

# Proyecto: Detección de árboles en imágenes aéreas

## Documento de Requisitos del Sistema (DRS)

Autores: Pablo Asensio Martínez  
Vanessa Lomas García  
Modelo: IEEE/ANSI 830-1998

<b>1</b>	<b>Introducción .....</b>	<b>2</b>
1.1	Propósito del documento .....	2
1.2	Alcance del sistema .....	2
1.3	Definiciones, acrónimos y abreviaturas .....	2
1.4	Referencias .....	3
1.5	Descripción del resto del documento .....	3
<b>2</b>	<b>Descripción general .....</b>	<b>4</b>
2.1	Perspectiva del producto .....	4
2.2	Funciones del producto .....	4
2.3	Características del usuario .....	6
2.4	Restricciones generales .....	6
2.5	Suposiciones y dependencias .....	6
<b>3</b>	<b>Requisitos específicos.....</b>	<b>7</b>
3.1	Funciones.....	7
3.2	Rendimiento .....	7
3.3	Restricciones de diseño .....	7
3.4	Atributos del sistema .....	8
<b>4</b>	<b>Presupuesto.....</b>	<b>9</b>

# 1 Introducción

En esta sección se proporcionará una introducción a todo el documento. Consta de varias subsecciones: propósito, ámbito del sistema, definiciones, referencias y visión general del documento.

El repositorio donde se va a desarrollar el sistema es el siguiente:

[https://github.com/pasensio97/AIVA\\_2021-imagenes\\_aereas](https://github.com/pasensio97/AIVA_2021-imagenes_aereas)

## 1.1 Propósito del documento

Documento dirigido al cliente José Vélez Serrano y a la dirección de Blind Visión. El objetivo del documento es fijar los requisitos del sistema impuestos por el cliente y explicar la manera en que se va a abordar el problema.

## 1.2 Alcance del sistema

Se desea crear una aplicación que, pasándole una imagen aérea, detecte el número de árboles que hay. En adelante, se denominará a dicha aplicación como TreeDetector.

TreeDetector será capaz de realizar un conteo de los árboles presentes en una imagen aérea y de localizarlos con una bounding box. La aplicación deberá poder utilizarse en cualquier estación del año, lo que implica que se deberán reconocer los árboles en cualquiera de los estados por los que pasan a lo largo de un año (con hojas y sin hojas).

Con esta aplicación se podrán realizar estudios de la cantidad de vegetación de hay en una determinada zona, o de forma más generalizada, en una determinada ciudad. Estos estudios podrían ser muy útiles para, por ejemplo, realizar un análisis de la calidad del aire.

## 1.3 Definiciones, acrónimos y abreviaturas

- **Dataset.** Hace referencia al conjunto de datos con el que se trabajará, en nuestro caso las imágenes aéreas.
- **Bounding Box.** Rectángulo que viene determinado por cuatro coordenadas(x, y, anchura, altura) y que se dibujará en las imágenes para indicar la posición de los árboles.
- **Deep Learning (en español, aprendizaje automático).** Es un conjunto de algoritmos de aprendizaje automático que utilizan redes neuronales para dar solución a problemas de por ejemplo clasificación o detección.
- **Red neuronal.** Son un modelo inspirado en el funcionamiento del cerebro humano, formado por un conjunto de nodos llamados neuronas que están conectadas y transmiten señales entre sí. Estas neuronas se organizan en capas y su objetivo es el de ‘aprender’ a realizar una tarea concreta, en este caso, detectar árboles en imágenes.
- **Data augmentation.** Técnica que permite aumentar el tamaño y diversidad de un dataset por medio de perturbaciones en los datos originales (rotaciones, traslaciones...). Esto hará que se mejorar la precisión y la generalización del modelo.

- **API.** Del inglés, *Application Programming Interface*, permite aumentar el tamaño y diversidad de un dataset por medio de perturbaciones en los datos originales (rotaciones, traslaciones...). Esto hará que se mejore la precisión y la generalización del modelo.
- **REST.** Del inglés, *representational state transfer*, es un estilo de arquitectura *software* para sistemas hipermedia distribuidos.

