UNIVERSIDAD MAYOR REAL Y PONTIFICIA DE SAN FRANCISCO XAVIER DE CHUQUISACA

FACULTAD DE CIENCIAS Y TECNOLOGÍA

CARRERA DE INGENIERÍA EN CIENCIAS DE LA COMPUTACIÓN



**RECONOCIMIENTO DE FIGURAS MUSICALES**

**UNIVERSITARIO**: Sacaca Colque Alexander

**CARRERA**: Ingeniería en Ciencias de la Computación

**MATERIA**: Desarrollo de aplicaciones inteligentes (SIS330)

## **Sucre, Bolivia**

Índice

[Sucre, Bolivia 1](#_Toc164260210)

[1. RESUMEN 3](#_Toc164260211)

[2. ANTECEDENTES 3](#_Toc164260212)

[3. PROBLEMÁTICA 3](#_Toc164260213)

[4. ABORDAJE O PROPUESTA DE SOLUCION 3](#_Toc164260214)

[5. OBJETIVO GENERAL 4](#_Toc164260215)

[6. OBJETIVOS ESPECIFICOS 4](#_Toc164260216)

[7. FUNDAMENTOS TEORICOS CONSIDERADOS EN EL TRABAJO 4](#_Toc164260217)

[8. METODOLOGIA 5](#_Toc164260218)

[9. RESULTADOS 12](#_Toc164260219)

[10. CONCLUSIONES 16](#_Toc164260220)

[11. RECOMENDACIONES 16](#_Toc164260221)

[12. REFERENCIAS BIBLIOGRAFICAS (Harvard o APA) 16](#_Toc164260222)

[13. ANEXOS 16](#_Toc164260223)

# **RESUMEN**

Este proyecto se centra en facilitar la vida de profesores de música y artistas que utilizan partituras en papel, digitalizando el contenido de sus cuadernos a archivos XML. La digitalización permitirá una fácil manipulación de las partituras en la aplicación MuseScore, eliminando la necesidad de ingresar manualmente las notas y mejorando el tiempo de editarlos. Se explorará la opción de transformar imágenes de partituras en archivos XML para lograr una transición rápida, con el objetivo de permitir la manipulación y edición eficiente de las partituras. La implementación se basa en técnicas de inteligencia artificial y procesamiento de imágenes para lograr una digitalización precisa de la información musical.

# **ANTECEDENTES**

El uso de partituras impresas en papel o escritas a mano sigue siendo común en el ámbito musical, pero presenta desafíos en términos de deterioro y limitaciones para la manipulación digital.   
Se vio en profesores a la hora de editar una notación musical tratan de hacerlo primero a mano para posteriormente fotocopiar y pasar a sus alumnos, pero se vio la dificultad de que es moroso a la hora de realizar dicha notación y que a veces algunas notas son incorrectas, también se vio en el ámbito de las bandas musicales, donde la necesidad de compartir partituras a menudo se satisface mediante la captura de fotografías. Sin embargo, esta práctica conlleva el inconveniente de tener que realizar una posterior transcripción manual debido a las limitaciones de calidad y formato de las imágenes capturadas. A medida que estas partituras fotocopiadas circulan, también enfrentan el riesgo de deterioro con el tiempo, afectando la calidad y durabilidad de los documentos musicales.

# **PROBLEMÁTICA**

El problema central que surge de la persistente utilización de partituras impresas en papel o creadas manualmente en el ámbito musical es la dificultad inherente en términos de manipulación digital y el riesgo de deterioro con el tiempo.

# **ABORDAJE O PROPUESTA DE SOLUCION**

Para afrontar este desafío, se empleará un avanzado modelo de visión por computadora basado en YOLOv8, reconocido por su óptimo equilibrio entre precisión y velocidad. El modelo experimentará varios procesos, utilizando una extensa base de datos de partituras musicales, perfeccionando así su capacidad para identificar y clasificar notas de manera eficiente, permitiendo a los usuarios capturar imágenes utilizando una cámara, facilitando la rápida identificación y clasificación de las notas musicales en el proceso de digitalización de partituras.  
Este modelo se encargara de identificar las figuras musicales del pentagrama y almacenar dicha imagen cuando los usuarios capturen la imagen utilizando una aplicación móvil complementaria

# **OBJETIVO GENERAL**

El problema central que surge de la persistente utilización de partituras impresas en papel o creadas manualmente en el ámbito musical es la dificultad inherente en términos de manipulación digital y el riesgo de deterioro con el tiempo.

