

Sistema para Facilitar la Toma de Asistencia de Estudiantes por Medio de Reconocimiento Facial – UNITEC

Javier Cano, Estudiante Universitario¹

¹Universidad Tecnológica Centroamericana (UNITEC), Facultad de Ingeniería, Departamento de Ingeniería en Sistemas Computacionales, Tegucigalpa, Honduras, 11101, cano@unitec.edu

Mentor: Román Pineda, M.Sc²

²Universidad Tecnológica Centroamericana (UNITEC), Facultad de Ingeniería, Departamento de Ingeniería en Sistemas Computacionales, Tegucigalpa, Honduras, 11101, roman.pineda@unitec.edu

Abstract– El proceso de toma de asistencia de los estudiantes en UNITEC es algo que sucede en todas las aulas de clase de la universidad. Tradicionalmente se utilizaban carnés y lectores de códigos de barra para marcar al estudiante, sin embargo, no se cuenta con otra alternativa, aun cuando la universidad está departiendo de los lectores de código de barra. La iniciativa del proyecto es proveer otra opción, por medio de un sistema de reconocimiento facial automático. Para demostrar la utilidad de dicho sistema, se desarrolló una herramienta de software capaz de cubrir las necesidades que tienen los profesores para tomar la asistencia de los estudiantes, por medio de tecnologías web y utilizando una cámara inteligente (DeepLens).

Keywords– Artificial intelligence, Facial recognition, DeepLens, Access control.

I. INTRODUCCIÓN

Con el enorme crecimiento de la tecnología, mejoras de hardware, software y mayores capacidades de procesamiento y almacenamiento, se vive en un mundo inundado de información. Esto abre posibilidades a nuevas fronteras tecnológicas que están revolucionando la manera en que se comportan las personas en su día a día. Ya no se necesitará saber claves de acceso complejas, sino que se será capaz de introducir la huella digital o utilizar el rostro para ser identificado.

Paralelo a esto, hoy es más fácil construir un sistema informático gracias al internet y a las herramientas proveídas por plataformas de servicios en la nube como ser Amazon Web Services (AWS). “AWS ofrece servicios que permiten a un individuo o una organización ejecutar prácticamente todo en la nube: desde aplicaciones empresariales y proyectos de investigación hasta juegos sociales y aplicaciones móviles” Stair & Reynolds [1].

Es por esto por lo que desarrollar una base de datos junto con una plataforma web disponible desde internet es posible sin la necesidad de comprar servidores in situ.

En UNITEC, se tiene una enorme oportunidad de mejora del sistema de asistencia, especialmente en la actualidad que se está construyendo el nuevo edificio de ingeniería. UNITEC utiliza carnés para identificar a los estudiantes en muchas áreas del campus universitario, principalmente en la entrada y salida de la universidad y para marcar asistencia en algunas de las aulas, ya que hay aulas donde el marcador de asistencia con carnés no está habilitado. En estas aulas, los docentes pasan asistencia de manera manual.

II. PLANTEAMIENTO DEL PROBLEMA

A. Precedentes del Problema

El control de acceso es un problema muy común en las organizaciones, ya sea de acceso físico o de acceso virtual, dependiendo de la organización. Este puede ser tedioso, especialmente si se quiere asegurar la seguridad de las personas sin restringir el acceso inadecuado a miembros de la organización.

UNITEC actualmente hace uso de carnés con códigos de barra para: otorgarles acceso a los estudiantes a la universidad y tomar asistencia en las clases que cursan los estudiantes. En los casos donde el marcador de asistencia del aula de clases no esté habilitado, se utiliza la plataforma de asistencia para que el docente marque a todos los estudiantes de manera manual.

B. Definición del Problema

El uso de carnés trae muchas desventajas: se tienen que realizar carnés de plástico para miles de estudiantes que ingresan a la universidad, si el estudiante pierde el carné ya no tiene acceso a la universidad o al sistema de asistencia automático, el estudiante puede olvidarse de pasar el carné para marcar su asistencia, el código de barra del carné se deteriora con el tiempo y con el constante uso, entonces se ve obligado a sacar otro carné, el carné es de fácil duplicación.

B. Justificación

El reconocimiento facial se aplica perfectamente al problema, ya que agiliza el proceso de tomar asistencia, aumenta la seguridad y la confiabilidad del mismo (como ser otros estudiantes pasando el carné de sus compañeros), y se deshace de los carnés en su totalidad. Con reconocimiento facial, los estudiantes simplemente entran al aula y el sistema se encargaría automáticamente de marcar a todos los que estén presentes en las respectivas secciones.