## 1.4 Referencias

- Aplicación LabelImg para etiquetar las imágenes de muestra.  
<https://github.com/tzutalin/labelImg>
- Yang, L., Wu, X., Praun, E., & Ma, X. (2009, November). Tree detection from aerial imagery. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* (pp. 131-137).

## 1.5 Descripción del resto del documento

En el apartado 2 se describe el producto que se va a desarrollar, se ponen ejemplos del material que se va utilizar para su desarrollo y se describe el entorno de trabajo que se va a utilizar. En el apartado 3 se describen los requisitos que tiene que cumplir el sistema que se va a diseñar.

## 2 Descripción general

En esta sección se describen todos aquellos factores que afectan al producto y a sus requisitos. No se describen los requisitos, sino su contexto. Esto permitirá definir con detalle los requisitos en la sección 3, haciendo que sean más fáciles de entender.

### 2.1 Perspectiva del producto

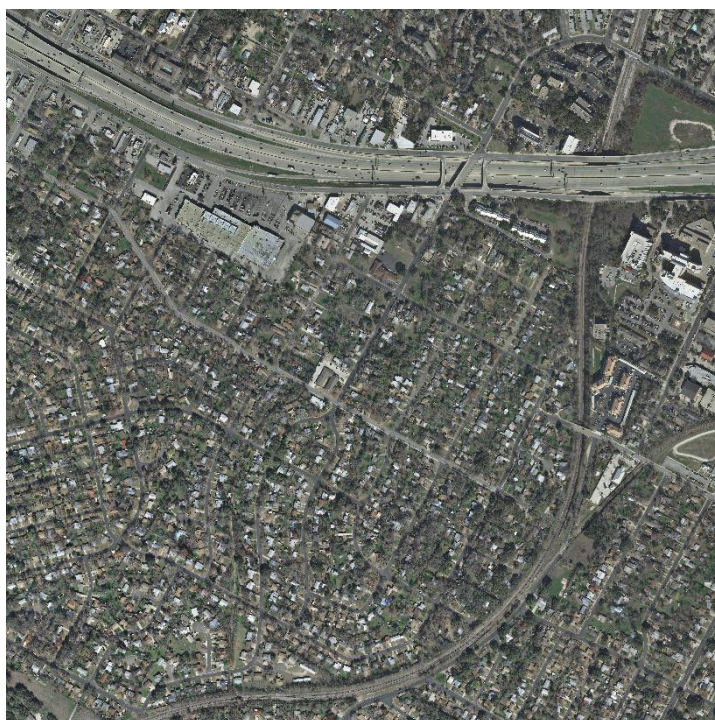
El objetivo es desarrollar un sistema que sea capaz de detectar árboles en imágenes aéreas. Dicho sistema deberá poder llamarse desde Java. El cliente dice que quiere es una API en Java que use REST.

Se abordará el problema mediante técnicas de Deep Learning. Es posible que esta técnica esté sujeta a cambios.

La aplicación inicialmente será diseñada para detectar árboles, pero en un futuro podría seguir desarrollándose para que fuese capaz de detectar arbustos o jardines. También se podría ampliar haciendo que realice la detección sobre otro tipo de imágenes, como pueden ser imágenes tomadas por un dron.

### 2.2 Funciones del producto

La aplicación recibirá una imagen aérea, como la mostrada en la Figura 1, y sobre ella se realizará la detección de los árboles.



