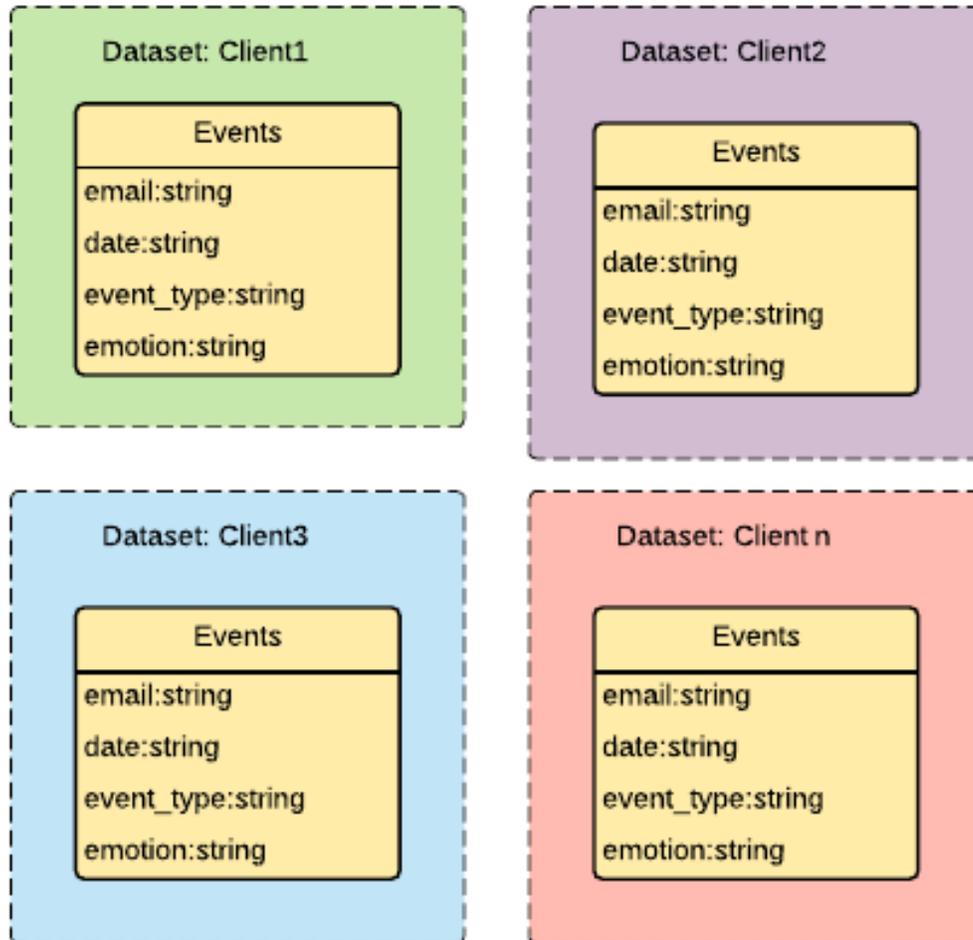


Figura 22: Diagrama de datos eventos

- **Events**: tabla donde se guarda el registro de los eventos de los usuarios y los estados de ánimo. los tipos de eventos son **in**, **out**. Los estados de ánimo son: **angry**, **normal**, **happy**, **sad**.

Los eventos se almacenarán en diferentes datasets los cuales serán uno por cada cliente.

5.2 Vista de despliegue

En esta vista se muestran los elementos que se utilizarán de Google Cloud Platform los cuales compondrán la infraestructura de la solución.

5.2.1 App Colaboradores

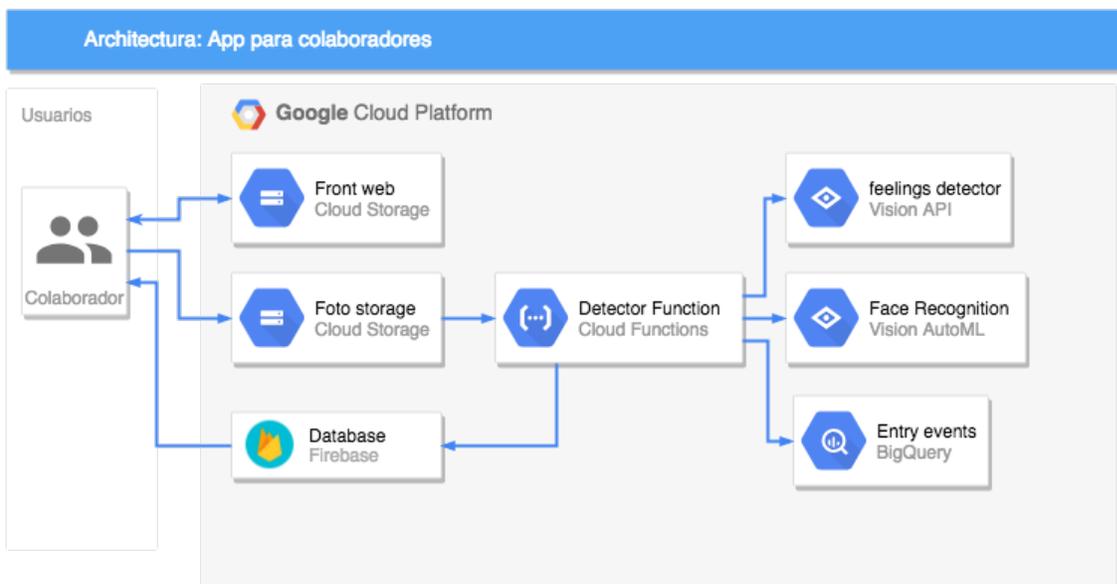


Figura 23: Diagrama de despliegue de la app para colaborado

- **Colaborador:** Persona registrada en el sistema a la cual se le registran los ingresos y salidas.
- **Front web:** Bucket de Cloud Storage en el cual se almacena la aplicación frontend escrita en javascript css y html. Este componente funcionara como un servidor web para los

archivos estáticos.

- **Foto storage:** Bucket de Cloud Storage en el cual se subirán las imágenes de las personas para poder ser detectadas. Cuando se suba una foto se lanzará un disparador de evento.
- **Detector Function:** Función que es accionada cuando se sube un foto al bucket de Foto storage. esta función es contiene toda la lógica de negocio para llamar a las diferentes APIs y almacenar los eventos en las bases de datos.
- **Database:** Base de datos en tiempo real la cual se utiliza como sistema de notificaciones para alertar al frontend que se realizó la detección del rostro del usuario.
- **Feelings detector:** para la detección de sentimientos se llama al API de visión de Google el cual no dirá si una persona está feliz, triste, enojado entre otros.
- **Face Recognition:** Para el reconocimiento facial se está utilizando Vision AutoML el cual es un componente de GCP el cual permite utilizar los modelos pre entrenados para reconocimiento de personas y rostros. Gracias a este componentes se puede realizar una especialización para que reconozca personas en concreto.
- **Entry events:** Es la bodega de datos en la cual se almacenará todos los eventos y la cual nos permitirá realizar procesamiento de terabytes de información para poder realizar las gráficas y estadísticas.

5.2.2 App de Administración

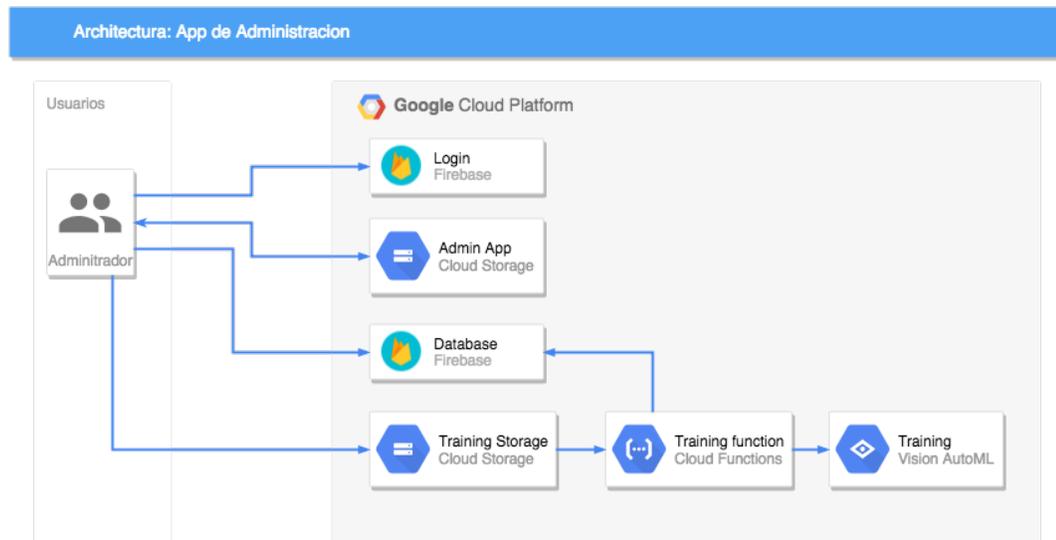


Figura 24: Diagrama de despliegue App de Administración

- **Login:** Sistema de autenticación basado en Firebase que permite validar a los usuarios contra diferentes servicios como Google, Facebook, Twitter, Correo, etc.
- **Admin App:** Bucket de Cloud Storage que sirve como servidor web para los archivos estáticos que componen la aplicación de front para la administración.
- **Database:** base de datos en tiempo real donde almacena todos los datos relevantes como la información de los usuarios.
- **Training Storage:** Bucket de Cloud Storage donde se almacenan las fotos de las personas para realizar el entrenamiento de los modelos de Vision AutoML para reconocer a personas en concreto.
- **Training:** Para el entrenamiento se utiliza AutoML con el fin de aprovechar los modelos

