



Centro Universitario de Oriente -CUNORI-
Ingeniería en Ciencias y Sistemas
Análisis y Diseño de Sistemas I
Ing. Hendrick Calderón

01-11-2020

Análisis y Diseño

DOCUMENTO DE ARQUITECTURA DE SOFTWARE

PROYECTO VISIÓN

**Autor: Victor Joab Morales Méndez
201644764**

Quinta Iteración

1. INTRODUCCIÓN

El presente documento contiene el diseño del sistema que pretende brindar la solución planteada en el modelo de negocio del *Proyecto Vision*, partiendo de los requisitos definidos en la fase de análisis de requerimientos que se plasmaron detalladamente en el documento “Especificación de Requerimientos”. Se describen las metodologías y tecnologías con las que los módulos del sistema serán construidos, los patrones de diseño que se utilizarán y los estándares de programación.

La arquitectura del sistema está representada siguiendo las recomendaciones de RUP. Las vistas necesarias para especificar el sistema se presentan, a continuación:

- Vista de Casos de Uso: Describe el proceso más significativo de los casos de uso.
- Vista Lógica: Describe la arquitectura del sistema, presentando varios niveles de refinamiento. Indica los módulos lógicos principales, sus responsabilidades y dependencias
- Vista de Procesos: Describe la descomposición del sistema en procesos.
- Vista de Implementación: Describe los componentes de despliegue construidos y sus dependencias.
- Vista de Despliegue: Presenta aspectos físicos como topología, infraestructura informática e instalación de ejecutables.
- Vista de Datos: Esta es una vista desde la perspectiva del almacenamiento de datos persistentes en el sistema.

2. OBJETIVOS

2.1 General

Presentar las especificaciones de diseño que deberá cumplir el sistema del proyecto vision de clasificación y reconocimiento de imágenes de billetes. Se basa en un panorama de arquitectura completo del sistema.

2.2 Específicos

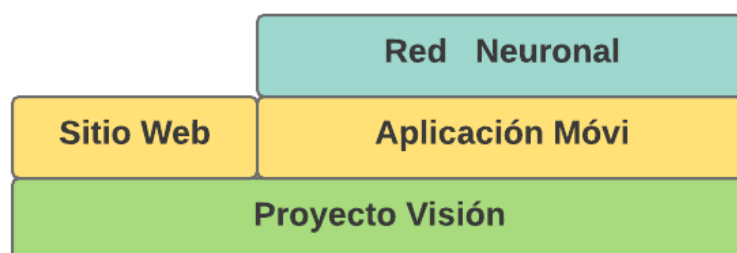
- Definir la estructura del sistema, sus módulos con sus funcionalidades y componentes adicionales.
- Definir los estándares de diseño que se seguirán en el desarrollo de la aplicación.
- Describir las tecnologías con las que será desarrollado el sistema.
- Describir la arquitectura del sistema de manera lógica.

3. METAS Y LIMITANTES ARQUITECTÓNICAS

Se describen las metas y requerimientos que debe cumplir el software, También captura las limitantes especiales que pueden aplicar: estrategias de diseño e implementación, herramientas de desarrollo, estructura del equipo, etcétera.

3.1 Plataforma Tecnológica

El sistema total del proyecto consta de una plataforma con una estructura similar a la de la figura.



3.1.1 Red Neuronal Convolucional

La red neuronal analizará en distintas capas las imágenes obtenidas con el fin de poder reconocer billetes y dar un valor de salida que corresponda a un valor monetario existente en los billetes oficiales.

Será desarrollada utilizando el Framework TensorFlow Lite con el lenguaje de programación Python 3. El entorno de desarrollo será Google Colab, puesto que es una herramienta poderosa donde Google pone a disposición hardware para la creación y entrenamiento de redes neuronales y otros tipos de tecnologías relacionados a ciencia de datos e inteligencia artificial.

3.1.2 Aplicación Móvil

Es la capa entre el usuario y el sistema de reconocimiento de los billetes (Red Neuronal). La aplicación móvil llevará acabo la clasificación de imágenes que se toman desde la cámara trasera del dispositivo. Para la clasificación de imágenes se usará la red neuronal y es aquí donde entra en juego Tensor Flow Lite, que permite crear redes neuronales ligeras y acoplarlas a aplicaciones móviles.

