



Tecnológico de Monterrey

Instituto Tecnológico y de Estudios Superiores de Monterrey

Mapeo de Recursos y Herramientas

TC3007C.501 Inteligencia Artificial Avanzada para la Ciencia de Datos II

Profesores:

Iván Mauricio Amaya Contreras

Blanca Rosa Ruiz Hernández

Félix Ricardo Botello Urrutia

Edgar Covantes Osuna

Felipe Castillo Rendón

Hugo Terashima Marín

Equipo 2

Integrantes:

Luis Ángel Guzmán Iribe – A01741757

Julian Lawrence Gil Soares – A00832272

Alberto H Orozco Ramos – A00831719

27 de Septiembre de 2023

Técnicas y herramientas de ingeniería empleadas

Para el desarrollo de nuestro sistema, empleamos diversas herramientas correspondientes a las diferentes secciones de nuestro sistema. Primeramente podemos hablar de una arquitectura de Modelo-Vista-Controlador como marco conceptual para entender el funcionamiento del sistema, una interfaz de usuario disponible desde la web que se comunica con una API rest que toma la información introducida por el usuario y la analiza utilizando técnicas de Computer Vision y Deep Learning, para guardar esta información en una base de datos.

En cuanto al funcionamiento del sistema de reconocimiento de asistencia y participaciones, hicimos uso de diferentes librerías que proporcionan diferentes herramientas de las áreas de visión computacional, deep learning, detección de objetos, etc... A grandes rasgos, empleamos herramientas que implementan diferentes usos de las redes neuronales convolucionales (CNN), una arquitectura de redes neuronales especializada en el tratamiento de datos en cuadrícula como imágenes y video, y las aplicamos en tareas de reconocimiento de objetos, rostros y poses con la finalidad de generar un informe de utilidad para los usuarios del sistema.

Infraestructura

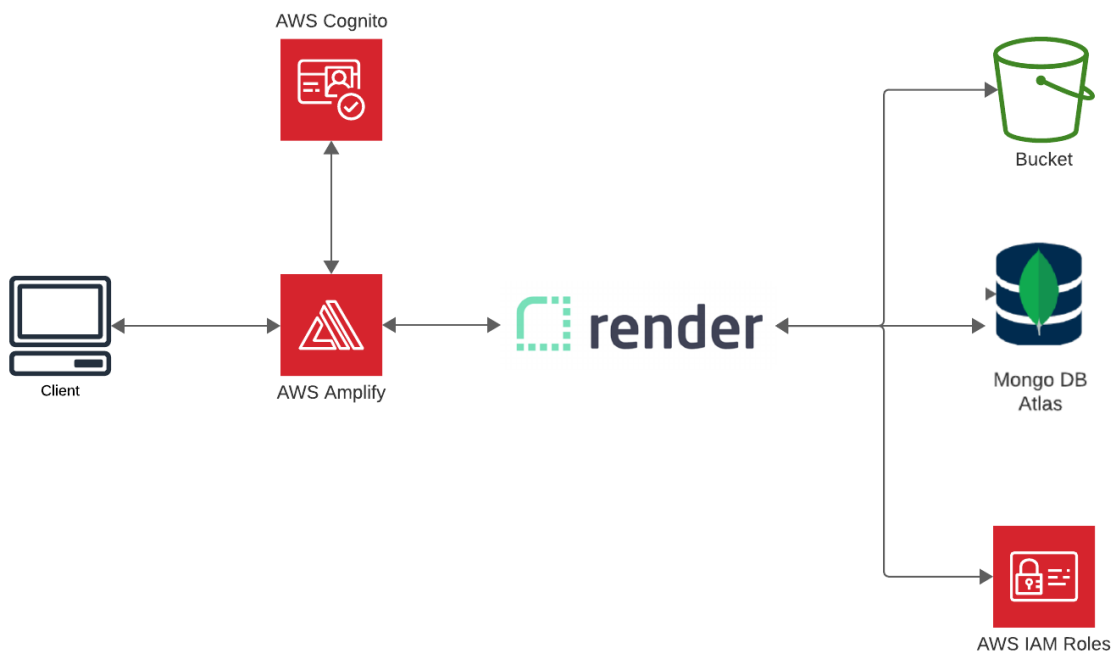


Fig. 7

Para el despliegue de nuestro sistema, empleamos una arquitectura que consta de diferentes servicios alojados en la nube. Empleamos a AWS como nuestro proveedor de nube principal, empleando los servicios de Cognito para el manejo seguro de la autenticación de usuario; Amplify para el alojamiento de nuestra interfaz web; S3 para el almacenamiento de imágenes y video referente a los usuarios y clases; y IAM para el manejo de roles y privilegios dentro de la aplicación.

Se utilizaron servicios auxiliares para complementar la infraestructura como MongoDB como proveedor de base de datos no relacionales en la nube, así como Render, un servicio de alojamiento de backend que permite servir nuestra API Rest para su consumo en la aplicación frontend.

Recursos utilizados (materiales, equipos, reactivos, entre otros)

Nuestro proyecto cuenta con 2 partes cruciales para su funcionamiento, estas son la aplicación web y la parte de detección y reconocimiento de objetos.

La parte de detección y reconocimiento fue implementada utilizando python con librerías de aprendizaje máquina. El primer componente es la librería de face recognition esta librería utiliza un modelo pre-entrenado para detectar caras dentro de una imagen o video. Utilizando este modelo y una imagen de las caras de las personas que debe de encontrar hicimos la función de detección y reconocimiento de rostros. Esto nos permite generar la funcionalidad de asistencia. También es una parte crucial de la detección de participación ya que nos permite acreditar al alumno correcto utilizando su rostro. Para la participación utilizamos dos librerías YOLO 5 y media pipe. YOLO 5 es una librería creada para la detección de objetos, en nuestro proyecto lo empleamos para detectar cuerpos ya que esto nos facilita generar las poses que nuestro algoritmo utiliza para detectar si una persona está participando dentro del aula. La última librería que utilizamos es media pipe, esta nos permite detectar puntos claves en los cuerpos encontrados dentro de un video y así poder generar un algoritmo para detectar si un alumno está participando.

Por parte de la aplicación web, optamos por dividir el proyecto en 2 segmentos principales, una interfaz frontend, y una API rest backend, cada uno construido de manera independiente. Para la interfaz frontend se utilizó la librería de JavaScript Vue.js, la cual permite el desarrollo eficiente de aplicaciones web estables y de calidad; a esta



sección se le agregan algunas dependencias adicionales como PrimeVue, una librería de componentes de interfaz de usuario de alta calidad, TailwindCSS, una librería de clases de utilidad para la estilización del sitio, y demás herramientas del ecosistema de Vue para el manejo de estado dentro de la aplicación, peticiones HTTPS, y autenticación de usuario.

Por último, desarrollamos una API Rest que permite la comunicación entre nuestra base de datos alojada en MongoDB, nuestra interfaz web, y la ejecución de modelos de reconocimiento facial y de poses. Esta API Rest se desarrolló usando FastAPI, una librería de Python que permite el desarrollo sencillo y escalable de esta clase de interfaces.