*Figura 1. Ejemplo de imagen aérea.*

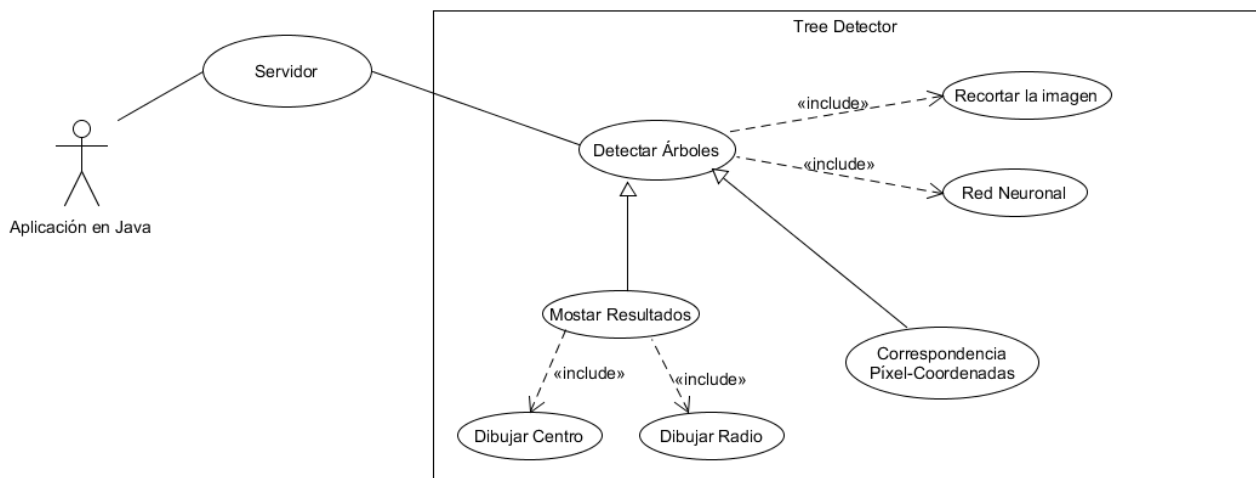
Cabe mencionar que las imágenes aéreas con las que se trabajará tiene una alta dimensionalidad por lo que será necesario dividir las imágenes en imágenes más pequeñas sobre las que se realice la detección de árboles.

La salida que se obtendrá será la imagen recibida donde se habrán señalado la posición de los árboles detectados con una circunferencia y su correspondiente centro. Un ejemplo del resultado que se obtendrá sobre una pequeña región de la imagen original se muestra en la Figura 2.



*Figura 2. Ejemplo de detección de árboles.*

A continuación se muestra un diagrama de casos de uso que muestra de manera simplificada el funcionamiento del sistema que se va a diseñar:



*Figura 3. Diagrama de casos de uso de la aplicación*

## 2.3 Características del usuario

No se ha indicado el perfil del usuario que utilizará la aplicación ni con qué propósito a parte del de contar árboles.

## 2.4 Restricciones generales

- Lenguaje de programación. Se utilizará Python como lenguaje de programación en el servidor, ya que cuenta con librerías muy potentes para trabajar en tareas de visión artificial como es OpenCV y con otras librerías como TensorFlow y Keras que permiten desarrollar modelos de Deep Learning de forma sencilla.
- Limitaciones del hardware. Para ejecutar la aplicación será necesario un servidor que disponga de un procesador de al menos 16 hilos y una tarjeta gráfica como puede ser la RTX 2060.
- Protocolos de comunicación. Llamadas se utilizará alguna biblioteca de REST para Java que interactuará con el servidor.

## 2.5 Suposiciones y dependencias

- Para el desarrollo de la aplicación se cuenta con 2 imágenes de dimensiones 5000x5000 proporcionadas por el cliente. Adicionalmente será necesario ampliar el *dataset* con imágenes que se recopilarán de la vista aérea de Google Maps. Otra forma de aumentar la cantidad de muestra es utilizar *data augmentation*.
- Las imágenes con las que se trabajará deberán tener una resolución suficiente para que los árboles puedan ser identificados por el ojo humano a la hora de hacerles zoom y etiquetarlos.
- La aplicación se desarrollará con la versión de Python 3.8, usando bibliotecas como: TensorFlow, Keras y OpenCV.

### 3 Requisitos específicos

Esta sección contiene los requisitos a un nivel de detalle suficiente como para permitir a los diseñadores diseñar un sistema que los satisfaga y que permita al equipo de pruebas planificar y realizar las pruebas que demuestren si el sistema satisface, o no, los requisitos.