La aplicación móvil será desarrollada en el lenguaje de programación Java usando la Api 26 de Android. El entorno de desarrollo usado será el IDE Android Studio y el SDK de android junto con Gradle como gestor de dependencias. Es necesario usar la biblioteca de tareas de Tensorflow Lite.

3.1.3 Sitio Web

El sitio web será implementado con el stack básico de desarrollo web, html5, css3 y javascript en su estándar ECMAScrip6.

El sitio web da a conocer mejor el proyecto a las personas. Contiene distintas vistas en las cuales se muestran generalidades del proyecto y la aplicación, además tendrá enlaces para calificar la aplicación en Google Play y realizar donaciones al proyecto en Patreon.

3.2 Tecnologías Usadas

El proyecto se construirá con distintas tecnologías. Se describen solamente aquellas las tecnologías principales.

3.2.1 TensorFlow Lite

Las Redes Neuronales pueden ser enormes y computacionalmente muy costosas, por lo que muchas de estas arquitecturas es imposible que puedan ejecutarse en dispositivos móviles. Para solventar este problema ha nacido la librería TensorFlow Lite que se utiliza para optimizar el modelo de varios modos para que puedan ejecutarse y formar parte de cualquier app en un dispositivo móvil.



3.2.2 Java

Java es uno de los lenguajes de programación para el entorno de desarrollo de Android. Es robusto, orientado a objetos y multiplataforma. Permite desarrollar aplicaciones móviles para dispositivos Android.



3.2.3 EcmaScript6

Se refiere al estándar ECMAScript que ya va por la versión 6 y determina cómo emplear el lenguaje Javascript, que permite desarrollar aplicaciones web, de forma eficiente con código que se ejecuta en el navegador.



3.3 Herramientas Utilizadas Para El Desarrollo

3.3.1 Google Colab

Colab es un servicio cloud, basado en los Notebooks de Jupyter, que permite el uso gratuito de las GPUs y TPUs de Google, con librerías como: Scikit-learn, PyTorch, TensorFlow, Keras y OpenCV. Todo ello bajo Python 2.7 y 3.6. Es una herramienta ideal, no solo para practicar y mejorar conocimientos en técnicas y herramientas de Data Science, sino también para el desarrollo de aplicaciones (pilotos) de machine learning y deep learning, sin tener que invertir en recursos hardware o de Cloud. Con esta herramienta se entrenará la red neuronal.



3.3.2 Android Studio

Android Studio es el entorno de desarrollo integrado oficial para la plataforma Android. Proporciona las herramientas más rápidas para crear aplicaciones en todo tipo de dispositivo con el sistema operativo Android. Aquí se desarrollará la aplicación móvil.



3.3.3 Visual Studio Code

Visual Studio Code es un editor de código fuente desarrollado por Microsoft para Windows, Linux y macOS. Diseñado originalmente para desarrollo web pero actualmente expandido a otras áreas de la tecnología.



3.4 Estándares de Desarrollo

En esta sección se describen los estándares de desarrollo utilizados para la construcción de la aplicación móvil, el sitio web y el desarrollo de la red neuronal.

3.4.1 MVC

Modelo Vista Controlador (MVC) es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. Se trata de un modelo muy maduro y

que ha demostrado su validez a lo largo de los años en todo tipo de aplicaciones, y sobre multitud de lenguajes y plataformas de desarrollo. Con este modelo de trabajo se va a realizar la aplicación móvil y el sitio web.

3.4.2 Programación Orientada a Objetos

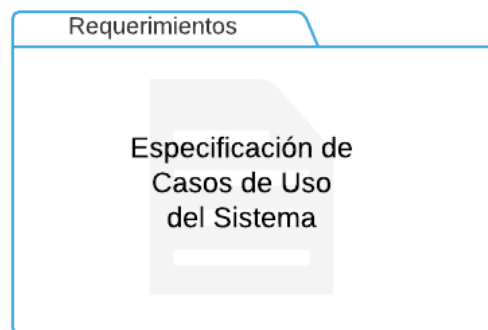
La Programación Orientada a Objetos es un paradigma de programación que viene a innovar la forma de obtener resultados. Los objetos se utilizan como metáfora para emular las entidades reales del negocio a modelar. Este paradigma es vital puesto que todo el análisis está enfocado a objetos, por lo cuál es más fácil llevar el análisis a codificación.