pre entrenados de Vision API y con ello facilitar las tareas de reconocimiento específico de personas.

5.3 Vista de implementación

En la vista de implementación se pondrá la distribución de los diferentes paquetes que tendrá la aplicación así como sus interacciones para poder ver la distribución arquitectónica de la solución.

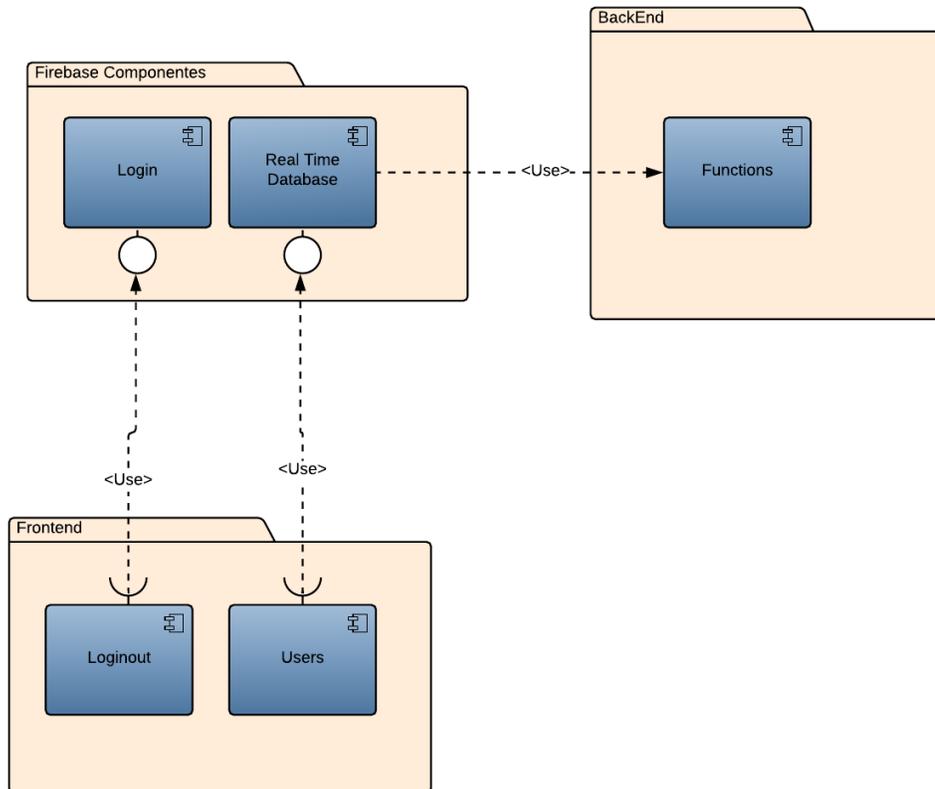


Figura 25: Diagrama de Componentes

6. Descripción del sistema.

En este capítulo veremos una descripción detallada de cómo fue desarrollado el sistema así como su entorno y librerías. Además se verán algunas capturas de pantalla que representan la funcionalidad de la aplicación.

6.1 Entorno de desarrollo Backend

- Lenguaje: python 3.7
- Entorno de ejecución: Google Cloud Functions
- Base de datos: Google Firebase
- Framework web: Flask
- Manejador de dependencias: PIP (PIP 18.0)
- Guía de estilo de código fuente: PEP8 (PEP 8 -- Style Guide for Python Code)

6.2 Entorno de desarrollo frontend

- Lenguaje: Javascript
 - Soporte: Chrome
 - Framework: Angular 1.5
 - Librerías gráficas (css): Preferiblemente un framework de grillas como grid960
- Preprocesador CSS: Stylus

- Manejador de dependencias: Bower
- Guía de estilos de código fuente: Google JavaScript Style Guide (Google JavaScript Style Guide)

6.3 Capturas de la aplicación

A continuación veremos algunas de las capturas de pantalla de la aplicación que demuestran cómo quedó el estado final.