#### 3.1 Funciones

##### 1. Detectar árboles en imágenes aéreas

Origen: La imagen aérea de la ciudad de Austin contiene árboles, además de casas y carreteras, y el cliente pide detectarlos y contarlos.

Para desarrollarlo y testarlo se precisa del máximo número de muestras posibles para entrenar una red neuronal. Como mínimo se deberá entrenar con 500 imágenes de resolución 250x250.

##### 2. Asignar a cada árbol su correspondientes coordenadas de altitud y latitud

Origen: Las imágenes aéreas sobre las que se quiera realizar la detección pueden solaparse y no se debe contar el mismo árbol más de una vez.

Para cumplir con este requisito se necesitará disponer de un mapa de coordenadas para cada imagen.

#### 3.2 Rendimiento

- El cliente no aceptará un sistema con un error superior al 10%.
- La velocidad de respuesta que espera el cliente del sistema tendrá que ser menor a un minuto por imagen.
- El cliente desea poder realizar la detección sobre, como mínimo, 100 imágenes al día.

#### 3.3 Restricciones de diseño

- Se necesitará estudiar la forma en la que se va a establecer la llamada de Java al programa en Python.
- Se dispone de poca muestra de prueba por lo que se tendrán que extraer más muestras de Google Maps, o aplicar *data augmentation*.
- Se necesitarán etiquetar de manera manual las imágenes para entrenar la red neuronal.
- Se utilizará Programación Orientada a Objetos ya que permitirá que los desarrolladores puedan trabajar en distintas partes del programa.
- Se tendrá que tener en cuenta que a veces reconocer árboles en imágenes aéreas no es tarea fácil ni siquiera para el ojo humano. Dos casos que se deben tener cuenta y en los que la detección puede no ser tan precisa como debería son el caso de árboles de tronco fino y sin hojas, y el caso en el que haya que etiquetar como diferentes dos árboles que están muy juntos. En esos casos la detección se complica.

### **3.4 Atributos del sistema**

- Portabilidad del desarrollo a otros entornos hardware o software. El cliente podrá acceder al servidor desde cualquier sitio.
- Mantenibilidad del sistema con el paso del tiempo. El código desarrollado cumplirá las normas de estilo de la empresa (nomenclatura de variables y funciones, tamaño de las funciones, comentarios...).



## 4 Presupuesto

### Presupuesto del proyecto: Detección de árboles en imágenes aéreas

Elaborado por: Pablo Asensio Martínez y Vanesa Lomas García

Fecha de inicio del proyecto: 23/03/2021

Duración del proyecto: 2 meses

Tarea / Actividad	Elemento	Unidades	Coste por unidad	Coste Total
Estudio previo del proyecto				
	Elaboración del documento de requisitos del sistema (DRS)	1	75,00 €	75,00 €
	Diseño conceptual del sistema	1	150,00 €	150,00 €
			<b>Subtotal 1</b>	<b>225,00 €</b>
Personal				
	Programación del sistema	2	1.500,00 €	6.000,00 €
			<b>Subtotal 2</b>	<b>6.000,00 €</b>
Hardware servidor				
	Procesador	1	250,00 €	250,00 €
	Tarjeta gráfica	1	450,00 €	450,00 €
	Otros componentes	1	300,00 €	300,00 €
			<b>Subtotal 3</b>	<b>1.000,00 €</b>
			<b>TOTAL (sin IVA)</b>	<b>7.225,00 €</b>
			<b>TOTAL (con IVA 21%)</b>	<b>8.742,25 €</b>

Tarea / Actividad	Elemento	Unidades	Coste por unidad	Coste Total
Mantenimiento				
	Disponibilidad y mantenimiento de los servidores (por mes)	1	2.500,00 €	2.500,00 €
			Subtotal 1	2.500,00 €
			TOTAL (sin IVA)	2.500,00 €
			TOTAL (con IVA 21%)	3.025,00 €