4. VISTA DE CASO DE USO

Esta sección lista los casos de uso o escenarios del modelo de caso de uso y el que representan elementos significativos de la funcionalidad central del sistema final.

4.1 Casos de Uso del Sistema

Los casos de uso del sistema fueron especificados detalladamente en el documento “Especificación de Casos de Uso del Sistema” en la sección de requerimientos.



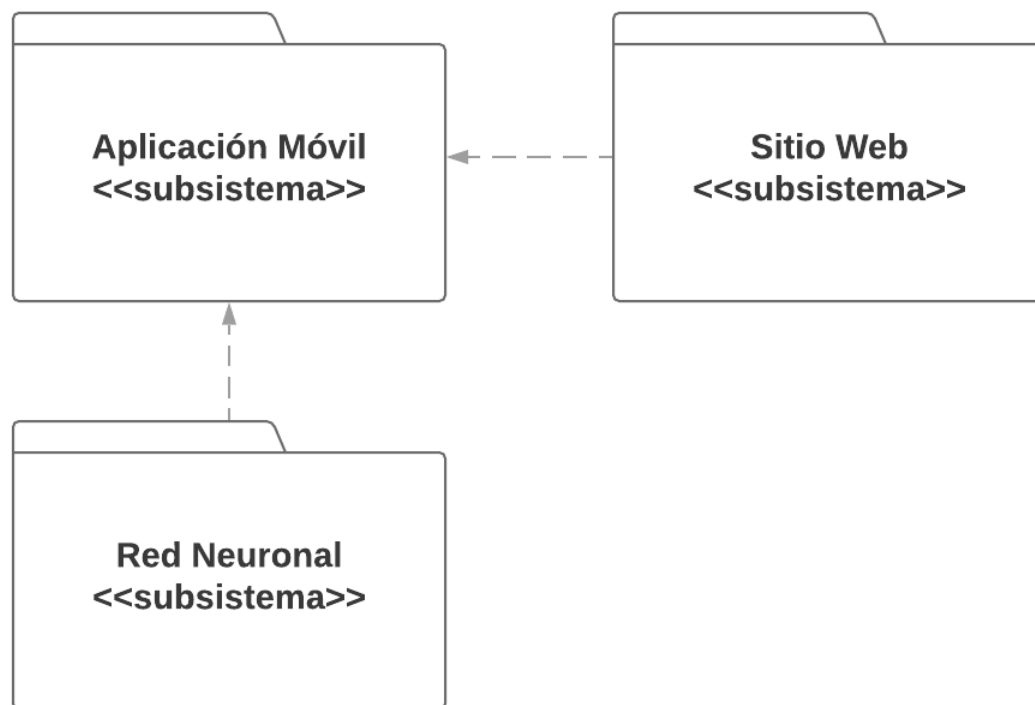
5. VISTA LÓGICA

Se presentan en este punto los sucesivos refinamientos que definen las diferentes unidades lógicas que componen la arquitectura del sistema del Proyecto Vision.

El primer refinamiento realizado consiste en la descomposición en subsistemas. Después se explora la composición de cada uno de los subsistemas. Finalmente se incluye la realización de los casos de uso descriptos en la sección anterior mediante los componentes arquitectónicos definidos.

5.1 Subsistemas

Cada subsistema consiste en el agrupamiento de diferentes funcionalidades relacionadas entre sí y posee la capacidad de funcionar como un sistema en sí mismo. La descomposición organiza la arquitectura en un conjunto de subsistemas funcionalmente cohesivos que interactúan entre sí para cumplir sus funciones.



