

# Entrega 1

# Proyecto IA

# Sempai

Maximiliano Durand  
Sebastián Closas  
Paulo Crescionini  
Vicente Castelló

<b>1. Problema</b>	<b>3</b>
<b>2. Solución</b>	<b>3</b>
<b>3. Modelo</b>	<b>4</b>
<b>3.1 Detección de rostro</b>	<b>4</b>
<b>3.2 Identificación</b>	<b>4</b>
<b>4. Herramientas</b>	<b>5</b>
<b>4.1 Gestión</b>	<b>5</b>
<b>4.2 Programación</b>	<b>5</b>
<b>4.3 Machine learning</b>	<b>5</b>
<b>5. Roles</b>	<b>6</b>
<b>6. Metodología de trabajo</b>	<b>6</b>
<b>8. Servicios en la nube a utilizar</b>	<b>6</b>
<b>9. Ciclos de desarrollo</b>	<b>6</b>
<b>9.1 Sprint 1</b>	<b>6</b>
<b>9.2 Backlog</b>	<b>7</b>
<b>10. Procesos de iteración a realizar</b>	<b>7</b>

# 1. Problema

Existe la necesidad de poder controlar la entrada y salida de personal de la empresa.

También se necesita restringir el acceso del personal a ciertas áreas por temas de seguridad.

Otras de las necesidades es llevar un registro del tiempo que el personal está en las zonas de recreo y esparcimiento. Por ejemplo, controlar el tiempo que una persona está en la cafetería

# 2. Solución

Se instalarán cámaras en lugares estratégicos de la empresa con el fin de registrar una imagen de las personas al momento de ingresar y salir de los distintos sectores. Se aplicará reconocimiento facial para poder identificar a las personas de la organización.

Con esto se podrá desarrollar una solución que permita responder las siguientes consultas:  
¿Cuánto tiempo está una persona en determinado lugar?

¿Una persona estuvo en determinado lugar?

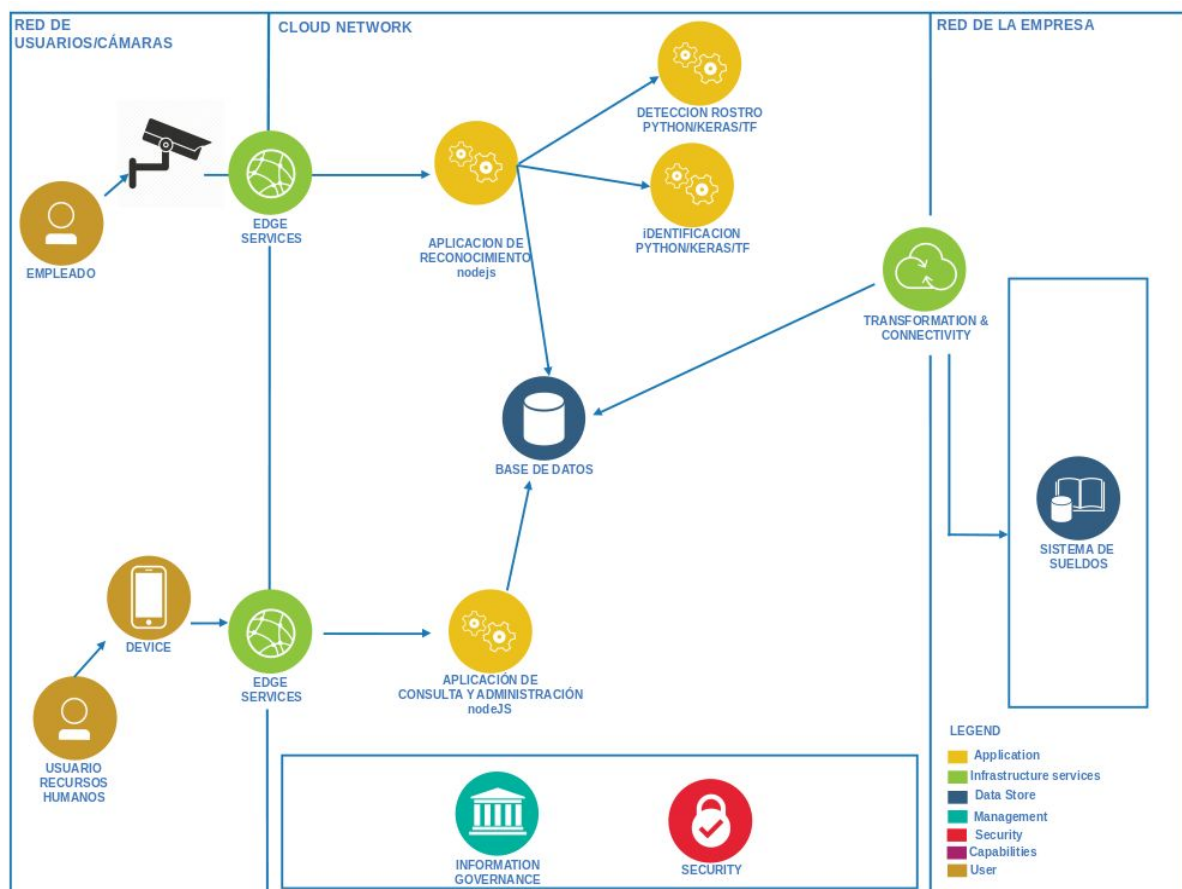
También se podrá identificar patrones en posicionamiento del personal.