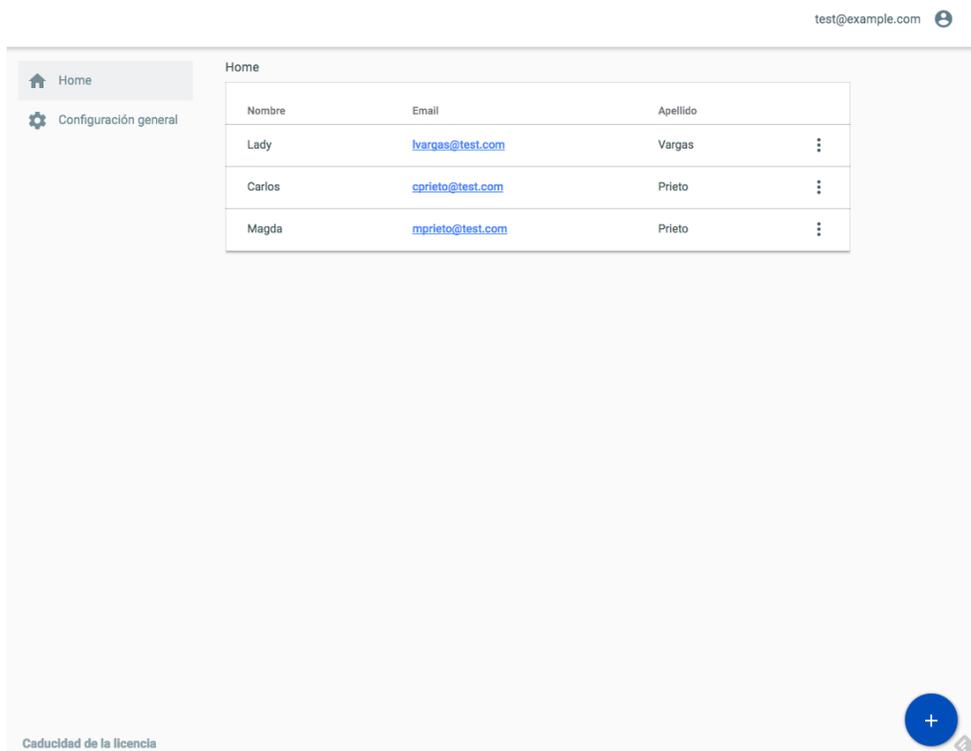


Figura 26: Listado de usuarios

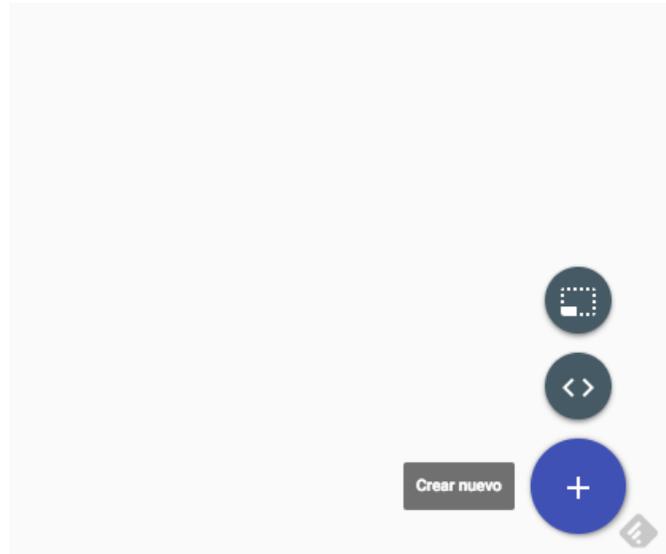


Figura 27: Boton nuevo usuario

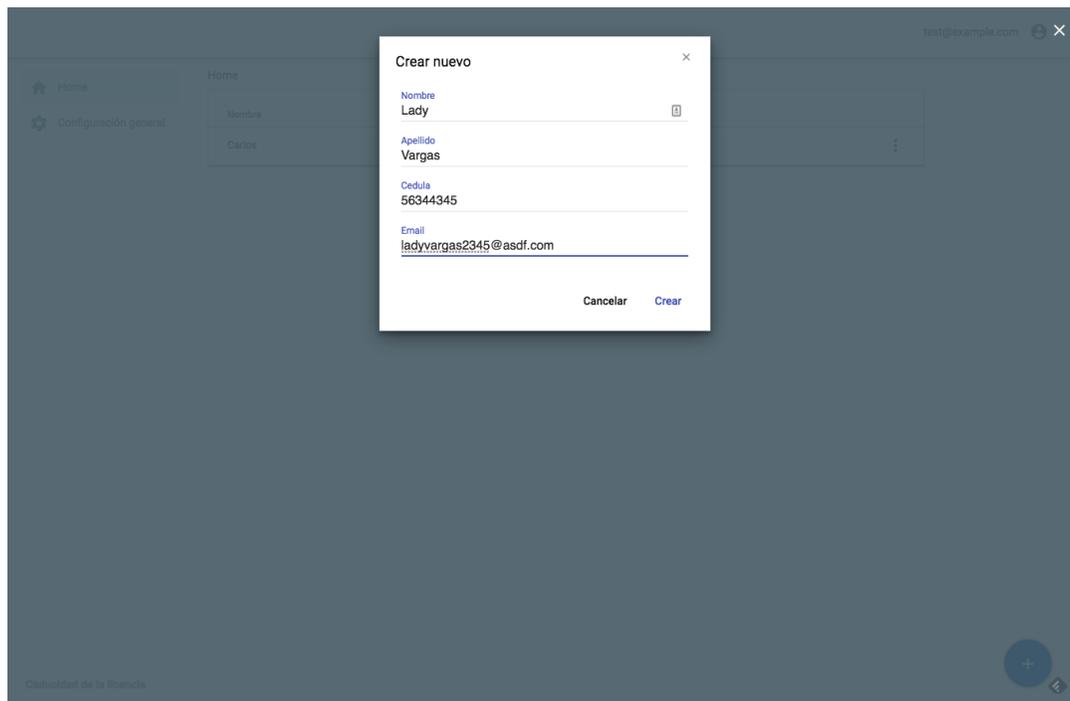


Figura 28: Creación de usuario



Figura 29: Opciones de menú de usuario

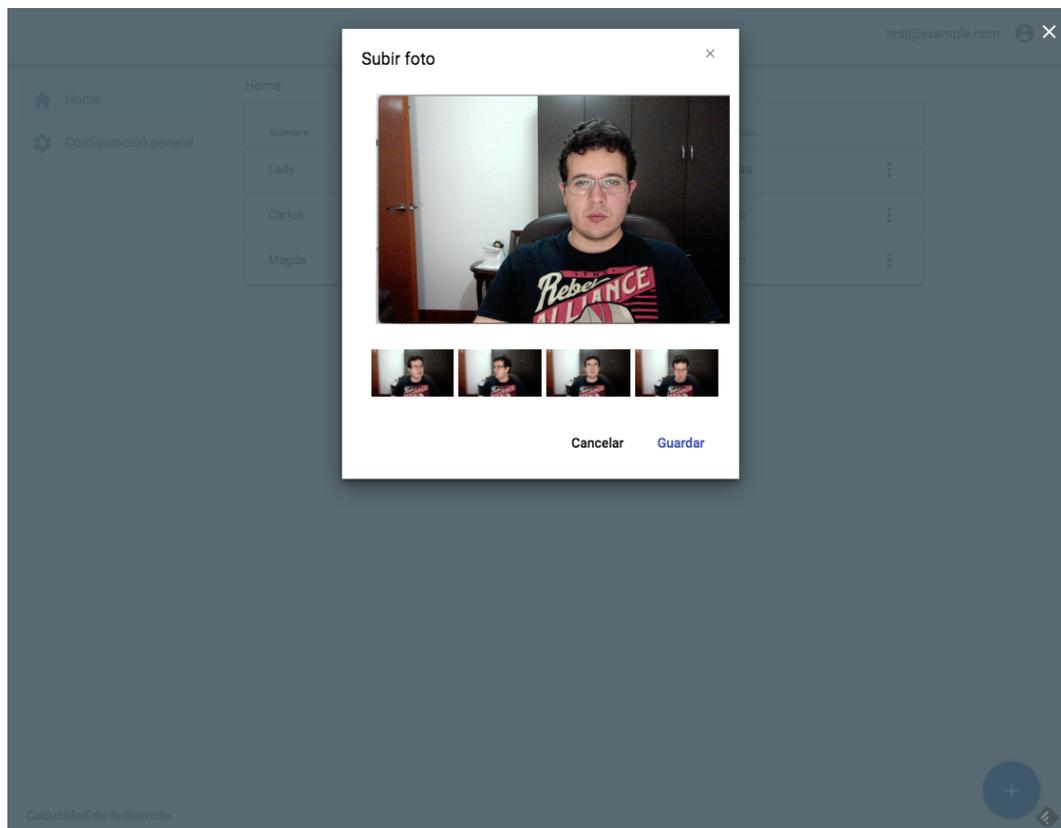


Figura 30: Carga de imagenes

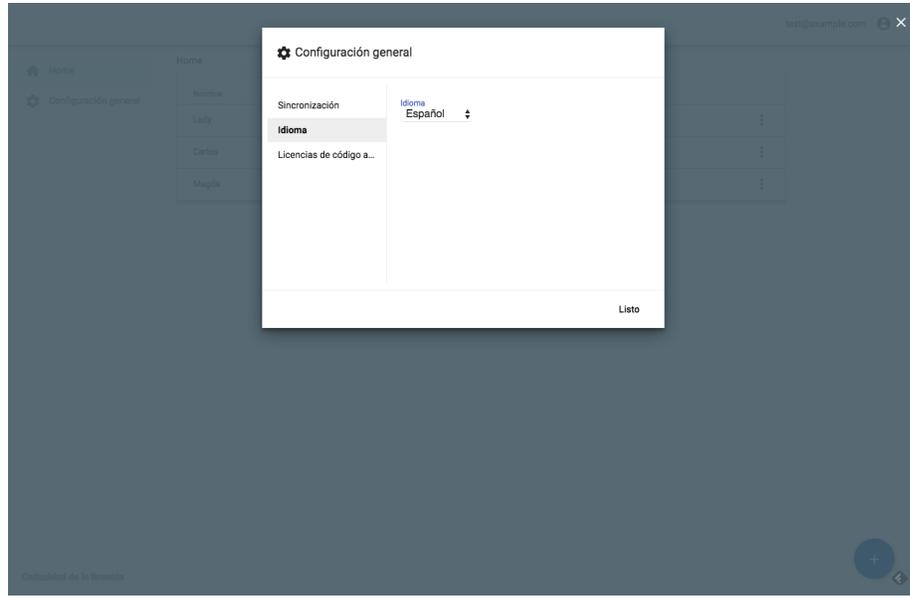


Figura 31: Cambio de idioma



Figura 32: Ingreso de usuario



Figura 33: Salida de usuario

7. Evaluación

Para evaluar la usabilidad de la aplicación se realizó una demostración a diez personas con diferentes perfiles entre los cuales hay contadores, ingenieros de sistemas, diseñadores gráficos, auxiliares administrativos, publicistas y periodistas para saber su opinión a través de una encuesta. Las preguntas realizadas fueron las siguientes:

- Nombre y Apellido
- Correo electrónico
- Qué tan intuitiva le pareció la aplicación?
- Fue fácil ubicar los diferentes componentes?
- Como le pareció el diseño gráfico de la aplicación?
- Estaría dispuesto a utilizar esta aplicación para registrar el ingreso y la salida en su trabajo diario?
- Recomendaría el uso de esta aplicación en su empresa?
- Que sugerencia tendría para futuras versiones de la aplicación?

Las gráficas de las respuestas son las siguientes:

Qué tan intuitiva le pareció la aplicación?