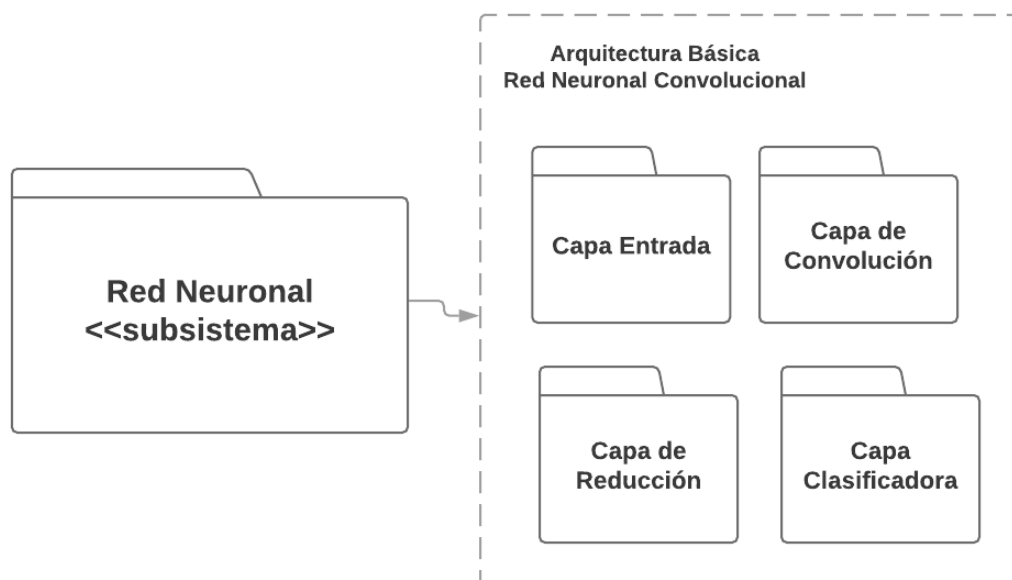
5.2 Descripción de los Subsistemas

- **Red Neuronal Convolucional:** Capaz de clasificar imágenes según el contenido de estas. Será entrenada para clasificar imágenes de billetes, clasificandolas según el valor al que corresponda el billete de la imagen.
- **Aplicación Móvil:** Es la interfaz que permite al usuario enviar imágenes a la red neuronal utilizando la cámara. El usuario toma foto del billete que desea analizar y la aplicación la envía a la red neuronal para que intente relacionar la imagen del billete con alguna clasificación existente.
- **Sitio Web:** Plataforma sobre la cuál se da a conocer el proyecto, sus objetivos y valores. Muestra como tener la aplicación y como usarla.

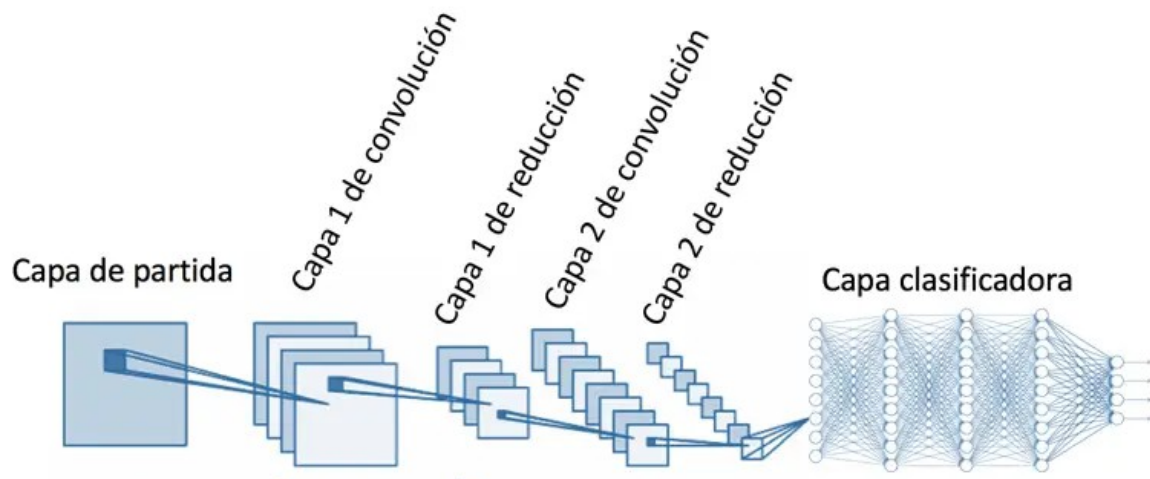
5.3 Diseño de Subsistemas

5.3.1 Red Neuronal Convolucional

Es un tipo de Red Neuronal Artificial con aprendizaje supervisado que procesa sus capas imitando al cortex visual del ojo humano para identificar distintas características en las entradas que en definitiva hacen que pueda identificar objetos y “ver”. Para ello, la red contiene varias capas ocultas especializadas y con una jerarquía: esto quiere decir que las primeras capas pueden detectar líneas, curvas y se van especializando hasta llegar a capas más profundas que reconocen formas complejas como un rostro o la silueta de un animal.