Lo que se quiere lograr con el proyecto es reducir la fricción entre los estudiantes o docentes y el sistema de asistencia, agilizando más el proceso, al igual que aumentar la seguridad y confiabilidad del sistema de asistencia, de manera que un estudiante no pueda ser marcado por otra entidad además de la misma cámara que realiza el monitoreo.

También se espera que este sistema sea beneficioso para la universidad, ahorrando valioso tiempo de clases de los profesores ya que no necesitarían pasar asistencia de forma manual.

III. HERRAMIENTAS

A. Glosario

- 1) Apache JMeter: programa diseñado para medir el rendimiento de aplicaciones. Originalmente fue diseñado para probar aplicaciones web, pero desde entonces se ha expandido a otras funciones de prueba, Apache Software Foundation [2].
- 2) API REST: interfaz de programación de aplicaciones de transferencia de estado representacional, es una manera de diseñar APIs, transfiriendo información estructurada de una localización web a otra por medio del Protocolo de Transferencia de Hipertexto (HTTP), típicamente en forma de Lenguaje de Marcado Extensible (XML) o Notación de Objeto JavaScript (JSON), por medio de datos marcados con etiquetas de Lenguaje de Marcado de Hipertexto (HTML), donde el servicio web se encuentra disponible en una dirección específica, Schneider [3].
- 3) AWS: plataforma de Amazon para servicios de computación en la nube, Amazon Web Services [4].
- 4) AWS DeepLens: cámara de video de Amazon completamente programable. Esta es capaz de recopilar y analizar datos más cerca de la fuente de información, reaccionar de manera autónoma a los eventos locales y se comunicarse de forma segura en las redes, Amazon Web Services [5].
- 5) AWS Lambda: servicio informático que le permite ejecutar código sin aprovisionar ni administrar servidores, Amazon Web Services [4].
- 6) AWS SageMaker: sirve para entrenar y validar un modelo de red neuronal convolucional (CNN) o importar un modelo previamente entrenado, Amazon Web Services [4].

- 7) IA: campo de la computación donde el sistema informático toma las características de la inteligencia humana, Stair & Reynolds [1].
- 8) Nube: la computación en la nube se refiere a un entorno informático donde el software y el almacenamiento se proporcionan como un servicio de internet Stair & Reynolds [1].
- 9) Visión por computador: campo muy amplio que barca una gran variedad de formas de procesamiento de imágenes y una sorprendente diversidad de aplicaciones, Goodfellow, Bengio, & Courville [6].

El sistema hace gran uso del ecosistema de AWS ya que se está utilizando el DeepLens. Este hace uso de AWS SageMaker para desarrollar el modelo de aprendizaje y AWS Lambda para controlar los eventos. Además, se hace uso de una instancia de Amazon RDS como la base de datos relacional, y Amazon S3 para el almacenamiento de rostros e imágenes de interés y capturados por la cámara. Los rostros e imágenes son procesados por Amazon Rekognition, y la traducción de rostro a número de cuenta se guarda en una tabla de DynamoDB, una base de datos no relacional ofrecida por Amazon. Amazon EC2 se encarga de la máquina virtual que contiene el servidor web.

IV. METODOLOGÍA

Se decidió por desarrollar la herramienta de software como varios módulos web, ya que esto da la facilidad de poder ingresar al sistema por medio de cualquier dispositivo que sea capaz de utilizar un navegador web, ya sea computadora de escritorio, laptop o dispositivo móvil (smartphone). Esto significa que el interfaz deberá ser capaz de adaptarse a cualquier tamaño de pantalla.

Además del interfaz web, se desarrollo una API REST para comunicarse con el sistema. Esta API está diseñada para cumplir con todas las necesidades que podrían tener los usuarios. Ya que el API utiliza HTTP, se usan los métodos GET, POST, PUT y DELETE para hacer distintas acciones. De igual manera se utilizan los parámetros dentro de la ruta o URL y el cuerpo (body) para enviar toda la información necesaria, recordando que el servidor web no guarda ningún tipo de estado, y es responsabilidad del cliente notificarle e identificarse con el servidor.

