



**Universidad de Montemorelos**

**Facultad de Ingeniería y Tecnología  
Ingeniería en Sistemas Computacionales**

# **Desarrollo de software para el reconocimiento de edificios utilizando Deep Learning**

**Harry Daniel Vargas Pérez  
1150533**

**Asesor: Ing. Ignacio Cruz Domínguez**

**Montemorelos, Nuevo León, México  
22 de abril de 2021**

## **Resumen**

Aquí va el abstract.

## **Palabras Clave**

Keywords

## ÍNDICE

<b>I</b>	<b>INTRODUCCIÓN</b>	<b>1</b>
I-A	Antecedentes . . . . .	1
I-B	Definición del problema . . . . .	1
I-C	Justificación . . . . .	1
I-D	Objetivos . . . . .	2
I-D1	Objetivos generales . . . . .	2
I-D2	Objetivos específicos . . . . .	2
<b>II</b>	<b>FUNDAMENTOS TEÓRICOS</b>	<b>2</b>
II-A	MARCO TEÓRICO . . . . .	2
II-A1	Inteligencia Artificial . . . . .	2
II-A2	Redes Neuronales . . . . .	2
II-A3	Deep Learning . . . . .	2
II-A4	Python . . . . .	2
II-A5	Anaconda . . . . .	2
II-A6	Tensorflow . . . . .	3
II-A7	Métricas de Evaluación . . . . .	3
II-A8	JavaScript . . . . .	3
II-A9	Node.js . . . . .	3
II-A10	API REST . . . . .	3
II-A11	Base de Datos noSQL . . . . .	3
II-A12	MongoDB Atlas . . . . .	3
II-A13	Heroku . . . . .	3
II-A14	Framework . . . . .	3
II-A15	React . . . . .	3
II-A16	React Native . . . . .	3
II-B	ESTADO DEL ARTE . . . . .	4
<b>III</b>	<b>DESARROLLO</b>	<b>4</b>
III-A	Creación del dataset . . . . .	4
III-A1	Recolección de imágenes . . . . .	4
III-A2	Organización de las imágenes . . . . .	4
III-B	Creación de modelo de IA . . . . .	4
III-B1	Preparación del entorno Python . . . . .	4
III-B2	Carga del dataset . . . . .	4
III-B3	Estructura de la red neuronal . . . . .	4
III-B4	Pruebas . . . . .	5
III-B5	Entrenamiento del modelo . . . . .	5
III-B6	Exportación . . . . .	5
III-C	Desarrollo de un API REST . . . . .	5
III-C1	Creación . . . . .	5
III-C2	Función del servicio . . . . .	5
III-D	Administración de ubicaciones . . . . .	5
III-D1	Funcionalidad . . . . .	6
III-E	Implementación del modelo en la app . . . . .	6
III-E1	Preparación del entorno react-native . . . . .	6
III-E2	Importación del modelo . . . . .	6
III-E3	Configuración de la cámara . . . . .	6
III-E4	Predicción . . . . .	6
III-E5	Despliegue de la información . . . . .	6
III-E6	Condicionamiento . . . . .	6
<b>IV</b>	<b>RESULTADOS</b>	<b>6</b>
<b>V</b>	<b>CONCLUSIONES</b>	<b>6</b>
	Referencias	7