Aunque la arquitectura de la red neuronal puede disponer de la cantidad que se desee de capas de convolución y reducción en la practica puede ser 2 capas o hasta 3 capas de cada una de estas.

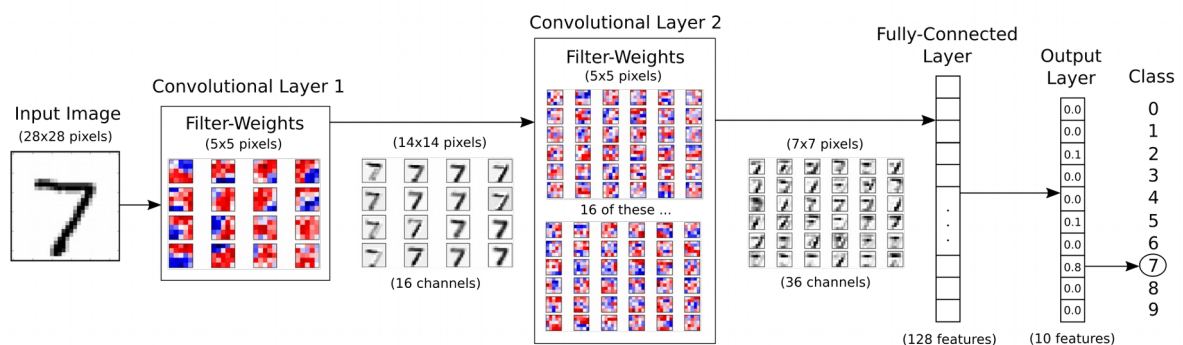


- **Capa de Entrada:** Para comenzar es necesario pre-procesar la entrada. La red toma como entrada los pixeles de una imagen. Si se tiene una imagen con apenas 28×28 pixeles, eso equivale a 784 neuronas. Esa es la capa de entrada.
- **Capa de Convolución:** Consisten en tomar “grupos de pixeles cercanos” de la imagen de entrada e ir operando producto escalar contra pequeñas matrices que se llaman filtros o kernel. Esos filtros supongamos de tamaño 3×3 pixels “recorren” todas las neuronas de entrada y genera una nueva matriz de salida, que en definitiva será una nueva capa de neuronas .
- **Capa de Reducción:** Se reduce la cantidad de neuronas antes de hacer una nueva convolución. Por cada convolución se genera otra capa de neuronas, si hiciéramos una nueva convolución a partir de esta capa, el número de neuronas de la próxima capa se iría por las nubes. Con un proceso de subsampling se reduce el tamaño de las imágenes filtradas pero en donde deberán prevalecer las características más importantes que detectó cada filtro.

- **Capa Clasificadora:** Se aplica una función llamada Softmax que conecta contra la capa de salida final que tendrá la cantidad de neuronas correspondientes con las clases que estamos clasificando. Si clasificamos perros y gatos, serán 2 neuronas. En el caso de los billetes serán 7 clases.