A. Arquitectura

La arquitectura del sistema es una basada en servicios en la nube, especialmente de AWS. La espina dorsal responsable de recibir todas las solicitudes entrantes a el sistema es una instancia de Amazon EC2, la cual tiene un servidor web encargado de responder a todas las solicitudes. Esta se comunica con otros servicios, como ser las bases de datos y el servicio de almacenamiento de archivos.

Por otra parte, la cámara DeepLens recibe el modelo de aprendizaje de reconocimiento facial de SageMaker. La cámara esta relacionada con un aula de clases, y en caso de que se abra asistencia en esta aula de clases, se inicia la captura de video y el reconocimiento de rostros de los estudiantes. Todo este proceso se ejecuta mientras la asistencia siga abierta, en caso de que esta cierre entonces la cámara vuelve nuevamente a un estado de escucha (ejemplo, ver Fig. 1).

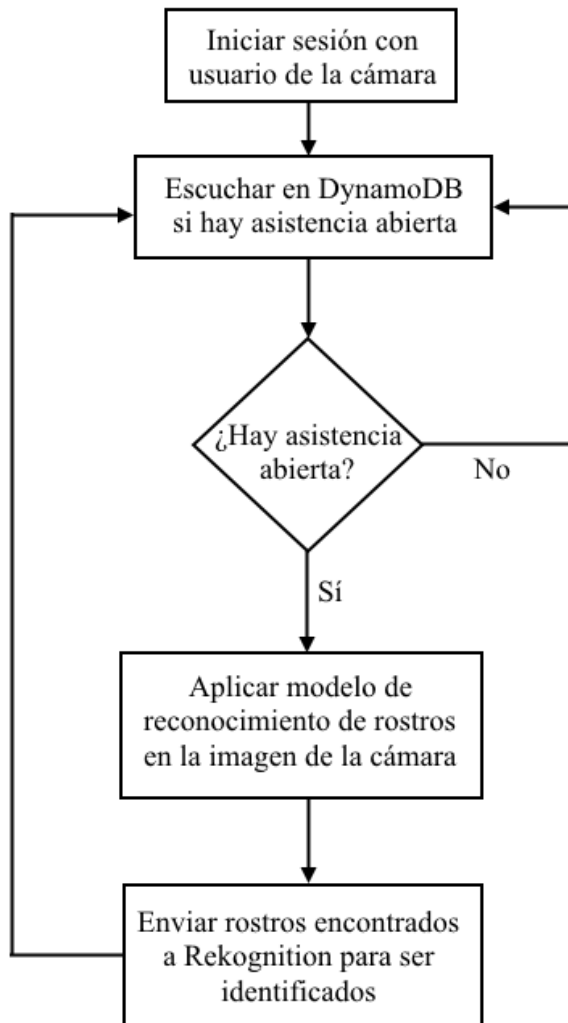


Fig. 1 Diagrama de flujo explicando el proceso del DeepLens.

Como se puede observar en la Fig. 1, la cámara tiene que iniciar sesión antes de realizar cualquier tarea. El iniciado de sesión ocurre de manera automática, ya que se tiene que configurar un usuario y contraseña en sus variables de entorno. Este se realiza de manera segura por medio de HTTPS.

B. Base de Datos

Para el diseño de la base de datos se decidió utilizar una base de datos relacional, en este caso MySQL. El sistema se beneficia mucho de una base de datos relacional ya que se garantiza la integridad de los datos, a diferencia de bases de datos no relacionales donde no hay una estructura bien definida para los datos.

La base de datos sigue los principios de normalización. Coronel, Morris, y Rob [7] definen la normalización como “un proceso para evaluar y corregir estructuras de tablas a fin de minimizar redundancias de datos, con lo cual se reduce la probabilidad de anomalías de datos”.

Además de la base de datos relacional, se hace uso de DynamoDB para dos tablas adicionales (una tabla para traducir los rostros de estudiantes a su número de cuenta, y otra tabla para ver qué aulas tienen asistencia abierta). Ya que las tablas que se implementan son pares de clave-valor, estas tablas presentan menor costo de mantenimiento y operación en una base de datos no relacional, y no se benefician de una base de datos robusta como MySQL.