10 respuestas

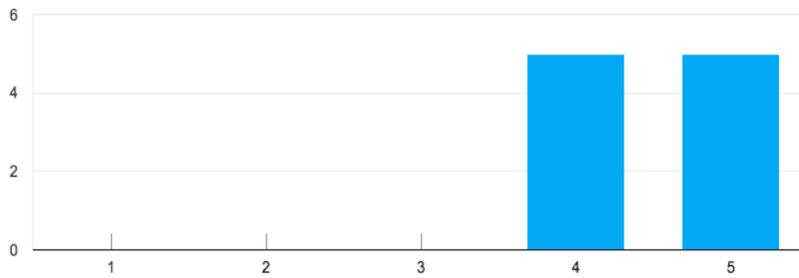


Figura 34: Estadísticas pregunta No. 01

Fue fácil ubicar los diferentes componentes?

10 respuestas

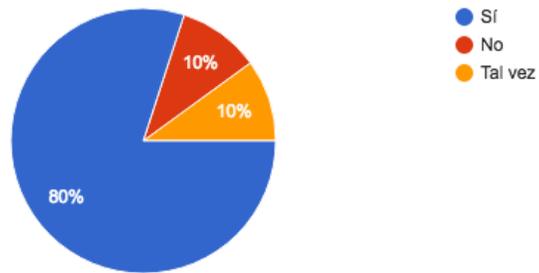


Figura 35: Estadísticas pregunta No. 02

Como le pareció el diseño gráfico de la aplicación?

10 respuestas

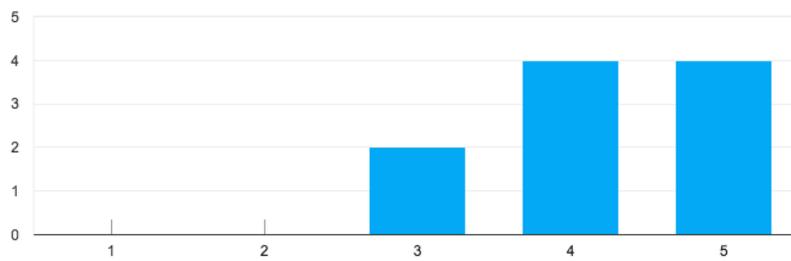


Figura 36: Estadísticas pregunta No. 03

Estaría dispuesto a utilizar esta aplicación para registrar el ingreso y la salida en su trabajo diario?

10 respuestas

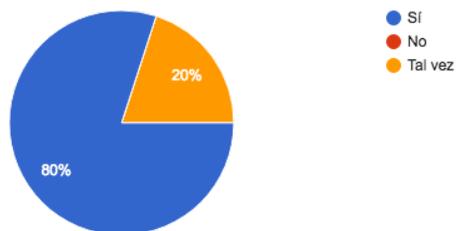


Figura 37: Estadísticas pregunta No. 04

Recomendaría el uso de esta aplicación en su empresa?

10 respuestas

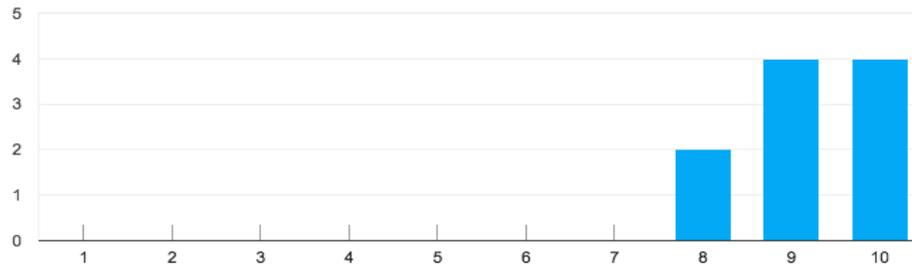


Figura 38: Estadísticas pregunta No. 05

Que sugerencia tendría para futuras versiones de la aplicación?

10 respuestas

Me parecio buena
Seria mas facil si no tuviera que hacer click para que detectara a la person
Debería poder servir para autenticarse en otras aplicaciones
Me pareció muy oscura y faltan mensajes para que sea más fácil de entender
Falta un elemento que permita motivar la utilización de la aplicación como Gamification
Me parece que asi esta bien :D
No tengo
Que la aplicación se pueda descargar en cualquier dispositivo.
Me gustaria poder ver las estadísticas de mis resultados.
NA

Figura 39: Estadísticas pregunta No. 06

8. Conclusiones y Líneas de trabajo futuro

8.1 Conclusiones.

Mirando de manera retrospectiva el proceso de creación tanto de la aplicación como de estas memorias se ha llegado a las conclusiones que serán presentadas a continuación:

- Se modeló y se implementó una arquitectura de micro servicios la cual pudo dar una base tecnológica robusta a la aplicación.
- Gracias a los modelos pre entrenados por medio de redes neuronales artificiales se logró incorporar tecnologías de reconocimiento facial y detección de sentimientos sin tener la necesidad de tener conocimientos avanzados en inteligencia artificial.
- Gracias a los servicios que están englobados en la plataforma de cómputo en la nube de Google se pudo desarrollar una aplicación altamente escalable que puede ser tomada como referencia para arquitecturas informáticas modernas de tipo serverless.
- En el desarrollo de este proyecto se logró adquirir el conocimiento para diseñar un data warehouse, que tiene la capacidad de procesar terabytes de información en cuestión de segundos, con el cual se le dio soporte a la analítica de la aplicación tomando como base la herramienta Bigquery.
- Cada vez es más relevante para las empresas conocer cómo se sienten sus empleados para mejorar sus condiciones de trabajo y nivel de vida, con el fin de aumentar la productividad.

8.2 Líneas futuras

La aplicación presentada en este trabajo es tan solo un prototipo y hay muchos campos de mejora los cuales serán presentados a continuación en forma de Backlog que se implementara para futuras versiones de la aplicación:

- Tomando ventaja de las nuevas tecnologías de IOT (Internet of Things (IoT)) de Google como es el caso de Android Things (Developers, s.f.) y la salida de la librería de aprendizaje de máquina para móviles TensorFlow Lite (Techrunch, s.f.) se puede diseñar un dispositivo de bajo coste que permita realizar el reconocimiento facial de manera local y fuera de línea para agilizar los procesos de entrada del personal.
- Por medio de la sincronización de usuarios con bases de datos de empleados externas, tales como directorios LDAP, se puede agilizar el proceso de registro de usuarios en la aplicación.
- Hay que definir los costos operativos del sistema para poder definir un precio de salida al público.

9. Bibliografía

Villatoro, F. R. (25 de 10 de 2011). *La Ciencia de la Mula Francis*. Obtenido de Jhon McCarthy (1927-2011), el padre de la inteligencia artificial:

<https://francis.naukas.com/2011/10/25/john-mccarthy-1927-2011-el-padre-de-la-inteligencia-artificial/>

Aprendizaje Supervisado y no Supervisado. (s.f.). Obtenido de Redes Neuronales:

<http://redesneuronares.blogspot.com/>

SearchCloudComputing. (s.f.). Obtenido de Platform as a Service (PaaS):

<https://searchcloudcomputing.techtarget.com/definition/Platform-as-a-Service-PaaS>

Armasu, L. (19 de 05 de 2016). *Tom's Hardware*. Obtenido de Google's Big Chip Unvenil For Machine: <https://www.tomshardware.com/news/google-tensor-processing-unit-machine-learning,31834.html>

Oufaska, Y. (28 de 05 de 2018). *Café Ágil*. Obtenido de Scrum con Desing Thinking:

<https://cafe-agil.com/2018/05/28/scrum-con-design-thinking/#more-423>

Gestion de la Comunicación en la empresa. (22 de 04 de 2013). Obtenido de Principio de Preto:

<http://sharingideas-josecavd.blogspot.com/2013/04/principio-de-pareto-principios-leyes-y.html>

Proyectos agiles.org. (s.f.). Obtenido de Qué es SCRUM: <https://proyectosagiles.org/que-es-scrum/>

A. L., & J. M. (2010). Obtenido de Investigating storage solution for large data:

<http://publications.lib.chalmers.se/records/fulltext/123839.pdf>

Internet of Things (IoT). (s.f.). Obtenido de Getting started with Cisco IoT:

<https://www.cisco.com/c/en/us/solutions/internet-of-things/overview.html>

Developers. (s.f.). Obtenido de Supported hardware:

<https://developer.android.com/things/hardware/>

Techcrunch. (s.f.). Obtenido de Google's TensorFlow Lite brings machine learning to Android

devices: <https://techcrunch.com/2017/05/17/googles-tensorflow-lite-brings-machine-learning-to-android-devices/>

TensorFlow. (s.f.). Obtenido de An Open source machine learning library for research and

production: <https://www.tensorflow.org/>

OpenCV. (s.f.). Obtenido de <https://opencv.org/>

Pytorch. (s.f.). Obtenido de Tensors and Dynamic neural networks in Python with strong GPU

acceleration: <https://pytorch.org/>

Google for Entrepreneurs. (08 de 08 de 2016). Obtenido de Rapid Prototyping 1 of 3: Sketching

& Paper Prototyping: <https://www.youtube.com/watch?v=JMjozqJS44M>

Un genio vagabundo amante de la lógica, el padre incomprendido de la cibernética. (s.f.).