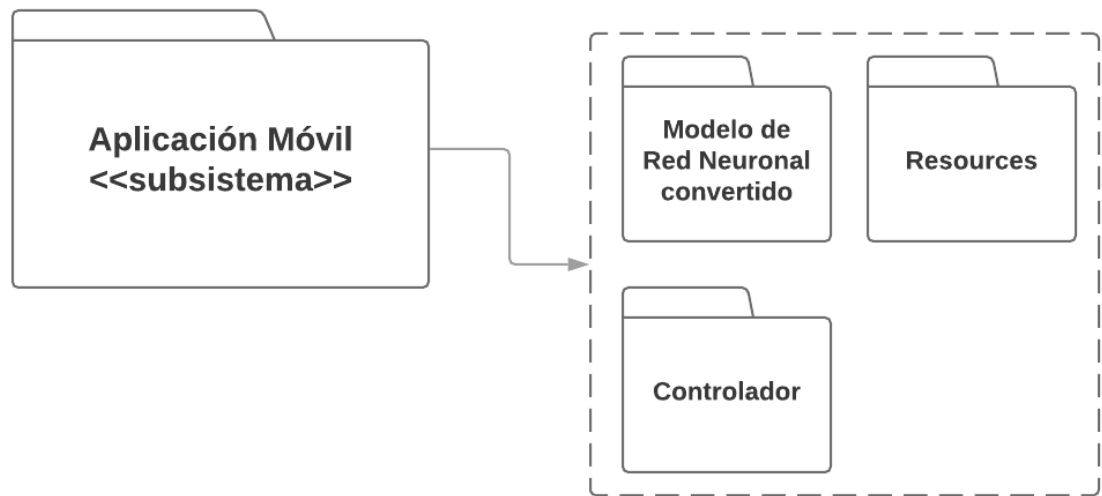
Las salidas al momento del entrenamiento tendrán el formato conocido como “one-hot-encoding” en el que para perros y gatos sería: [1,0] y [0,1], para coches, aviones ó barcos sería [1,0,0]; [0,1,0]; [0,0,1]. La función de Softmax se encarga de pasar a probabilidad (entre 0 y 1) a las neuronas de salida. Por ejemplo una salida [0,2 0,8] nos indica 20% probabilidades de que sea perro y 80% de que sea gato.

Se muestra un ejemplo del proceso de una red neuronal convolucional, donde la imagen es un numero 7, se preprocesa, luego se va dividiendo en más neuronas, por ultimo se aplanan y se manda a la capa clasificadora de salida para obtener la clasificación a la que pertenece.

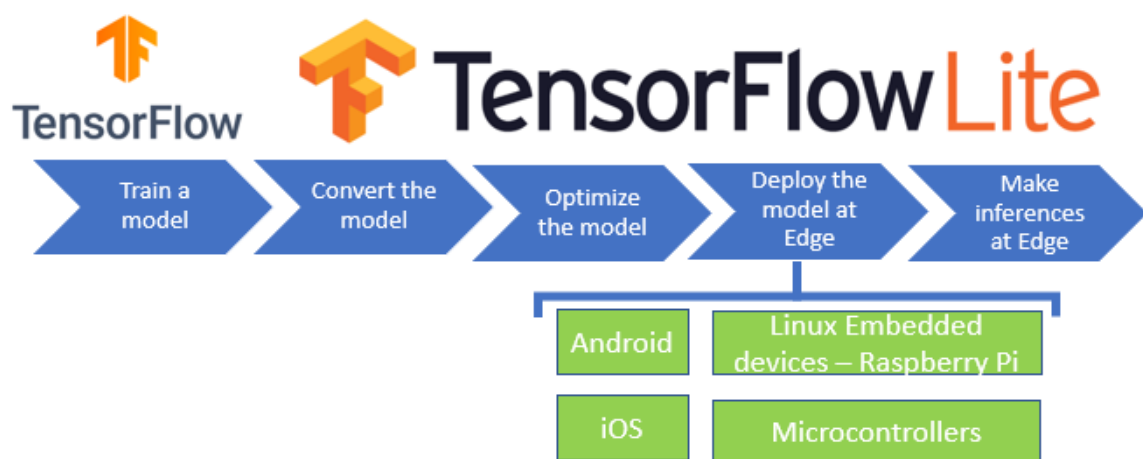


5.3.2 Aplicación Móvil

Para poder acoplar la red neuronal convolucional a la aplicación móvil es necesario realizar una conversión de modelo de red.