# Desarrollo de software para el reconocimiento de edificios utilizando Deep Learning

Harry Daniel Vargas Pérez e Ignacio Cruz Domínguez

*Facultad de Ingeniería y Tecnología, Universidad de Morelos, México*

## I. INTRODUCCIÓN

### A. Antecedentes

Diariamente las empresas reciben multitud de visitas, algunas de ellas de manera personal, otras de tipo empresarial y otras más de tipo comercial, ya sea en mayor o menor medida. Independientemente de su formalidad y relevancia, la empresa puede aprovechar estas visitas para generar cercanía y potenciar su marca con posibles clientes a futuro dependiendo del trato y la impresión que causen. En beneficio de la imagen de la empresa, las visitas deben ser atendidas de acuerdo a las características propias de la empresa u organización para dar una impresión correcta del funcionamiento de la misma. Para atender a las visitas, las organizaciones dedican distintos recursos como tiempo, dinero y esfuerzo, además de requerir una preparación del recurso humano para poder brindar una atención de calidad y adecuada según sea la visita que reciban. A los visitantes les interesa la información acerca de lo que observan durante el recorrido, como los principales edificios de la organización o los espacios en los que se encuentran. Distintas universidades (agregar referencias) también ofrecen recorridos por sus instalaciones para que posibles nuevos alumnos se interesen en realizar su preparación académica con ellos; además de otros recorridos a grupos en espacios como bibliotecas o facultades específicas que dan más ayuda al interesado o interesados a poder hacer una evaluación más completa e integral de la universidad que están visitando. También algunas universidades están obteniendo por ofrecer un recorrido virtual a través de sus instalaciones [1], [2], [3], esta es una gran oportunidad que puede ser aprovechada más en el tiempo que se está viviendo debido al confinamiento por causa del COVID-19, este recorrido atraviesa los lugares más referentes del plantel mediante un video o un mapa utilizando Google Maps para poder desplazarse sobre las áreas determinadas. Existen plataformas como civitatis [4] que ofrecen recorridos a lugares alrededor del mundo de manera precencial que incluyen visitas a universidades, museos, zonas turísticas, entre otras. Dicha plataforma también está bajo las recomendaciones e instrucciones de parte del personal médico en cada lugar.

### B. Definición del problema

Con la Revolución Industrial 4.0 [5], la necesidad de implementar tecnología en las organizaciones ha impulsado

Harry Vargas es estudiante de la Facultad de Ingeniería y Tecnología en la Universidad de Morelos, e-mail: 1150533@alumno.um.edu.mx.

Ignacio Cruz, Ing. es profesor en la Facultad de Ingeniería y Tecnología en la Universidad de Morelos., Nuevo León, Mexico, e-mail: icruz@um.edu.mx.

también al Internet de las Cosas (IoT por sus siglas en inglés *Internet of Things*), donde básicamente todo con lo que nos relacionamos estaría conectado a internet, almacenando datos a la nube [6]. Al realizar un recorrido es interesante e importante para el visitante, obtener información y realimentación acerca de los principales lugares que se visitan. En este proyecto se desarrolla una aplicación móvil que provee información y realimentación acerca de los principales edificios e instalaciones de la Universidad de Morelos. En esta aplicación el usuario hace uso de la cámara para poder capturar una imagen del edificio sobre el cual se requiere información y tomarla como entrada. Para proveer ésta realimentación se utilizan técnicas de reconocimiento basadas en aprendizaje profundo (*Deep Learning*) para lograr una identificación aceptable. La implementación de esta característica permite lograr una cercanía a los valores e información importante de la institución. La aplicación es relevante porque permite que los nuevos visitantes o grupo de visitantes, puedan tener información adicional además de la provista por el personal que guía el recorrido, el cual tiene que ser instruido y capacitado previamente. Este proceso requiere invertir tiempo de preparación en conocer bien los lugares e historias. En ocasiones, para el personal guía, puede resultar monótono repetir las descripciones e información, para lo cual una aplicación no, además de dar un plus tecnológico en una actividad que pareciera monótona hasta el momento.

### C. Justificación

Aprovechando el auge y accesibilidad de los dispositivos móviles y basado en la necesidad presentada, en este trabajo se desarrolla un software que ayuda en la guía de recorridos, al identificar principales sitios, edificios e instalaciones, tarea que es realizada por personal humano de manera continua. Esta actividad se ve pausada momentáneamente debido al confinamiento impuesto por las autoridades sanitaria por causa de la pandemia del COVID-19. Se desarrolla una aplicación móvil que funcione en las plataformas de Android y iOS, que permite identificar los departamentos y facultades dentro de la institución utilizando algoritmos de Inteligencia Artificial y Deep Learning para conseguir una predicción aceptable. Ésta aplicación es útil para el departamento de promoción y reclutamiento en la Universidad de Morelos al llevar a cabo su actividad de guía que es realizada de forma recurrente, facilitando que los visitantes obtengan la información de los edificios a partir de una entrada visual mediante la cámara del dispositivo móvil. La información provista por la aplicación móvil enriquece la experiencia y la relación con la institución.

El visitante puede obtener información relevante, no sensible o pública de cada departamento o facultad como horario de atención, teléfono de contacto, las carreras con las que cuenta cierta facultad, historia, entre otras.

#### D. Objetivos

1) *Objetivos generales:* Se espera obtener una aplicación que permita ofrecer una ayuda a una persona que desconozca la Universidad de Morelos, sumado a esto la disminución del trabajo de acompañamiento en los recorridos y la oportunidad de hacer más didáctico y tecnológico el recorrido a la institución. Esta innovación permitiría a la universidad contar con una aplicación descriptiva de sus instalaciones, además de la oportunidad de compartir la información específica de los horarios de atención en cada una e información de contacto.