Obtenido de hojaderouter.com:

https://www.eldiario.es/hojaderouter/ciencia/Walter_Pitts-McCulloch-pioneros-cibernetica-inteligencia_artificial_0_367814000.html

Guijarro, V., & G. L. (2010). *La quimera del autómata matemático*. Madrid.

El coloso de Rodas. (2017). Obtenido de Guia de Grecia:

<http://www.guiadegrecia.com/dodeca/coloso.html>

Golem: Jewish Magical and Mystical Traditions on the Artificial Anthropoid. (1990). Albany.

Gonzalez, R. (07 de 2011). *Descartes: Las intuiciones modales y la inteligencia artificial*

clasica. Obtenido de https://scielo.conicyt.cl/scielo.php?script=sci_arttext&pid=S0718-22012011000100014

Alan Turing - Maquina Computacional e Inteligencia. (s.f.). Obtenido de

<https://matap.dmae.upm.es/cienciaficcio/DIVULGACION/3/TestTuring.htm>

Basic Questions. (s.f.). Obtenido de <http://www-formal.stanford.edu/jmc/whatisai/node1.html>

Fifth Generation Computer Systems. (s.f.). Obtenido de ScienceDirect:

<https://www.sciencedirect.com/book/9780444864406/fifth-generation-computer-systems>

The Story of AlphaGo so Far. (s.f.). Obtenido de AlphaGo:

<https://deepmind.com/research/alphago/>

Prieto, M. (s.f.). *Los Asistentes de Voz se Hacen Mayores*. Obtenido de Expansión - Tecnología:

<http://www.expansion.com/tecnologia/2017/03/25/58d58bb3e2704e164b8b45c0.html>

An open source machine learning library for research and production. (s.f.). Obtenido de

TensorFlow: <https://www.tensorflow.org/>

A Tour of Machine Learning Algorithms. (s.f.). Obtenido de Machine Learning Mastery:

<https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/>

Gefter, A. (02 de 2015). *The Man Who Tried to Redeem the World with Logic*. Obtenido de

Nautilus: <http://nautil.us/issue/21/information/the-man-who-tried-to-redeem-the-world-with-logic>

Artificial Neural Networks as Models of Neural Information Processing. (s.f.). Obtenido de

<https://www.frontiersin.org/research-topics/4817/artificial-neural-networks-as-models-of-neural-information-processing>

Object Detection by Contour Segment Networks. (s.f.). Obtenido de <http://calvin.inf.ed.ac.uk/wp-content/uploads/Publications/ferrari-eccv06.pdf>

Geitgey, A. (24 de 07 de 2016). *Machine Learning is Fun!* Obtenido de Medium:

<https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78>

Top 10-2017. (s.f.). Obtenido de https://www.owasp.org/index.php/Top_10-2017_Top_10

Proyecto Perceptron Simple (Código C++). (11 de 06 de 2018). Obtenido de Inteligencia

Artificial: <https://info--red.blogspot.com/2018/06/proyecto-perceptron-simple-codigo-c.html>

(s.f.). Obtenido de IBM Watson: The inside story of how Jeopardy-winning supercomputer was

born, and what it wants to do next: <https://www.techrepublic.com/article/ibm-watson-the-inside-story-of-how-the-jeopardy-winning-supercomputer-was-born-and-what-it-wants-to-do-next/>

(s.f.). Obtenido de PEP 8 -- Style Guide for Python Code: <https://www.python.org/dev/peps/pep-0008/>

(s.f.). Obtenido de PIP 18.0: <https://pypi.org/project/pip/>

(s.f.). Obtenido de Google JavaScript Style Guide:

<https://google.github.io/styleguide/jsguide.html>

(s.f.). Obtenido de OpenFace: <https://cmusatyalab.github.io/openface/#openface>

ANEXOS

Artículo

Abstract

The main objective of the work was designed and construct a facial recognition system that supports organizations in the purpose of collecting information related with entrances and exits of employees to its offices or workplaces. In addition, through feelings recognition, organizations can be given valuable assistance in collection of information on the work climate to carry statistics that allow the implementation of improvement actions.

For the implementation of this system, Scrum methodology was used, which allows rapid progress thanks to its agile software development practices, ensuring its quality and reliability.

In conclusion, it was determined that the development of machine learning systems already has a sufficient level of maturity to which we can involve them in real cases and thereby enrich the systems giving support to our organizations.

Keywords: Facial recognition, machine learning, neural networks, practical application.

I. Introducción

El aprendizaje de máquina es una de las tecnologías que más ha evolucionado en el campo de la Inteligencia artificial. Gracias a esto se ha comenzado a dar grandes avances en algunos sectores como lo son el reconocimiento de fraudes en tarjetas de crédito, la conducción de carros autónomos, comprensión del lenguaje natural, y traducciones entre diferentes idiomas. A pesar de todos los avances que se han hecho y de tener múltiples librerías, frameworks y tecnologías tanto de código libre como propietarias que facilitan el alcance del aprendizaje de máquina, aún es un concepto que pocas personas entienden.

El propósito de este trabajo es llevar esta tecnología a un caso práctico de la vida real, el cual puede ayudar a las compañías a mejorar y conocer el ambiente laboral de una manera mas practica que la elaboración de complejas encuestas en las cuales no se demuestra el verdadero estado de ánimo de los empleados.

Uno de los problemas que aquejan a las organizaciones es la manera de poder controlar el horario de los empleados de una manera eficiente y sin ser intrusivo. Esto ha llevado a las organizaciones a implementar soluciones costosas basadas en hardware como lo son las tarjetas de acceso que en caso de pérdida pueden representar riesgos de seguridad para la organización. Por otro lado las áreas de recursos humanos no han logrado una manera eficiente de poder recolectar información relevante para poder medir cómo se sienten los empleados y de esta manera tener estadísticas del clima laboral. Como respuesta a estos inconvenientes se decidió crear un software que permite poder recoger la información de las entradas y salidas de los empleados a sus oficinas o plantas de trabajo por medio del reconocimiento facial. Además el software está en la capacidad de recolectar estadísticas de clima laboral por medio de la detección de sentimientos de los empleados para dar apoyo en la toma de decisiones de los departamentos de recursos humanos para mejorar las condiciones de trabajo de los empleados.

Para llevar a cabo este software se utilizaron múltiples tecnologías que están bajo Google Cloud Platform las cuales facilitan la implementación de los algoritmos necesarios para realizar el reconocimiento facial y la detección de sentimientos.

II. Contexto y estado del arte

Como parte de las tecnologías propuestas para el desarrollo de este trabajo se evalúa la utilización de Inteligencia artificial como medio de detección de rostros y de sentimientos. Por ello es necesario entender de manera clara algunos de los conceptos que están alrededor como lo son el aprendizaje de máquina, aprendizaje profundo y redes neuronales artificiales para llevarlos a un medio concreto y poderlos aplicar en la arquitectura que se propondrá. **Aprendizaje de máquina**

Aprendizaje automático o aprendizaje de máquina es una rama de la inteligencia artificial que pretende que los programas computacionales puedan aprender con lo cual podrán generar conocimiento y con este aprendizaje resolver problemas.

Como resultado del aprendizaje de máquina se obtienen modelos los cuales se utilizan posteriormente para el análisis de información algunos de estos modelos son modelos geométricos, modelos probabilísticos o modelos lógicos.

Los algoritmos utilizados para el aprendizaje de máquina, Están clasificados en los siguientes métodos:

- **Aprendizaje supervisado**
- **Aprendizaje no supervisado**
- **Aprendizaje semi supervisado**
- **Aprendizaje por refuerzo**

Modelos de análisis visual

Antes de realizar el reconocimiento de rostros hay que realizar una serie de pasos anteriores los cuales constituyen la ruta para generar un modelo de aprendizaje de máquina el cual pueda ayudarnos con nuestra labor.