# **OBJETIVOS ESPECIFICOS**

* Estudio e interpretación de partituras musicales.
* Estudiar el estado del arte de la identificación de figuras musicales, considerando la precisión, la eficiencia y la innovación de los diferentes enfoques de inteligencia artificial existentes del proyecto.
* Realizar un análisis y verificación exhaustivas de proyectos similares que hayan sido implementados y aplicar mejores prácticas en la construcción del proyecto.
* Desarrollar un modelo de detección de las notas musicales utilizando YOLOv8
* Estudiar el proceso de conversión precisa de la información musical a archivos XML que es compatible MuseScore

# **FUNDAMENTOS TEORICOS CONSIDERADOS EN EL TRABAJO**

a. Ámbito de la inteligencia artificial, técnicas, algoritmos, modelos base, entre otros.

Modelo de Visión por Computadora: YOLOv8 (You Only Look Once versión 8) es un modelo de visión por computadora y una arquitectura de red neuronal profunda que puede realizar la segmentación de instancias, que consiste en separar y etiquetar cada planta individual en una imagen, así como la identificación y la clasificación de objetos, que consiste en reconocer y categorizar las notas según sus características.

Aprendizaje Automático (Machine Learning): El proyecto se basa en técnicas de transfer learning y fine-tuning, aplicadas a modelos preentrenados de Visión por Computadora. Esto permite aprovechar el conocimiento de modelos ya entrenados en grandes conjuntos de datos para mejorar la precisión.

**ROC de Música**o **Reconocimiento Óptico de Música** (OMR) es la tecnología que permite el [reconocimiento óptico de caracteres](https://es.wikipedia.org/wiki/Reconocimiento_%C3%B3ptico_de_caracteres) (OCR por Optic Character Reconization) para interpretar [partituras](https://es.wikipedia.org/wiki/Partituras) o partituras impresas en formato editable o reproducible. Una vez procesada, la música se puede guardar en formatos de uso común, por ejemplo: [MIDI](https://es.wikipedia.org/wiki/MIDI) (para reproducción) y [MusicXML](https://es.wikipedia.org/wiki/MusicXML" \o "MusicXML)

b. Ámbito al que se aplicó la inteligencia artificial

Se ha explorado el impacto de la inteligencia artificial en proyectos emblemáticos, como el desarrollado por la Universidad Católica de Colombia en 2019. Este proyecto se centró en el reconocimiento automático de elementos musicales en partituras mediante técnicas de aprendizaje profundo. A diferencia de este, nuestro enfoque abarca la segmentación de instancias en imágenes, así como la identificación y digitalización, destacando la versatilidad de la inteligencia artificial en diversos contextos.

# **METODOLOGIA**

1. Sustento metodológico

La metodología se ha diseñado meticulosamente para alcanzar los objetivos específicos del proyecto y garantizar la precisión en la identificación de las figuras musicales. Está compuesta por las siguientes etapas fundamentales:

* 1. **Recopilación de Datos Fotográficos:**

Se llevará a cabo una recopilación exhaustiva de datos visuales, en forma de fotografías de partituras musicales. Estas imágenes formarán el conjunto de datos principal que será empleado en el entrenamiento del modelo YOLOv8. Se prestará especial atención a la calidad de las imágenes y a la diversidad de partituras representadas.

* 1. **Preprocesamiento de Datos Visuales:**

Las imágenes recolectadas pasarán por un proceso de preprocesamiento que incluye la estandarización del tamaño, la eliminación de imágenes borrosas y la aplicación de técnicas de mejora de la calidad. Esto tiene como objetivo mejorar la calidad y consistencia de las imágenes, lo que permitirá al modelo YOLOv8 generalizar y reconocer variaciones en las partituras musicales.

* 1. **Entrenamiento del Modelo YOLOv8:**

El modelo YOLOv8 será empleado para llevar a cabo el entrenamiento de una red neuronal profunda. Durante esta fase, el modelo aprenderá de manera autónoma a identificar y clasificar las notas musicales presentes en las imágenes de las partituras. El entrenamiento se repetirá hasta alcanzar un nivel óptimo de precisión.

* 1. **Validación y Ajustes Finales:**

Después de finalizar el entrenamiento, se llevará a cabo una validación del modelo utilizando un conjunto de datos independiente. Se realizarán ajustes necesarios para mejorar la precisión y reducir la posibilidad de errores en la identificación de las notas musicales.