Se plantea el funcionamiento de la solución en tres etapas:

1. Adquisición de la imagen.
  - a. Se toma una imagen (cruda) de la cámara. Esto puede ser mediante polling o envío directo desde la cámara si la misma detecta movimiento.
  - b. Se utiliza un modelo (CNN) que permita detectar la ubicación (coordenadas) de una cara dentro de la imagen.
  - c. Se recorta la imagen, resultando en una imagen que contiene solo la cara de la persona.
2. Procesamiento e identificación
  - a. Dada la imagen de la persona (o varias imágenes capturadas) se aplicará otro modelo de identificación. Este modelo sería otro CNN que dada la imagen de la cara retorna un conjunto de características, representadas como un vector, que permitan identificar a una persona. Estas características, podrían ser rasgos como color de pelo, forma de la cara, color de ojos, o podrían ser rasgos abstractos identificados por el modelo al momento del entrenamiento. Depende del modelo que utilicemos.
3. Almacenamiento
  - a. Una vez identificada la persona se registra en la base de datos:
    - i. La hora

- ii. El lugar (determinado por la cámara)
    - iii. La persona
    - iv. Porcentaje de acierto
    - v. La imagen como prueba (opcional).
  - b. Un proceso debe determinar si se trata de registros de entrada, salida o simplemente de presencia. También se debe encargar de agrupar registros duplicados.
4. Consulta
- a. El sistema mostrará reportes de marcas por persona
  - b. Se podrá conectar con un sistema externo, por ejemplo del liquidacion de sueldos para enviar un reporte mensual de las horas trabajadas por cada empleado.

A continuación se muestra un diagrama de la arquitectura propuesta:



## 3. Modelo

### 3.1 Detección de rostro

Para este modelo se utilizarán una Red Neuronal Convolucional.

Su entrada va a ser la imagen completa y con un modelo definido con la librería OpenCV (Open Source Computer Vision Library) obtendremos el rostro de la imagen.

Primer paso sería la detección de la(s) caras o rostros, mediante un modelo entrenado que detecta caras frontales (face classifier). Y que nos devuelva las coordenadas (x1,y1) (x2, y2) de la posición de la cara dentro de la imagen o nos devuelve las coordenadas de la esquina superior izquierda x,y más el ancho y el largo.

### 3.2 Identificación

Utilizaremos una red del tipo “Red Neuronales Convolucional” (CNN). El modelo va a ser Secuencial y luego le iremos agregando las distintas capas que serian: capas de convolucionales (profundidad a la red), Capas de pooling “MaxPooling2D” (para ir reduciendo el tamaño de las imágenes) y Capas de Dropout (para evitar casos de overfitting). La idea es ir alternando estas 3 capas e ir reduciendo el tamaño de las imágenes. Y por último una capa Flatten y una capa Densa.

La funcion de activacion que vamos a utilizar es “relu”, que solo se activa si son positivos, y que se comporta bien con imágenes y tiene buen desempeño en redes convolucionales.

## 4. Herramientas

### 4.1 Gestión

- **Slack** - Para la comunicación del equipo
- **Trello** - Para la administración de tareas.
- **Google Drive** - Para compartir archivos y editar documentos.

### 4.2 Programación

- Git y GitHub - como repositorio de código
- Visual Studio Code - IDE
- NodeJS.

- Python.
- Express.
- JavaScript.

## 4.3 Machine learning

- Google Colab.
- TensorFlow.
- Keras.

## 5. Roles

A continuación se detalla la asignación de roles para el proyecto:

- Arquitecto: Vicente, Paulo
- Backend Developer: Sebastián
- Frontend Developer: Maximiliano
- Data Scientist: Sebastián
- Desarrollador del Modelo IA: Maximiliano
- Tester: Paulo
- QA: Vicente

## 6. Metodología de trabajo

Se va a seguir la metodología IBM Garage, con sus stages:

- Culture. Cambiar las cabezas.
- Discover. Descubrir y resolver problemas.
- Envision. Marcar metas.
- Develop. Desarrollar soluciones MVPs, que aporten valor al cliente.
- Reason. Integrar Inteligencia Artificial y Ciencia de Datos a la solución.
- Operate. Operar la solución.
- Learn. Nunca dejar de aprender para mejorar la solución.

## 8. Servicios en la nube a utilizar

Analizaremos las opciones que más se ajusten a nuestros requerimientos:

- IBM Watson.
- Microsoft Azure.
- Amazon SageMaker.

- Google Vision AI.

## 9. Ciclos de desarrollo

En el stage de Desarrollo utilizaremos Scrum.

A continuación se enumera el plan inicial de los distintos sprints.

### 9.1 Sprint 1

- Definición de roles.
- Definición de herramientas.
- Investigar posibles soluciones.
- Investigar servicios en la nube a utilizar.
- Definir Ciclos de desarrollo.
- Definir procesos de iteración a realizar.
- Armado de documento de primera entrega de proyecto.

### 9.2 Backlog

- Personas key.
- Empathy mapping.
- As-Is Scenario.
- Ideation.
- Hills.
- Playback.
- Prototypes.
- To Be Scenario.
- Sponsor User.

## 10. Procesos de iteración a realizar

En el proceso global se irá iterando de acuerdo al proceso de IBM Garage.

En el desarrollo se irá iterando en los distintos sprints sobre los objetivos establecidos en el punto 9.