Detección de objetos: El primero de ellos es la detección de objetos la cual consiste en que nuestro modelo simplifique las imágenes en una sucesión de líneas que son las que delinear el contorno de los diferentes objetos. Para este propósito es necesario poder verificar el contraste de las imágenes y ver los cambios de colores que nos permitirán generar una sucesión de líneas con las cuales se podrá generar el contorno.

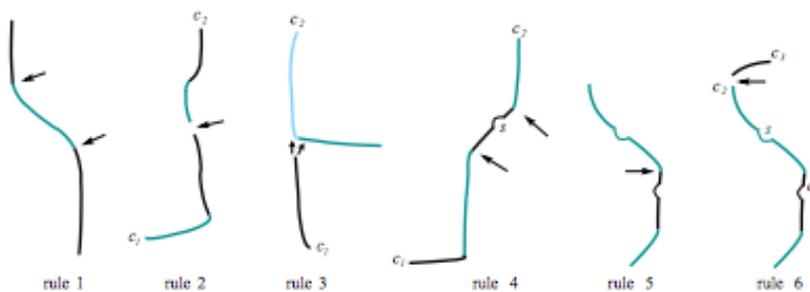


Figura 1: Seis reglas para delinear el contorno (Object Detection by Contour Segment Networks)

Detección y reconocimiento facial: Un caso específico de la detección de objetos es la detección de rostros la cual es una tarea en la que específicamente se buscan rostros humanos. Este algoritmo se centra en la detección frontal de rostros por medio de los puntos del contorno de la cara, ojos cejas y nariz.

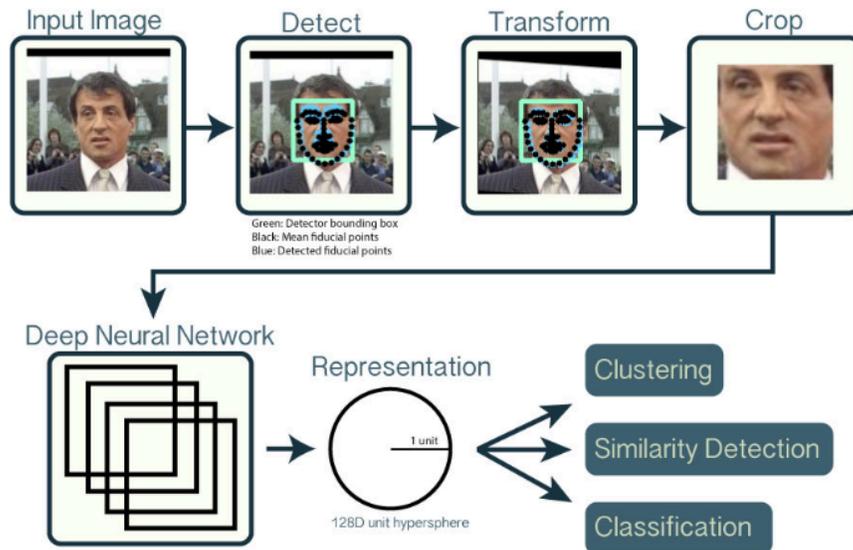


Figura 2:Proceso de deteccin de rostro (OpenFace, s.f.)

Pero la detección de rostros es solo el principio de la solución que se quiere implementar. Adicionalmente hay que realizar el reconocimiento lo cual permite identificar individuos por sus características únicas para poder realizar la diferenciación entre múltiples personas. Algunos de los problemas más comunes son:

- Encontrar en la foto los rostros
- Entender que si la persona está con la cara en otra dirección o incluso hay una mala iluminación en la foto aun sigue siendo la misma persona.
- Ser capaz de diferenciar esas características únicas con las cuales se puede diferenciar un rostro.
- Finalmente poder comparar esas características con una base de datos de los rostros almacenados para poder realizar la detección adecuada.

Por esto es que se necesita un gran poder de cómputo y el apoyo de las redes neuronales artificiales para poder realizar estas tareas.

Se realiza un mapeo de los píxeles y luego el algoritmo enumera cada uno de los puntos relevantes en el rostro y hace un análisis de características claves como la separación de las cejas, distancia entre la nariz y los ojos entre otras.

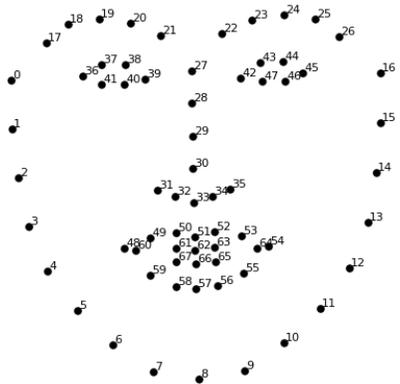


Figura 3: Puntos clave del rostro (*A Tour of Machine Learning Algorithms, s.f.*)

APIs de inteligencia artificial de Google

Dentro del catálogo de servicios que tiene Google en su plataforma de nube se encuentra una categoría enfocada a temas de aprendizaje de máquina los cuales están contemplados para personas con diferentes habilidades que van desde los programadores hasta los científicos de datos con experiencia en inteligencia artificial.

Del lado de las APIs Google tiene toda una gama de servicios los cuales son accesibles de manera programática y brindan el potencial de dotar con capacidades de inteligencia artificial a los programas sin la necesidad de ser un experto en ciencias de datos. Entre estas APIs se encuentran:

- Cloud Speech que permite convertir un audio en texto.
- Cloud Text-to-Speech que tiene la capacidad de convertir un texto en voz.
- Cloud Natural Language descubre la estructura y el significado del texto mediante potentes modelos de aprendizaje automático en una API REST fácil de usar.
- Translation ofrece una interfaz de programación sencilla para traducir una cadena arbitraria a cualquier idioma compatible
- Cloud Video Intelligence facilita la búsqueda y el descubrimiento de vídeos mediante la extracción de metadatos, la identificación de palabras clave y la anotación de contenidos.
- Cloud Vision clasifica rápidamente imágenes en miles de categorías (por ejemplo, "barco de vela" o "torre Eiffel"), detecta objetos y caras por separado dentro de ellas e incluso encuentra y lee las palabras que contienen.

III. Implementación

Gracias a el servicio de los modelos pre entrenados de Google Cloud AutoML que permite personalizar la solución de detección de rostros para el reconocimiento facial se logró tener una funcionalidad robusta y sin tener que tener conocimientos especializados en ciencias de datos.

Para realizar la detección de sentimientos se utilizó Vision API de la plataforma de nube de Google, la cual permite obtener meta información de una imagen como puede ser la composición de colores, los elementos relevantes, detección de la ubicación de los rostros humanos, entre otras cosas.

Este sistema se implementó por medio de la metodología scrum la cual permite avanzar de manera rápida gracias a sus prácticas ágiles de desarrollo de software asegurando la calidad y fiabilidad del mismo

Para la definición de requerimientos se utilizaron las historias de usuario además de la metodología de prototipado de interfaces gráficas propuesto por Google para poder tener una idea rápida de cómo funcionaría la aplicación. Las historias definidas son:

- Registro de tiempo con reconocimiento facial
- Detección de estado de ánimo
- Alta de nuevos colaboradores
- Reporte de ingreso
- Reporte de estado de ánimo

Este software fue pensado desde sus principios para ser altamente escalable y orientado a una arquitectura multitenancy la cual permite realizar una separación de los datos de los diferentes clientes de una manera segura y confiable. Esta Arquitectura será mostrada a continuación:

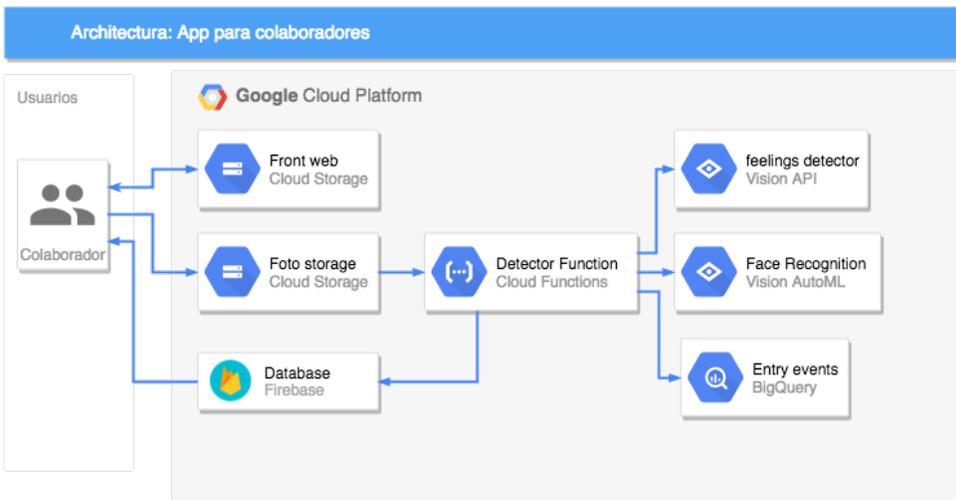


Figura 4: Diagrama de Arquitectura de App Colaboradores

- **Colaborador:** Persona registrada en el sistema a la cual se le registran los ingresos y salidas.
- **Front web:** Bucket de Cloud Storage en el cual se almacena la aplicación frontend escrita en javascript css y html. Este componente funcionara como un servidor web para los archivos estáticos.
- **Foto storage:** Bucket de Cloud Storage en el cual se subirán las imágenes de las personas para poder ser detectadas. Cuando se suba una foto se lanzará un disparador de evento.