C. Pruebas de estrés

El sistema está compuesto en gran parte por el servidor web que se encuentra en una instancia EC2 de AWS. Para poder medir la capacidad que tiene el sistema de soportar carga, se hace uso de la herramienta Apache JMeter. Primero se decidió realizar una prueba con carga moderada, haciendo 7 tipos de consultas distintas al servidor, seleccionadas de manera arbitraria:

- Cargar la página web
- Ingresar a la página con un usuario válido
- Tratar de ingresar a la página con un usuario incorrecto
- Obtener información de secciones
- Obtener información de clases

Primero se realizó una prueba con una carga moderada, teniendo 3 hilos de procesamiento, cada uno realizando 20 consultas. En total se realizaron 420 consultas al servidor. En la Fig. 2 se puede observar un gráfico de los resultados obtenidos, donde las mediciones se realizan en milisegundos.

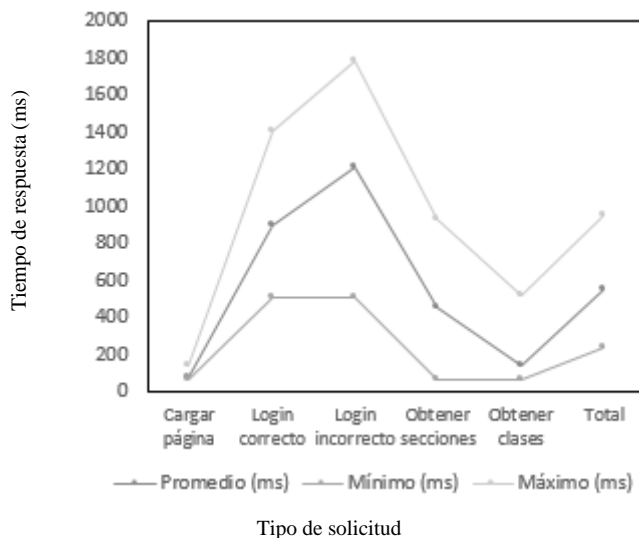


Fig. 2 Pruebas de estrés al servidor con 3 hilos de ejecución.

Como se puede observar, las operaciones más costosas para el servidor son las operaciones de autenticación. Esto se debe a que hay que encriptar las contraseñas ingresadas por el usuario para poder ingresar al sistema, y esto toma de mucho procesamiento.

Seguido de la autenticación, obtener la información de secciones es la segunda operación más costosa ya que las secciones tienen mucha información.

Luego se volvió a correr la misma prueba, pero esta vez con 10 veces más el número de hilos (30 hilos en total). En total se realizaron 3150 consultas al servidor. En la Fig. 3 se muestran los datos recolectados.

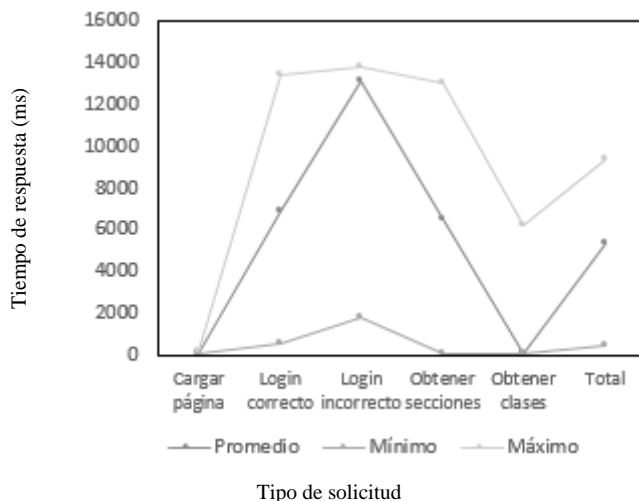


Fig. 3 Pruebas de estrés al servidor con 30 hilos de ejecución.

Como se puede observar en el gráfico, aunque las consultas de obtención de datos no se ven muy afectadas, los tiempos de ejecución se elevan llegando a tardar hasta casi 14 segundos por una solicitud de autenticación.

Aunque en un ambiente de producción es muy poco probable que 30 usuarios intenten acceder a los mismos recursos al mismo tiempo de manera continua, el sistema aún puede lidiar moderadamente con una carga de tan alta magnitud.

Cabe destacar que, en ambas pruebas realizadas, un 100% de las solicitudes recibidas por el servidor fueron resueltas de manera exitosa.