##### 2) *Objetivos específicos:*

- Identificar instalaciones y edificios importantes del campus de la Universidad de Morelos mediante técnicas de Deep Learning, a partir de la entrada proveniente de la cámara del dispositivo móvil del usuario.
- Construir un clasificador que toma la imagen y predice a qué facultad o edificio corresponde.
- Describir el proceso de clasificación e identificación, además de proveer medidas del rendimiento y pruebas.
- Desarrollar una aplicación que utiliza el modelo obtenido utilizando React Native.
- Describir las herramientas de desarrollo y el proceso llevado a cabo en la integración

## II. FUNDAMENTOS TEÓRICOS

### A. MARCO TEÓRICO

#### 1) *Inteligencia Artificial:*

La Inteligencia Artificial (IA) es parte de las ciencias de la computación que se encarga de los diseños de sistemas inteligentes, los cuales contienen características que son relacionadas con la inteligencia en la conducta humana [7]. La IA ha tenido múltiples aplicaciones no solo en la informática o robótica, también está presente actualmente en su implementación en áreas empresariales para poder realizar predicciones de resultados en tiempo real y también para procesar gran cantidad de datos. A su vez, la inteligencia artificial ha tenido un gran progreso gracias al aumento en los recursos económicos, tecnológicos y de datos para poder trabajar y realizar más investigaciones en este campo[8].

#### 2) *Redes Neuronales:*

Una de las definiciones que se puede utilizar para dar significado a las redes neuronales artificiales puede ser: “Una red neuronal es un modelo computacional, paralelo, compuesto de unidades procesadoras adaptativas con una alta interconexión entre ellas” [9] Básicamente la comparativa con una red neuronal biológica se refleja en la distribución de las operaciones que se realizan en el conjunto de los elementos

básicos que representan las neuronas. Estas neuronas están interconectadas entre sí y se denominan pesos sinápticos, estos pesos varían con el paso del tiempo de entrenamiento. Así se puede definir el aprendizaje de una red neuronal como el proceso en el cual se modifican las conexiones entre las neuronas para poder realizar un objetivo definido.

Las redes neuronales artificiales tienen una topología que consiste en la forma en que las neuronas están organizadas, de manera que crean agrupaciones (capas) más cerca de la entrada o de la salida de la red. Algunos de los parámetros fundamentales en estas redes son el número de capas formadas, la cantidad de neuronas en cada capa, el grado de conectividad y el tipo de conexiones entre las neuronas [10].

#### 3) *Deep Learning:*

El *deep learning* (aprendizaje profundo) se asocia a la idea de “imitar el cerebro a partir del uso de hardware y software, para crear una inteligencia artificial pura”, la cual usa una capacidad de abstracción jerárquica para poder separar los datos de entrada y poder obtener características de los datos de un nivel de complejidad alto [11]. En otra definición más sencilla de entender, el *deep learning* es el aprendizaje que una máquina puede hacer para cumplir con un determinado objetivo, obteniendo datos ocultos o profundos desde una red neuronal [12].

#### 4) *Python:*

Python [13] es un lenguaje dinámico y fuertemente tipado, además es orientado a objetos [14]. Python presenta ventajas ante otros lenguajes de programación como:

- Es un lenguaje expresivo, el código suele ser más corto que utilizando otros lenguajes, además de ser considerado un lenguaje de alto nivel.
- La lectura del código resulta fácilmente legible, ya que su sintaxis permite escribir de manera elegante.
- Python posee múltiples estructuras de datos que pueden ser utilizadas de manera simple.

Estas son algunas de las ventajas que presenta trabajar con Python, además de su continua actualización debido al desarrollo de mantenimiento que tiene [15].

#### 5) *Anaconda:*

Anaconda [16] es una interfaz gráfica de usuario (GUI) que permite iniciar aplicaciones y administrar fácilmente paquetes, entornos y canales de conda sin usar la línea de comandos [17]. Está disponible tanto para Windows, como para MacOS y Linux. Anaconda cuenta con distintos paquetes, en esta investigación se utilizó la versión “*Individual Edition*” [18]. Este paquete cuenta con *Anaconda Navigator* la cual es una GUI de escritorio, que permite la creación y administración de ambientes, además de la instalación de algunas herramientas de desarrollo como lo es *Jupyter Notebook* [?], que es una aplicación web que permite la escritura de código en Python y es instalada predeterminadamente junto con Anaconda. Dicha herramienta es la utilizada para el desarrollo de la parte de inteligencia artificial en este proyecto.