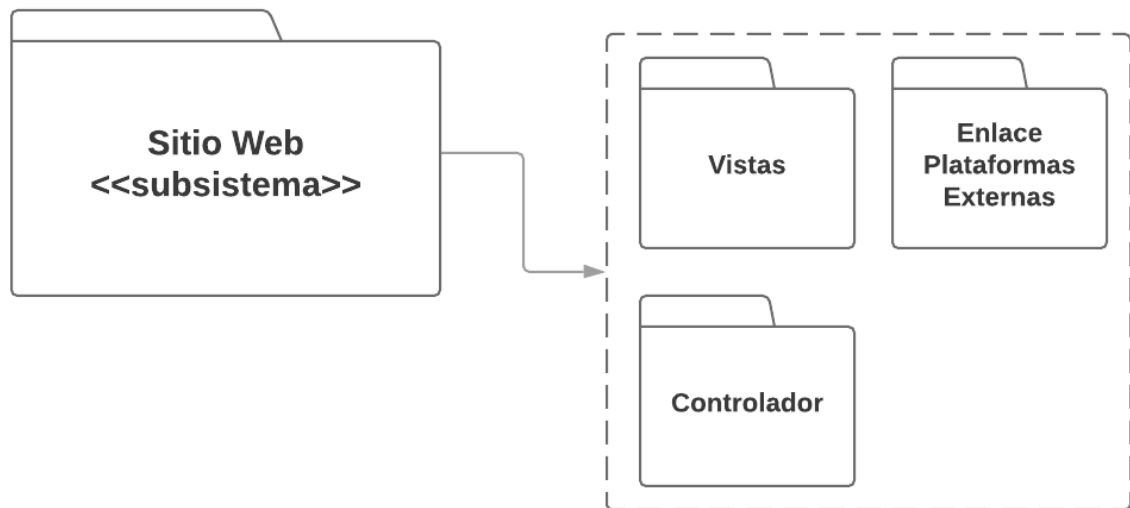
- **Modelo Red Nueronal Convertido:** Tensorflow Lite permite convertir un modelo de red ya entrenado, optimizarlo y desplegarlo para un dispositivo específico. El proceso de conversión se realiza hasta tener un modelo de red neuronal funcional. El proceso de conversión para aplicaciones móviles sería similar al mostrado en la figura.



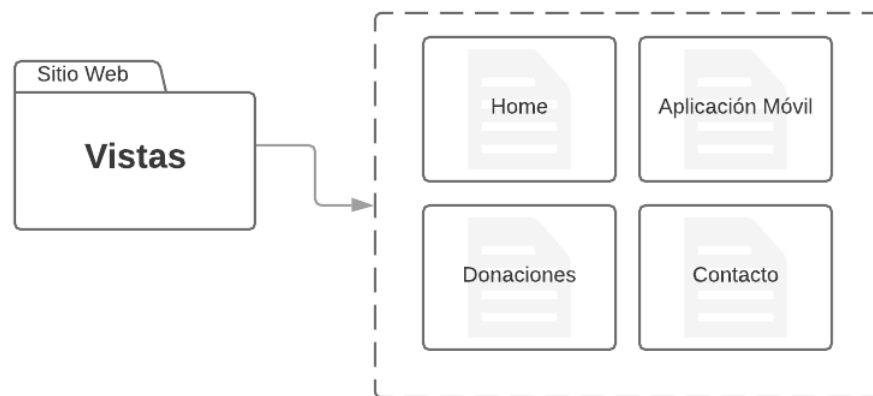
- **Resources:** Vistas e iconos que usa la aplicación.
- **Controlador:** Código Java que construye y gestiona la aplicación móvil.

5.3.3 Sitio Web

El sitio web contiene una estructura similar a la de la imagen.



- **Vistas:** Son las diferentes secciones de información del sitio web. Se han definido cuatro vistas para el sitio web: Home, Aplicación Móvil, Donaciones, Contacto.



- **Enlace a Plataformas Externas:** Los enlaces que dirigen a Patreon y Google Play.
- **Controlador:** Código Javascript del sitio web.

5.4 Realización de Casos de Uso

Para detallar el proceso e interacción secuencial de caso de uso se usaron diagramas de actividades y de secuencia por cada caso de uso o conjuntos de caso de uso cruciales, los cuales fueron plasmados en el documento “Realización de Casos de Uso del Sistema” en la sección de Análisis y Diseño.



6. VISTA DE PROCESOS

Esta sección describe la descomposición del sistema en procesos. Los diagramas de flujos de datos proveen de una vista basada en datos y procesos. Los diagramas de flujos de datos se encuentran en el archivo “*Diagramas de Flujo de Datos*” en la sección de Análisis y Diseño.