- **Detector Function:** Función que es accionada cuando se sube un foto al bucket de Foto storage. esta función es contiene toda la lógica de negocio para llamar a las diferentes APIs y almacenar los eventos en las bases de datos.
- **Database:** Base de datos en tiempo real la cual se utiliza como sistema de notificaciones para alertar al frontend que se realizó la detección del rostro del usuario.
- **Feelings detector:** para la detección de sentimientos se llama al API de visión de Google el cual no dirá si una persona está feliz, triste, enojado entre otros.
- **Face Recognition:** Para el reconocimiento facial se está utilizando Vision AutoML el cual es un componente de GCP el cual permite utilizar los modelos pre entrenados para reconocimiento de personas y rostros. Gracias a este componentes se puede realizar una especialización para que reconozca personas en concreto.
- **Entry events:** Es la bodega de datos en la cual se almacenará todos los eventos y la cual nos permitirá realizar procesamiento de terabytes de información para poder realizar las gráficas y estadísticas.

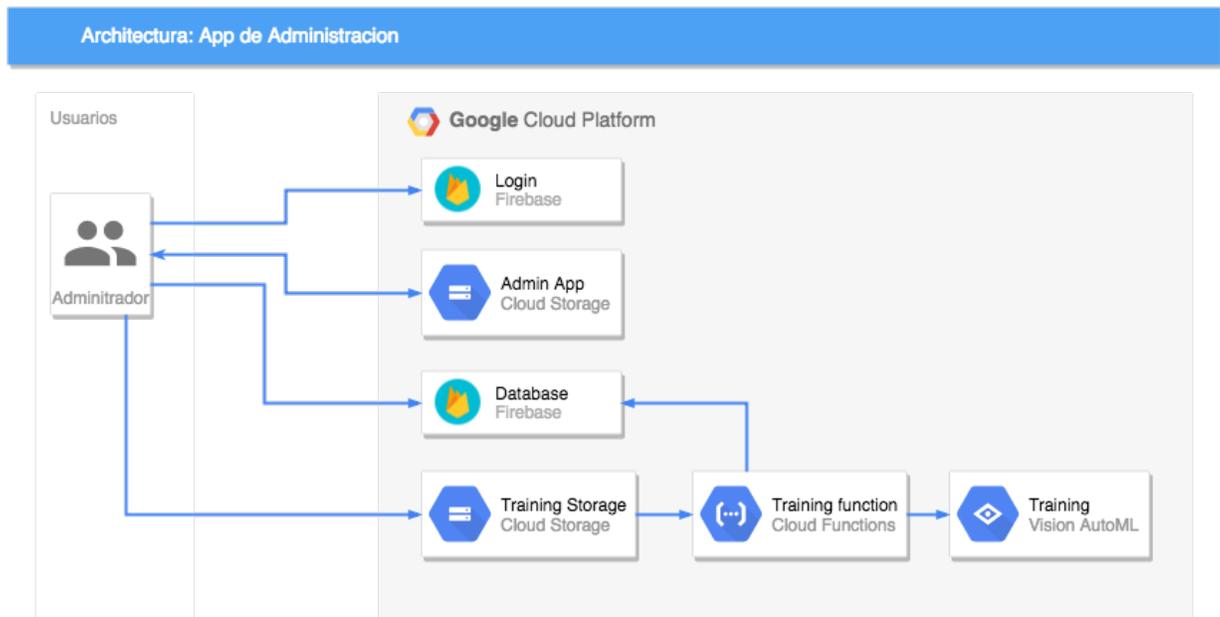


Figura 5: App de Administracion

- **Login:** Sistema de autenticación basado en Firebase que permite validar a los usuarios contra diferentes servicios como Google, Facebook, Twitter, Correo, etc.
- **Admin App:** Bucket de Cloud Storage que sirve como servidor web para los archivos estáticos que componen la aplicación de front para la administración.
- **Database:** base de datos en tiempo real donde almacena todos los datos relevantes como la información de los usuarios.

- **Training Storage:** Bucket de Cloud Storage donde se almacenan las fotos de las personas para realizar el entrenamiento de los modelos de Vision AutoML para reconocer a personas en concreto.
- **Training:** Para el entrenamiento se utiliza AutoML con el fin de aprovechar los modelos pre entrenados de Vision API y con ello facilitar las tareas de reconocimiento específico de personas.

Con respecto al esquema de datos la solución quedó de la siguiente manera:

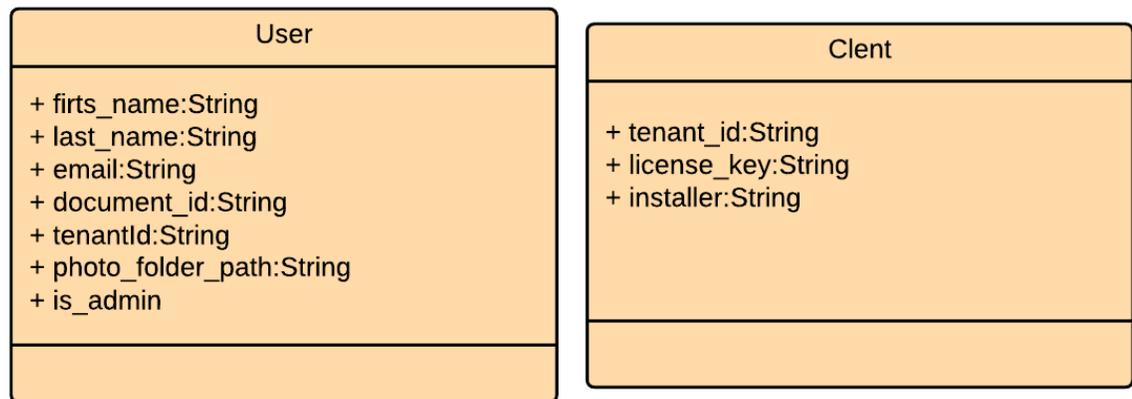


Figura 6: Diagrama de clases de datos

- **User:** entidad donde se guarda la información básica de los usuarios así como la ruta donde se almacenan las fotos para procesar el modelo.
- **Client:** entidad donde se almacena información del cliente como el dominio para el que se instaló y el usuario instalador.

IV. Descripción del sistema

El sistema será descrito a continuación

Entorno de desarrollo Backend

- **Lenguaje:** python 3.7
- **Entorno de ejecución:** Google Cloud Functions
- **Base de datos:** Google Firebase
- **Framework web:** Flask
- **Manejador de dependencias:** PIP
- **Guía de estilo de código fuente:** PEP8

Entorno de desarrollo frontend

- **Lenguaje:** Javascript
- **Soporte:** Chrome
- **Framework:** Angular 1.5

- **Librerías gráficas (css):** Preferiblemente un framework de grillas como grid960
Preprocesador CSS: Stylus
- **Manejador de dependencias:** Bower
- **Guía de estilos de código fuente:** Google JavaScript Style Guide

V. Evaluación

Para evaluar la usabilidad de la aplicación se realizó una demostración a diez personas con diferentes perfiles entre los cuales hay contadores, ingenieros de sistemas, diseñadores gráficos, auxiliares administrativos, publicistas y periodistas para saber su opinión a través de una encuesta.