#### 6) *Tensorflow*:

Tensorflow [19] es una librería de software de código abierto que es usada para implementar sistemas de aprendizaje automático [14]. Esta plataforma fue desarrollada por Google para compartir con otros investigadores y que desarrollen sistemas capaces de crear y entrenar redes neuronales para reconocer patrones; además de que provee distintas herramientas, librerías y recursos de la comunidad para poder realizar innovaciones en el campo del aprendizaje automático. Tensorflow es capaz de ser utilizado en distintos lenguajes de programación, en esta investigación es utilizado con el lenguaje Python y es instalado dentro de un ambiente gestionado por Anaconda.

#### 7) *Métricas de Evaluación*:

Accuracy  
Loss

#### 8) *JavaScript*:

JavaScript (*JS*) es un lenguaje de programación muy utilizado en los últimos años debido a su flexibilidad, es un lenguaje multiparadigma y muchos de los sitios web lo utilizan para las animaciones y más recientemente también es utilizado para back-end trabajando del lado del servidor y no solamente del lado del cliente [20]. Es un lenguaje de script, lo que significa que no ocupa ser compilado, solamente interpretado [21].

#### 9) *Node.js*:

En su sitio web oficial, se define a Node.js como un entorno de ejecución de JavaScript E/S asíncrono y que está diseñado para crear aplicaciones network escalables [22]. Utiliza el motor de JavaScript V8 de Google [23]. Entre sus características destaca que es de código abierto y multiplataforma, es decir, que puede ser utilizado por distintos sistemas como Windows, Linux, MacOS, entre otros. El lenguaje utilizado en Node.js es JavaScript puro, con la ventaja que permite usar dicho lenguaje del lado del cliente y también del servidor [20]. Node.js es utilizado para el desarrollo del *back-end* en diversos proyectos actualmente.

Express.js [24] es un marco web utilizado en el desarrollo de aplicaciones web minimalistas y comúnmente utilizado para la creación de APIs (*Application Programming Interface*), ya que crea un servidor de HTTP que permite habilitar un puerto para la comunicación.

Mongoose [25] es una herramienta para el modelado de objetos de MongoDB, soporta promesas y llamadas de respuesta (*callbacks*). También funciona para crear una conexión con la base de datos.

#### 10) *API REST*:

Se denomina API por sus siglas en inglés *Application Programming Interface*. Es un conjunto de métodos o *endpoints*

que son ofrecidos por una biblioteca de código, que será utilizado por otro software, demostrando así la capacidad de comunicación entre distintos componentes de software. REST también es un acrónimo en inglés (*Representational State Transfer*) y refiere a un tipo de arquitectura que sigue cuatro principios fundamentales:

- Incluye el uso de métodos HTTP: aquí se establece una relación uno-a-uno entre los métodos HTTP y las operaciones de un CRUD (*Create, Read, Update y Delete*), siendo el verbo POST utilizado para crear un registro, GET para obtenerlo, PUT para actualizar y DELETE para eliminarlo. Estas 4 funciones básicas son importantes ya que engloban las acciones necesarias al manejar información.
- No mantiene estados (*stateless*): cada petición debe de ser completa e independiente, que contenga todos los datos para ser completada la orden.
- Utiliza URIs en forma de directorio para realizar una operación definida: las rutas definidas para los endpoints deben de ser intuitivas, se utiliza en estilo de directorio para reconocer la jerarquía.
- Transfiere XML, JSON o ambos: el formato JSON (*JavaScript Object Notation*) es un formato de archivo simple para describir datos jerárquicamente [26], [23].

#### 11) *Base de Datos noSQL*:

Las bases de datos NoSQL, son bases de datos de código abierto, sin esquema, escalables horizontalmente (se pueden agregar más servidores para trabajar en la misma carga de trabajo) y de alto rendimiento. Estas características las hacen muy diferentes de las bases de datos relacionales, la opción tradicional para datos espaciales. Las bases de datos NoSQL están bien adaptadas para manejar los desafíos típicos de big data, incluidos el volumen, la variedad y la velocidad. Por estas razones, las industrias privadas los adoptan cada vez más y se utilizan en la investigación. Se ha hecho más común utilizar las base de datos NoSQL en los últimos años debido a su capacidad para administrar datos no estructurados (por ejemplo, datos de redes sociales) [27].