V. CONCLUSIONES

1. Se desarrolló un sistema web de asistencia con reconocimiento facial, haciendo uso de AWS DeepLens y muchas más herramientas de AWS para desarrollar una infraestructura totalmente en la nube. El sistema es capaz de reemplazar o depender menos de los carnés estudiantiles y agilizar el proceso de asistencia para estudiantes y docentes.
2. Se logró crear una base de datos con la capacidad de guardar toda la información necesaria para que el sistema pueda operar exitosamente a través del tiempo, siendo esta capaz de funcionar a lo largo de múltiples periodos académicos.
3. Se alcanzó a definir y entrenar un modelo de aprendizaje capaz de realizar la tarea de reconocimiento facial, logrando identificar los rostros de los estudiantes y obteniendo información de interés para el sistema.
4. Se desarrolló un módulo web para estudiantes, docentes y personal administrativo para poder ingresar al sistema de asistencia con facilidad desde cualquier dispositivo (laptop, de escritorio o móvil).
5. Se documentó todo el desarrollo del sistema para que futuras personas que busquen estudiar o modificar el mismo lo puedan realizar con facilidad, o incluso junto con el código fuente puedan replicar el sistema en su totalidad.
6. No se logró comparar la satisfacción de los estudiantes y los docentes con el nuevo sistema de asistencia comparado con el actual sistema de asistencia debido a la problemática sanitaria mundial por la enfermedad infecciosa COVID-19.
7. Se logró realizar pruebas de estrés en el sistema una vez este fue desarrollado, logrando comprobar la robustez del mismo ante un escenario de alta carga de procesamiento.

VI. RECOMENDACIONES

1. Se recomienda que, al momento de definir, entrenar e implementar un modelo de aprendizaje de reconocimiento facial con AWS DeepLens, se haga uso de muchas de las herramientas que ofrece AWS como ser SageMaker y Rekognition, ya que, con una arquitectura bien diseñada, estas herramientas pueden integrarse y complementarse muy bien, y llegan a cubrir todas las necesidades de un sistema de reconocimiento facial, por ejemplo SageMaker se encarga de ofrecer un modelo de aprendizaje de reconocimiento facial ya entrenado y Rekognition junto con DeepLens se encargan de implementar este modelo con el sistema de asistencia.
2. Se recomienda que al diseñar y desarrollar la arquitectura que cumpla con las necesidades que tiene la universidad, se haga un énfasis en los procesos que la misma tiene en pie para cumplir con sus obligaciones, como ser el proceso de toma de asistencia, analizar cuáles son los recursos necesarios para poder crear un software aislado del sistema actual y poner mucha importancia en los requisitos de información, qué tipo de datos se necesitan en las bases de información, y cómo se obtendrán esos datos.
3. Se sugiere que se retome la pregunta de si se sienten beneficiados los estudiantes y los docentes con un sistema de reconocimiento facial en contraste con el sistema de asistencia actual, y tratar de hacer la migración del proceso de asistencia en el sistema actual a un nuevo sistema lo más fluida posible ya que cambios muy bruscos en los procesos pueden impactar a los usuarios de una manera negativa.
4. Es importante tener una arquitectura del sistema sólida y bien definida antes de iniciar el proceso de desarrollo, ya que esta es el plano para cómo se integrarán distintas partes del sistema en un muy alto nivel, y puede llegar a eliminar mucha incertidumbre al momento del desarrollo.

VII. REFERENCIAS

- [1] Stair, R. M., & Reynolds, G. W. (2016). Principios de Sistemas de Información (10 ed.). Cengage Editores. Recuperado el 30 de enero de 2020, de <https://cengageeditores-ip.vitalsource.com/#/books/9786075264059/>
- [2] Apache Software Foundation. (2020). Apache JMeter. Obtenido de Apache JMeter: <https://jmeter.apache.org/>
- [3] Schneider, G. P. (2013). Comercio Electrónico. México D.F.: Cengage Editores.
- [4] Amazon Web Services. (2020). Servicios de informática en la nube. Recuperado el 14 de febrero de 2020, de AWS: <https://aws.amazon.com/es/>
- [5] Amazon Web Services. (2020). AWS DeepLens Developer Guide. Recuperado el 1 de febrero de 2020, de AWS: <https://docs.aws.amazon.com/deeplens/latest/dg/deeplens-dg.pdf>
- [6] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. Massachusetts: MIT Press. Recuperado el 1 de febrero de 2020, de Deep Learning: <https://www.deeplearningbook.org/>

- [7] Coronel, C., Morris, S., & Rob, P. (2011). Bases de datos, diseño, implementación y administración. México, D.F.: Cengage Editores.