

U.T.N. F.R.L.P. Proyecto 2019 - Grupo 8

# Integración con Diseño de Sistemas N.C.T.S.P.

Sistema de gestión y control de acceso de personal con  
Inteligencia Artificial



## INTEGRANTES

ALBIERO, AGUSTIN 40%

CAPRARO, LUCIANO 10%

GONZÁLEZ ROJAS, SANTIAGO 10%

MERLANO, MARIO 40%

## 1. Justificación de la Metodología elegida

Utilizamos la metodología de trabajo KANBAN ya que nos permite gestionar el trabajo en equipo de una manera rápida, visual y dinámica. Además, utilizamos técnicas de estimación para poder subdividir las tareas correctamente, junto con la división en Springs por bloques de funcionalidad.

Elegimos esta metodología ya que nos parece la más adecuada para el desarrollo de software en equipo, nos está permitiendo poder ver el avance del proyecto de una manera muy visual y rápida. Pudiendo saber al instante en que tarea está trabajando cada integrante del equipo y en cuanto tiempo tiene estimado finalizar su tarea.

## 2. Elección de la versión del producto

La versión del producto a mostrar será correspondiente al entregable de Interfaz Gráfica, módulo de Estadísticas del Personal.

## 3. Especificación de la Plataforma Tecnológica

### a. Justificación del Lenguaje de Desarrollo

Los lenguajes utilizados para el desarrollo del proyecto son Python y Javascript.

Debido a que las Lambdas de AWS brindan la oportunidad de utilizar diferentes lenguajes para su desarrollo la elección fue tomada debido a la afinidad que tenía cada desarrollador con los lenguajes.

Además utilizamos como complemento para el frontend React que el principal motivo de elección fue poder hacer constantes consultas de una manera dinámica a servicios con una mayor velocidad.

Uno de los factores principales del uso de Python fue que la necesidad de invocar los métodos de la librería de OpenCV que está desarrollada en dicho lenguaje el cual nos permite implementar el reconocimiento facial.

### b. Justificación del motor de Base de Datos

dynamoDB nos permite una mayor capacidad de respuesta en contraposición con las bases de datos relaciones, además permite una excelente integración con AppSync, el cual es un servicio de GraphQL que permite la sincronización en tiempo real y un buen manejo de datos offline. Además de una integración completa con las funciones Lambda de Amazon Web Services por pertenecer ambas tecnologías a la misma empresa.

### c. Servidor de Aplicaciones

En nuestro proyecto utilizamos una estructura **serverless** con una estructura FaaS (Function as a Service), utilizando las tecnologías de AWS: Lambdas y el storage de S3.

Elegimos esta configuración ya que nos permite alta escalabilidad y disponibilidad gracias al servicio de AWS (Amazon Web Service). Todo el

procesamiento y entrenamiento de las imágenes se realiza en sus servidores, pudiéndose enfocar más en la lógica e implementación.

Amazon S3 tiene una interfaz de servicios web simple que puede utilizar para almacenar y recuperar cualquier cantidad de datos, en cualquier momento, desde cualquier parte de la web. Utilizamos esta tecnología ya que nos permite una excelente integración con las funciones Lambda. Todas las fotos que capta la cámara se suben a S3 para el procesamiento y entrenamiento del modelo. Una ventaja de esta elección es que AWS nos permite ejecutar las funciones lambda cada vez que se suba una foto a S3 satisfactoriamente.

Para finalizar, queremos volver a remarcar las facilidades que nos brinda AWS para poder realizar el sistema, mediante los servicios de S3 para almacenamiento de las fotos, las funciones Lambda y AWS Rekognition para llevar a cabo el procesamiento y entrenamiento del modelo.

#### **4. Detallar el plan de pruebas**

Como metodología utilizamos el desarrollo en parejas ( Pair programming ). El cual permite el desarrollo de código de calidad, ya que es verificado constantemente por el compañero de equipo.

1. Ingreso manual de datos de empleados.
2. Ingreso manual de imágenes de los mencionados empleados.
3. Testeo de las funciones lambda para procesar dichas imágenes.
4. Observar registros de CloudWatch. Comprobar comportamiento y análisis de errores.
5. Iniciar desde el paso 1. las veces que sea necesaria.

**Aclaración:** este es nuestro actual plan de pruebas, el que efectuamos actualmente. Pero en un futuro estos tests serán automatizados a través del proceso denominado integración continua.

5. Carpeta del drive **“5.Modelado de datos DER”**
6. Carpeta del drive **“6.Modelado de la solución”**
7. Carpeta del drive **“7. Mockups de la interfaz gráfica”**

#### **8. Definir roles de usuario que tendrán acceso al sistema**

En una primera versión de nuestro producto solo habrá un único rol administrador, el cual podrá gestionar toda la aplicación ( dar de alta usuarios, cargarle imágenes para entrenar el modelo, consultar estadísticas, etc)

En una futura actualización se agregará el rol para el personal que esté encargado de estar físicamente en el establecimiento junto con el sistema de cámaras, para que pueda fiscalizar correctamente el ingreso del personal.

9. Carpeta del drive **“9. Product Backlog”**
  - a. **No usamos ninguna ceremonia en específico**

- b. Dividimos el desarrollo por bloque de funcionalidades en Sprints, algunos ejemplos fueron setupear el entorno de desarrollo de Python, luego instalar y poner en marcha la librería OpenCV para reconocimiento facial, luego estuvo la parte de conexión con AWS y sus servicios de S3 y Lambadas,etc.