#### 12) *MongoDB Atlas*:

F

#### 13) *Heroku*:

F

#### 14) *Framework*:

F

#### 15) *React*:

Axios

#### 16) *React Native*:

F

## B. ESTADO DEL ARTE

### III. DESARROLLO

Para el desarrollo de este proyecto se dividieron en etapas para tener entregables del desarrollo de manera controlada.

A continuación, se muestran los siguientes puntos para el desarrollo del proyecto:

#### A. Creación del dataset

Para poder crear un modelo de inteligencia artificial capaz de reconocer los edificios en la Universidad de Montemorelos, es necesario previamente recolectar imágenes que el modelo necesitará para poder entrenarse y ser capaz de reconocer edificios.

1) *Recolección de imágenes:* Las imágenes utilizadas son de la fachada de los edificios como las facultades, departamentos y otros espacios importantes, esto porque es lo que el usuario verá cuando desee saber una predicción del lugar. Se hizo la recolección de N imágenes de cada edificio (clase), siendo N edificios reconocidos. Esto hace un total de N imágenes capturadas para crear el modelo. Estas imágenes son fotografías tomadas por un dispositivo móvil (iPhone 12 Pro Max), esto debido a que el modelo sería utilizado en una aplicación móvil y las imágenes con las que sería probado el modelo serían capturadas por dispositivos móviles. Todas las imágenes fueron tomadas en distintos ángulos, altura y condiciones del clima en dirección a la fachada del edificio; todas las imágenes fueron tomadas durante el día.

2) *Organización de las imágenes:* Las imágenes fueron exportadas a formato .jpeg para poder ser utilizadas en el modelo realizado. Todas las imágenes fueron separadas en distintos directorios (carpetas), el nombre de la carpeta es dependiendo del edificio al que pertenecen las fotografías.

#### B. Creación de modelo de IA

Para poder reconocer edificios en una imagen, se crea un modelo de clasificación con redes neuronales convolucionales profundas. A continuación se describe la manera en como fue creado y entrenado el modelo:

1) *Preparación del entorno Python:* Para la realización de esta investigación se creó un ambiente de desarrollo en Anaconda 1.10.0. Junto con esto, se instala la versión 3.7.10 del lenguaje de programación Python que es utilizada en el desarrollo del código fuente, también la versión 2.3.0 de tensorflow que permite la creación de la red neuronal, además de instalar TensorflowJS en su versión 3.3.0 para exportar el modelo y ser utilizado en la aplicación. Para la escritura del código se utilizó la herramienta Jupyter Notebook contenida por Anaconda. Dentro de Tensorflow se encuentra la librería

de Keras que ha sido utilizada para hacer uso de las capas utilizadas en el modelo.

El proyecto está estructurado en un directorio, en él se encuentra un archivo .ipynb (formato de Jupyter Notebook) con el código que recolecta el dataset, diseña la red neuronal y entrena el modelo que posteriormente es exportado. En una carpeta al interior del directorio principal se encuentran las imágenes a utilizar para el entrenamiento del modelo, descrito previamente en el apartado 3.1.

2) *Carga del dataset:* Para cargar las imágenes contenidas en un folder dentro del directorio, se utiliza una herramienta contenida dentro del paquete de keras, esta herramienta permite particionar el total de imágenes recolectadas entre un paquete de entrenamiento y uno de prueba, el porcentaje de división es 80% para entrenamiento y 20% para pruebas. Dichas imágenes son tomadas en una dimensión fija (224 x 224).

3) *Estructura de la red neuronal:* Para crear la red neuronal se utiliza la estructura Secuencial contenida dentro de Keras. Se comienza con una capa donde se aplica una rotación, un zoom y una vuelta aleatoriamente; posteriormente la siguiente capa cambia de tamaño a la imagen para ajustarlo a las dimensiones utilizadas en el experimento (224 x 224 píxeles). A continuación se utiliza una capa convolucional con 16 neuronas, esto permite crear un núcleo de convolución con la salida de la capa anterior como entrada y obtener un tensor de salidas, en otras palabras, la convolución consiste en ir comparando la imagen de entrada con una matriz creada (kernel). Dentro de esta capa se utiliza la función de activación ReLU (*rectified linear activation unit*), que básicamente lo que hace esta función es que si recibe cualquier número negativo devuelve un 0 y si no, devuelve ese número positivo, esto para la comparación con el kernel. Pasada la convolución se implementa una capa de *Max Pooling 2D* para hacer una reducción de las dimensiones de alto y ancho de la salida de la capa convolucional anterior, pero no modifica su profundidad. Posteriormente, se vuelve a crear otra capa convolucional seguida de una capa de *max pooling*, pero en esa ocasión con 32 neuronas en la capa convolucional. A continuación, se implementa una capa *Dropout* para desactivar neuronas aleatorias y evitar que estas neuronas creen dependencia entre sí, ayudando a que las neuronas no dependan entre sí. Después, se agrega una capa *Flatten* para convertir los parámetros que vienen en forma de matriz a un arreglo vector, para posteriormente crear una capa *Dense*, es decir, completamente conectada con una dimensionalidad de salida de 128 y utilizando también la función de activación ReLU. Al final se agrega otra capa *Dense* con una salida del número total de clases para obtener la predicción. Esta descripción puede ser observada en la Figura 1, donde se muestran las capas contenidas dentro del modelo.