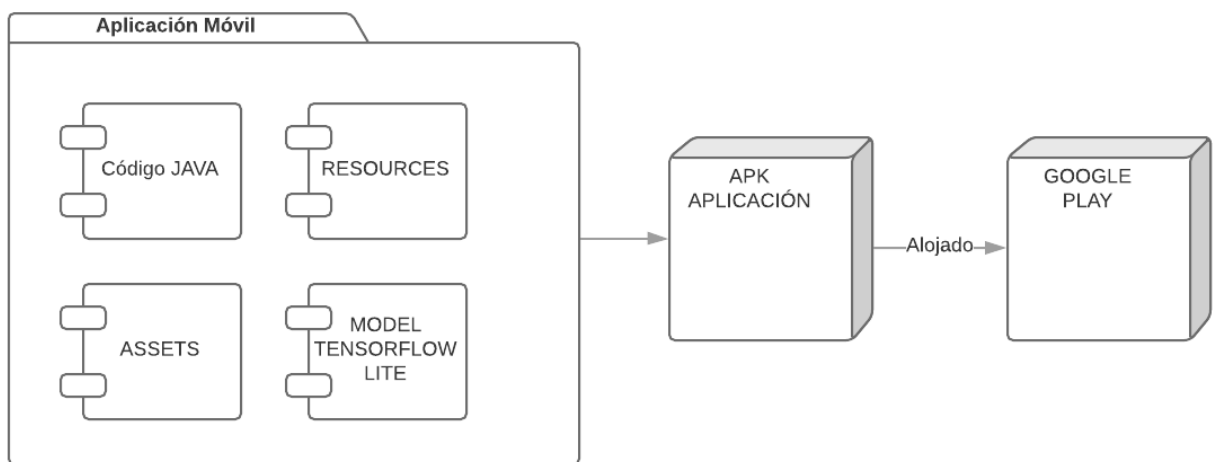


7. VISTA DE DESPLIEGUE

La Vista Implementación, describe los componentes finales de despliegue contruidos y sus interacciones con los nodos principales.

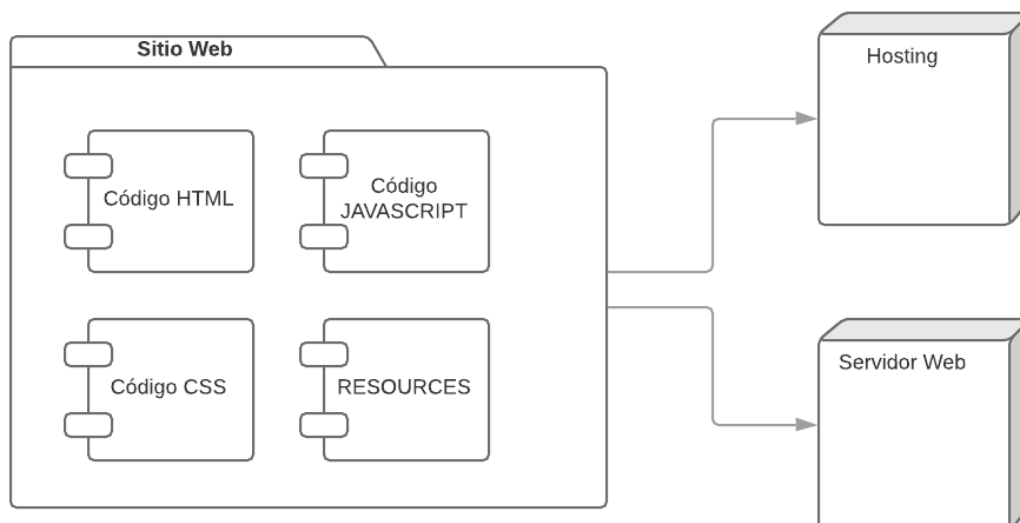
7.1 Aplicación Móvil

Al final se obtienen cuatro componentes del desarrollo de la aplicación móvil. Estos son primero compactados en un apk para luego ser alojados en Google Play para su descarga.



7.2 Sitio Web

Parecido como la aplicación móvil, el sitio web tiene cuatro componentes finales del desarrollo.



8. VISTA DE DATOS

Esta sección es un subconjunto del modelo de implementación que describe la representación lógica y física de los datos persistentes en el sistema. Estos se describe en el documento “*Modelo de Datos*” en la sección de Análisis y Diseño.