* 1. **Desarrollo de la Aplicación Móvil:**

Una vez entrenado el modelo, se procederá al desarrollo de la aplicación móvil. Esta aplicación permitirá a los usuarios capturar imágenes de partituras musicales y enviarlas al modelo entrenado para su identificación. La interfaz de la aplicación se diseñará priorizando la intuición y la facilidad de uso, facilitando así la digitalización de las partituras musicales.

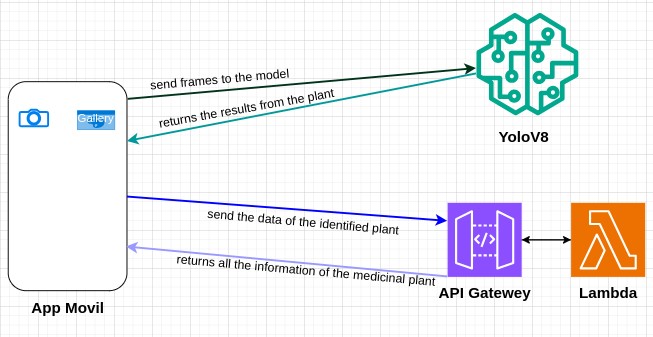
1. Técnicas de recolección de datos

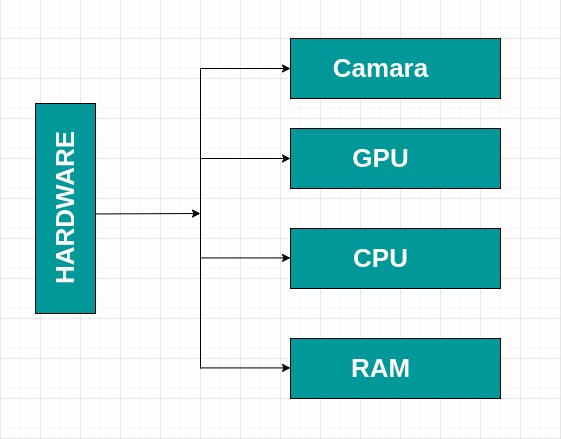
En la ejecución de este proyecto, se emplearán diversas técnicas de recolección de datos con el propósito de obtener información valiosa sobre las partituras musicales. Estas técnicas se han centrado en la obtención de datos visuales y descriptivos relacionados con las partituras musicales. Entre las técnicas utilizadas se incluyen:

* Fotografía de Campo: Se llevará a cabo la captura de imágenes de partituras musicales en su entorno natural. Esto involucra la toma de fotografías de partituras específicas, incluyendo sus notas, claves, y otros elementos identificativos.
* Investigación Documental: Se consultará una variedad de fuentes bibliográficas y recursos en línea para recopilar información detallada sobre las partituras musicales, incluyendo su composición, estructura y características distintivas.
* Entrevistas y Encuestas: Se llevarán a cabo entrevistas y encuestas a profesores de música, artistas y otros expertos en el campo para recopilar información anecdótica y experiencial sobre la manipulación y digitalización de partituras musicales. Esto ha permitido enriquecer la base de datos con conocimiento práctico y perspectivas diversas.
* Bases de Datos Existentes: Se recurrirá a bases de datos existentes sobre partituras musicales, tanto en formato digital como impreso, para enriquecer el conjunto de datos y ampliar el alcance del proyecto.

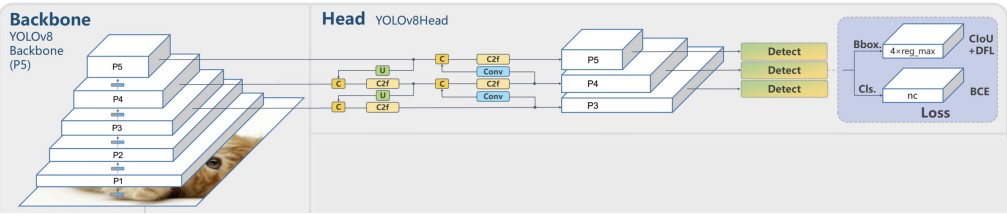
Estas técnicas de recolección de datos se combinarán para obtener un conjunto de información completo sobre las partituras musicales y facilitar su digitalización y manipulación en el contexto del proyecto.