Después de esto, el modelo es compilado utilizando la función de coste "*Sparse Categorical Crossentropy*" para poder tratar el valor entre el valor estimado y el valor real, esto para optimizar el resultado. Además, se usa el optimizador

Model: "sequential\_2"

Layer (type)	Output Shape	Param #
sequential (Sequential)	(None, 224, 224, 3)	0
sequential_1 (Sequential)	(None, 224, 224, 3)	0
conv2d (Conv2D)	(None, 224, 224, 16)	448
max_pooling2d (MaxPooling2D)	(None, 112, 112, 16)	0
conv2d_1 (Conv2D)	(None, 112, 112, 32)	4640
max_pooling2d_1 (MaxPooling2D)	(None, 56, 56, 32)	0
dropout (Dropout)	(None, 56, 56, 32)	0
flatten (Flatten)	(None, 100352)	0
dense (Dense)	(None, 128)	12845184
dense_1 (Dense)	(None, 8)	1032
Total params: 12,851,304		
Trainable params: 12,851,304		
Non-trainable params: 0		

Figure 1. Modelo de la red neuronal

adam para la función del error y el *accuracy* (exactitud) como métrica de evaluación.

4) *Pruebas*: Las pruebas realizadas implican mover los hiperparámetros con los que se realiza el experimento, estos son las dimensiones de la imagen, el tamaño del lote o el número de épocas de entrenamiento. La variación de estos valores presenta un movimiento en los resultados al momento de entrenar, los valores definidos y utilizados en el código son los que dieron un mejor desempeño al momento de entrenar y evaluar el modelo.

5) *Entrenamiento del modelo*: Para el entrenamiento del modelo creado se tomaron los dos grupos de imágenes creados, uno de entrenamiento y uno de prueba, además de indicar un total de 15 épocas de entrenamiento. El objetivo visible durante el entrenamiento debe ser que el *accuracy* incrementa a medida que pasan las épocas y el error disminuya.

6) *Exportación*: El modelo después de ser entrenado y que ofrezca buen desempeño en su evaluación, es exportado con una extensión .h5, esto ayudará a después convertirlo a un modelo de TensorflowJS.

### C. Desarrollo de un API REST

Uno de los objetivos del proyecto es poder devolver al usuario información adicional y específica de cada edificio. Para llevar a cabo esta tarea, se realizó una API Rest conectada a una base de datos que permitiría el manejo y visualización de la información complementaria.

1) *Creación*: Para la creación del API capaz de almacenar información y permitir a un administrador manejar dicha información, se utiliza Node.js en su versión 12.18.4 para poder crear dicha API que tendrá interacción con una base de datos alojada en MongoDB Atlas en su versión gratuita. Una vez finalizada, se utiliza Heroku para poder desplegar la

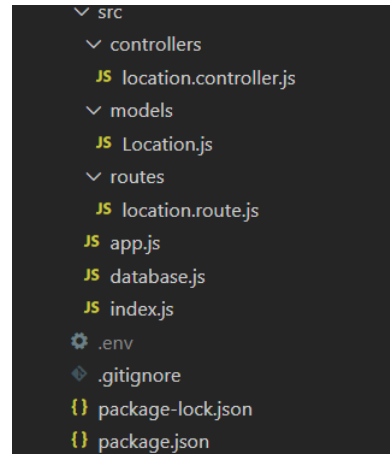


Figure 2. Estructura del API

API y poder acceder a ella desde cualquier lugar mediante un link generado.