- Las preguntas realizadas fueron las siguientes:
 - Qué tan intuitiva le pareció la aplicación
 - Fue fácil ubicar los diferentes componentes
 - Como le pareció el diseño gráfico de la aplicación
 - Estaría dispuesto a utilizar esta aplicación para registrar el ingreso y la salida de su trabajo diario
 - Recomendaría el uso de esta aplicación en su empresa
 - Que sugerencia tendría para futuras versiones de la aplicación

VI. Conclusiones

- Se modeló y se implementó una arquitectura de micro servicios la cual pudo dar una base tecnológica robusta a la aplicación.
- Gracias a los modelos pre entrenados por medio de redes neuronales artificiales se logró incorporar tecnologías de reconocimiento facial y detección de sentimientos sin tener la necesidad de tener conocimientos avanzados en inteligencia artificial.
- Gracias a los servicios que están englobados en la plataforma de cómputo en la nube de Google se pudo desarrollar una aplicación altamente escalable que puede ser tomada como referencia para arquitecturas informáticas modernas de tipo serverless.
- En el desarrollo de este proyecto se logró adquirir el conocimiento para diseñar un data warehouse, que tiene la capacidad de procesar terabytes de información en cuestión de segundos, con el cual se le dio soporte a la analítica de la aplicación tomando como base la herramienta Bigquery.
- Cada vez es más relevante para las empresas conocer cómo se sienten sus empleados para mejorar sus condiciones de trabajo y nivel de vida, con el fin de aumentar la productividad.

VII. Líneas futuras

Como parte de la evolución de este producto se tiene pensado poder utilizar las capacidades de Android Things (La plataforma de internet de las cosas de Google) y TensorFlow Lite (librería ligera de machine learning que corre en Android) para diseñar un dispositivo capaz de realizar reconocimiento facial en lugares donde se tenga un acceso limitado a internet.

VIII. Biografía

- 1) Villatoro, F. R. (25 de 10 de 2011). *La Ciencia de la Mula Francis*. Obtenido de Jhon McCarthy (1927-2011), el padre de la inteligencia artificial: <https://francis.naukas.com/2011/10/25/john-mccarthy-1927-2011-el-padre-de-la-inteligencia-artificial/>
- 2) *Aprendizaje Supervisado y no Supervisado*. (s.f.). Obtenido de Redes Neuronales: <http://redesneuronales.blogspot.com/>
- 3) *SearchCloudComputing*. (s.f.). Obtenido de Platform as a Service (PaaS): <https://searchcloudcomputing.techtarget.com/definition/Platform-as-a-Service-PaaS>
- 4) Armasu, L. (19 de 05 de 2016). *Tom's Hardware*. Obtenido de Google's Big Chip Unvenil For Machine: <https://www.tomshardware.com/news/google-tensor-processing-unit-machine-learning,31834.html>
- 5) Oufaska, Y. (28 de 05 de 2018). *Café Ágil*. Obtenido de Scrum con Desing Thinking: <https://cafe-agil.com/2018/05/28/scrum-con-design-thinking/#more-423>
- 6) *Gestion de la Comunicación en la empresa*. (22 de 04 de 2013). Obtenido de Principio de Preto: <http://sharingideas-josecavd.blogspot.com/2013/04/principio-de-pareto-principios-leyes-y.html>
- 7) *Proyectos agiles.org*. (s.f.). Obtenido de Qué es SCRUM: <https://proyectosagiles.org/que-es-scrum/>
- 8) A. L., & J. M. (2010). Obtenido de Investigating storage solution for large data: <http://publications.lib.chalmers.se/records/fulltext/123839.pdf>
- 9) *Internet of Things (IoT)*. (s.f.). Obtenido de Getting started with Cisco IoT: <https://www.cisco.com/c/en/us/solutions/internet-of-things/overview.html>
- 10) *Developers*. (s.f.). Obtenido de Supported hardware: <https://developer.android.com/things/hardware/>
- 11) *Techcrunch*. (s.f.). Obtenido de Google's TensorFlow Lite brings machine learning to Android devices: <https://techcrunch.com/2017/05/17/googles-tensorflow-lite-brings-machine-learning-to-android-devices/>
- 12) *TensorFlow*. (s.f.). Obtenido de An Open source machine learning library for research and production: <https://www.tensorflow.org/>
- 13) *OpenCV*. (s.f.). Obtenido de <https://opencv.org/>
- 14) *Pytorch*. (s.f.). Obtenido de Tensors and Dynamic neural networks in Python with strong GPU acceleration: <https://pytorch.org/>
- 15) Google for Entrepreneurs. (08 de 08 de 2016). Obtenido de Rapid Prototyping 1 of 3: Sketching & Paper Prototyping:

- <https://www.youtube.com/watch?v=JMjozqJS44M>
- 16) *Un genio vagabundo amante de la lógica, el padre incomprendido de la cibernética*. (s.f.). Obtenido de hojaderouter.com:
https://www.eldiario.es/hojaderouter/ciencia/Walter_Pitts-McCulloch-pioneros-cibernetica-inteligencia_artificial_0_367814000.html
 - 17) Guijarro, V., & G. L. (2010). *La quimera del autómatas matemático*. Madrid.
 - 18) *El coloso de Rodas*. (2017). Obtenido de Guia de Grecia:
<http://www.guiadegrecia.com/dodeca/coloso.html>
 - 19) *Golem: Jewish Magical and Mystical Traditions on the Artificial Anthropoid*. (1990). Albany.
 - 20) Gonzalez, R. (07 de 2011). *Descartes: Las intuiciones modales y la inteligencia artificial clasica*. Obtenido de
https://scielo.conicyt.cl/scielo.php?script=sci_arttext&pid=S0718-22012011000100014
 - 21) *Alan Turing - Maquina Computacional e Inteligencia*. (s.f.). Obtenido de
<https://matap.dmae.upm.es/cienciaficcio/DIVULGACION/3/TestTuring.htm>
 - 22) *Basic Questions*. (s.f.). Obtenido de <http://www-formal.stanford.edu/jmc/whatisai/node1.html>
 - 23) *Fifth Generation Computer Systems*. (s.f.). Obtenido de ScienceDirect:
<https://www.sciencedirect.com/book/9780444864406/fifth-generation-computer-systems>
 - 24) *The Story of AlphaGo so Far*. (s.f.). Obtenido de AlphaGo:
<https://deepmind.com/research/alphago/>
 - 25) Prieto, M. (s.f.). *Los Asistentes de Voz se Hacen Mayores*. Obtenido de Expansión - Tecnología:
<http://www.expansion.com/tecnologia/2017/03/25/58d58bb3e2704e164b8b45c0.html>
 - 26) *An open source machine learning library for research and production*. (s.f.). Obtenido de TensorFlow: <https://www.tensorflow.org/>
 - 27) *A Tour of Machine Learning Algorithms*. (s.f.). Obtenido de Machine Learning Mastery: <https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/>
 - 28) Gefter, A. (02 de 2015). *The Man Who Tried to Redeem the World with Logic*. Obtenido de Nautilus: <http://nautilus.us/issue/21/information/the-man-who-tried-to-redeem-the-world-with-logic>
 - 29) *Artificial Neural Networks as Models of Neural Information Processing*. (s.f.). Obtenido de <https://www.frontiersin.org/research-topics/4817/artificial-neural-networks-as-models-of-neural-information-processing>
 - 30) *Object Detection by Contour Segment Networks*. (s.f.). Obtenido de <http://calvin.inf.ed.ac.uk/wp-content/uploads/Publications/ferrari-eccv06.pdf>
 - 31) Geitgey, A. (24 de 07 de 2016). *Machine Learning is Fun!* Obtenido de Medium:
<https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78>
 - 32) *Top 10-2017*. (s.f.). Obtenido de https://www.owasp.org/index.php/Top_10-2017_Top_10

- 33) *Proyecto Perceptron Simple (Código C++)*. (11 de 06 de 2018). Obtenido de Inteligencia Artificial: <https://info--red.blogspot.com/2018/06/proyecto-perceptron-simple-codigo-c.html>
- 34) (s.f.). Obtenido de IBM Watson: The inside story of how Jeopardy-winning supercomputer was born, and what it wants to do next: <https://www.techrepublic.com/article/ibm-watson-the-inside-story-of-how-the-jeopardy-winning-supercomputer-was-born-and-what-it-wants-to-do-next/>
- 35) (s.f.). Obtenido de PEP 8 -- Style Guide for Python Code: <https://www.python.org/dev/peps/pep-0008/>
- 36) (s.f.). Obtenido de PIP 18.0: <https://pypi.org/project/pip/>
- 37) (s.f.). Obtenido de Google JavaScript Style Guide: <https://google.github.io/styleguide/jsguide.html>
- 38) (s.f.). Obtenido de OpenFace: <https://cmusatyalab.github.io/openface/#openface>