En la figura 2 se muestra la manera en como está estructurado el código fuente desarrollado para el API, en la carpeta de *controllers* se encuentran los endpoints que serán utilizados para el manejo de la información. El archivo *Location.js* contiene el modelo con los campos requeridos para generar un nuevo registro de una ubicación. En el directorio de *routes* se encuentra el manejo de las rutas utilizando *express* para asignarle una dirección a cada endpoint. En el archivo *database.js* está la configuración a la base de datos creada previamente en la plataforma de MongoDB Atlas.

2) *Función del servicio*: Esta API será utilizada por la aplicación móvil y también por la plataforma web, siendo la app la que consumirá más la visualización de la información, mientras que la plataforma web será el medio por el cuál pueda ser insertada nueva información o modificación de la misma. Dentro de sus funciones, permite un *CRUD* (*Create*, *Read*, *Update* y *Delete*) completo para las ubicaciones. Existen 3 endpoints sobre el obtener información, uno permite obtener todas las ubicaciones disponibles, otro permite una búsqueda específica por el id del registro creado y uno último que permite obtener información específica que será utilizada para cuando la app prediga sobre algún edificio reconocido. También se cuenta la opción de crear, actualizar y eliminar un registro, estos últimos son utilizados en la plataforma web. Las respuestas regresadas son el formato JSON, esto permite un buen manejo de la información y también es sencillo de utilizar en los frameworks utilizados, ya que es común encontrarlo en las librerías disponibles más por ser de JavaScript, además de que en MongoDB se maneja en dicho formato igualmente.

### D. Administración de ubicaciones

Se requirió de la creación de una plataforma para la administración de las ubicaciones e información contenida de las mismas, para esto se desarrolló una plataforma web utilizando el framework React en su versión 17.0.1 que a continuación será descrita.



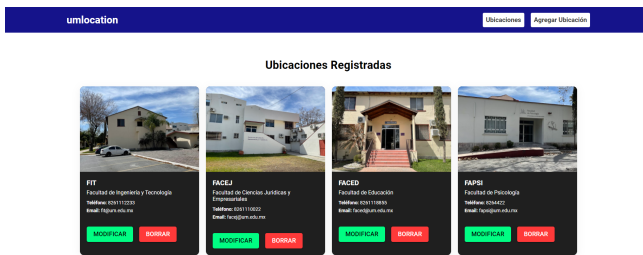


Figure 3. Interfaz principal plataforma web

1) *Funcionalidad*: Esta aplicación permite la visualización, modificación, eliminación y creación de registros de ubicaciones, mediante un formulario que proviene del modelo provisto por el API especificada en el punto anterior. Se utiliza axios para realizar las llamadas a los endpoints necesitados en el manejo de los datos. El modelo de las ubicaciones incluye un nombre, un teléfono de contacto, sitio web en caso de contar con alguno, correo electrónico, el horario de atención, una imagen de la fachada y también las coordenadas. Toda esta información es la que se maneja y permite dar un plus al usuario que desconoce la universidad y quisiera contactarse con alguna facultad o departamento en específico. En la pantalla principal mostrada en la Figura 4 se puede apreciar la manera en cómo se ve la lista de las ubicaciones ya registradas, las funcionalidades disponibles están marcadas por los botones de “modificar” y “borrar”, además de que al hacer click sobre cada tarjeta se puede ver más información sobre ese item. También se puede agregar un nuevo registro con el botón en la barra superior. Las imágenes no son guardadas en la base de datos, son subidas previamente a una carpeta compartida en *Google Drive* y así poder generar un link que es el guardado en el registro del formulario.

Cabe mencionar que en esta versión la plataforma no cuenta con un login, por lo que no tiene seguridad al momento de administrar en los registros realizados.

#### E. Implementación del modelo en la app

1) *Preparación del entorno react-native*: Se agregó paquetes de tensorflow

Estructura

2) *Importación del modelo*: Cargar el modelo

3) *Configuración de la cámara*: Creación de la vista para capturar la imagen

4) *Predicción*: Utilizar el modelo y devolver resultado

5) *Despliegue de la información*: Retornar información adicional (consulta al API) junto con la predicción.

6) *Condicionamiento*: Bloqueo de la función si no está dentro del área de la universidad

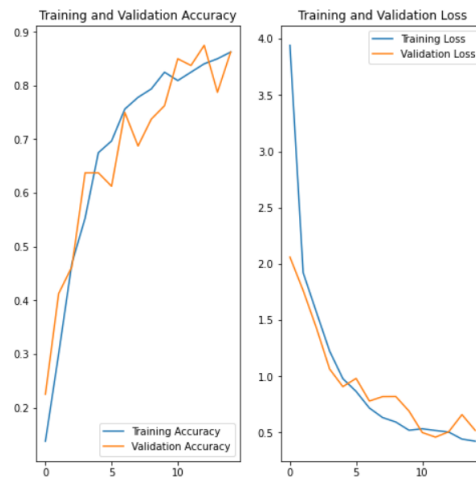


Figure 4. Interfaz principal plataforma web

## IV. RESULTADOS

Evaluación del modelo

Interfaces (antes de tomar la imagen, después de la predicción)

## V. CONCLUSIONES

Content

## REFERENCIAS

- [1] "Recorrido Virtual." [Online]. Available: <http://www.comitedeanalisis.unam.mx/recorrido-virtual.html> [Accessed: 26- Apr- 2021].
- [2] "Recorridos Virtuales." [Online]. Available: <https://www.iberopuebla.mx/la-ibero/campus/recorridos-virtuales> [Accessed: 26- Apr- 2021].
- [3] "Recorrido Virtual." [Online]. Available: <https://www.uic.mx/uic-360/virtualtour.html> [Accessed: 26- Apr- 2021].
- [4] "Visitas guiadas y excursiones en todo el mundo." [Online]. Available: <https://www.civitatis.com/es/> [Accessed: 26- Apr- 2021].
- [5] J. L. del Val Román, "Industria 4.0: la transformación digital de la industria," *Facultad de Ingeniería de la Universidad de Deusto*, 2016.
- [6] D. Evans, "Internet de las cosas," *Cisco Internet Business Solutions Group (IBSG)*, 2011.
- [7] S. I. Mariño and C. R. Primorac, "Propuesta metodológica para desarrollo de modelos de redes neuronales artificiales supervisadas," *IJERI: International Journal of Educational Research and Innovation*, 2016.
- [8] B. P. Orozco, "Inteligencia Artificial," *INCyTU*, 2018.
- [9] A. Serrano, E. Soria, and J. Martín, *Redes Neuronales Artificiales*. Universidad de Valencia, 2009.
- [10] D. J. Matich, "Redes Neuronales: Conceptos Básicos y Aplicaciones," *Universidad Tecnológica Nacional*, 2001.
- [11] G. Restrepo Arteaga, "Aplicación del aprendizaje profundo ("deep learning") al procesamiento de señales digitales," Master's thesis, Universidad Autónoma de Occidente, 2015.
- [12] J. Padiál, "Técnicas de programación "deep learning": ¿simulacro o realización artificial de la inteligencia?," *Naturaleza y Libertad. Revista de estudios interdisciplinarios*, 2019.
- [13] "Python." [Online]. Available: <https://www.python.org/> [Accessed: 26- Apr- 2021].
- [14] G. Zaccane, *Getting Started with TensorFlow*. 2016.
- [15] A. Marzal, I. Gracia, and P. García, *Introducción a la programación con Python 3*. Universitat Jaume I, 2014.
- [16] "Anaconda." [Online]. Available: <https://www.anaconda.com/> [Accessed: 02- May- 2021].
- [17] V. Silaparasetty, *Deep Learning Projects Using TensorFlow 2*. Apress, 2020.
- [18] "Anaconda individual edition." [Online]. Available: <https://www.anaconda.com/products/individual> [Accessed: 02- May- 2021].
- [19] "TensorFlow." [Online]. Available: <https://www.tensorflow.org/> [Accessed: 26- Apr- 2021].
- [20] J.-T. Park, H.-G. Kim, and I.-Y. Moon, "The JavaScript and Web Assembly Function Analysis to Improve Performance of Web Application," *International Journal of Advanced Science and Technology*, 2018.
- [21] A. Purificación, *Manual de JavaScript*. EDITORIAL CEP S.L., 2011.
- [22] "Node.js." [Online]. Available: <https://nodejs.org/es/> [Accessed: 02- May- 2021].
- [23] E. Castelo, "Diseño y desarrollo de una API REST sobre la que se apoyará una red social y una aplicación Android (SOCIAL PILGRIM)," *Universidad de Vigo*, 2016.
- [24] "Express." [Online]. Available: <http://expressjs.com/> [Accessed: 02- May- 2021].
- [25] "Mongoose." [Online]. Available: <https://mongoosejs.com/> [Accessed: 02- May- 2021].
- [26] J. Brownlee, *Deep Learning With Python*. Machine Learning Mastery, 2016.
- [27] Z. Li, "NoSQL Databases," *Geographic Information Science Technology Body of Knowledge*, 2018.