1. Materiales & herramientas
   1. Arquitectura software desarrollado.



* El componente **aplicación móvil** es el software desarrollado con **Flutter**, un framework de código abierto para crear aplicaciones nativas para iOS y Android. Esta aplicación recibe la imagen desde la camara o galería, la procesa y la envía al modelo YoloV8. Además, muestra los resultados y la información de las notas musicales del usuario.
* El componente del modelo **YoloV8** está integrado en la aplicación móvil y es una red neuronal convolucional profunda. Recibe las imágenes procesadas desde la aplicación y devuelve los resultados de detección y clasificación de las notas musicales.
* La API, construida con servicios de **AWS, Lambda y API Gateway**, almacena información sobre las notas musicales, como sus nombres y posiciones. La API recibe las imágenes desde la aplicación Flutter y proporciona información de las notas musicales.
  1. Esquema y descripción de componentes del software.
  + **Cámara (Cam):** La cámara es el componente de entrada que captura imágenes en tiempo real o imágenes estáticas desde la galería. La aplicación utiliza la cámara para adquirir los datos visuales de las notas musicales que se va a analizar.
  + **CPU (Unidad Central de Procesamiento):** La CPU, el cerebro de nuestro dispositivo, es responsable de procesar los datos capturados por la cámara. Realiza tareas como la conversión de imágenes en datos digitales, el control de flujo de la aplicación y el procesamiento inicial de las imágenes.
  + **GPU (Unidad de Procesamiento Gráfico):** La GPU es un componente esencial para acelerar la detección de objetos. Aprovecha su capacidad de procesamiento paralelo para ejecutar operaciones intensivas de análisis de imágenes, como la detección de figuras musicales. Esto permite una respuesta rápida y suave al usuario al mostrar resultados.
  + **RAM (Memoria de Acceso Aleatorio):** La RAM proporciona un espacio de memoria temporal para almacenar datos críticos durante el procesamiento. Los datos de imágenes procesados, los resultados de la detección y otra información relevante se almacenan en la RAM para su acceso rápido. Esto es esencial para la eficiencia de la aplicación.
  1. Esquema y descripción de modelos o componentes inteligentes (esquemas y/o gráficas).

**Arquitectura de YoloV8**



**Backbone (Espina dorsal):**

El backbone se refiere a la parte de la red neuronal que extrae características de la entrada (como una imagen) y crea una representación de características.

En el contexto de YOLOv8, el backbone es responsable de procesar la entrada y aprender características a diferentes niveles de abstracción.

Los modelos preentrenados, como ResNet, Xception o MobileNet, a menudo se utilizan como backbones en YOLOv8 para aprovechar su capacidad de extracción de características.

**Head (Cabeza):**

La cabeza es la parte de la red que sigue al backbone y se encarga de realizar tareas específicas, como detección de objetos o segmentación.

En YOLOv8, la cabeza incluye capas adicionales que procesan las características extraídas por el backbone.

La cabeza puede variar según la tarea: por ejemplo, en detección de objetos, la cabeza genera cajas delimitadoras y puntajes de confianza para cada objeto detectado.

* 1. Valores de parámetros e hiperparámetros aplicados
  + epochs = 50: Define el número de épocas de entrenamiento, en este caso, se establece en 50.
  + batch = 8: Especifica el tamaño del lote (batch size) utilizado durante el entrenamiento. En este caso, se utiliza un tamaño de lote de 8.
  + imgsz = 640: Indica el tamaño de las imágenes de entrada durante el entrenamiento (640x640 píxeles en este caso)..
  + pretrained = True: Indica que se utilizará una versión preentrenada del modelo como punto de partida.
  1. Técnicas de depuración aplicadas

Durante el desarrollo, he aplicado varias técnicas de depuración para asegurarme de que mi código funcione de manera óptima. Estas incluyen:

* Análisis de trazas de pila para identificar y corregir errores.
* Impresiones estratégicas para verificar valores de variables y flujos de ejecución.
* Visualización de gráficos y resultados intermedios para comprender el procesamiento de datos.
* Control de tiempos de ejecución en visual code para identificar posibles cuellos de botella.
* Consulta de la documentación de visual code y bibliotecas asociadas para solucionar problemas específicos.
  1. Especificaciones técnicas.

Se necesita tener estas siguientes especificaciones técnicas en cuenta:

Bibliotecas y Dependencias:

Utilizadas

* + Python
  + Music21
  + Ultralytics
  + Tesseract OCR
  + Fluter
  + Torchvision

Para la aplicacion:

* + camera
  + fluter\_vision
  + image\_picker
  + fluter\_speed\_dial

**Google Colab:** Versión actual en la nube (se accede vía navegador)

**Otros**:

Se utiliza Google Drive para el almacenamiento y acceso a los datos.

* 1. Lenguajes de programación, frameworks, entre otros. ix. Otros aspectos importantes
  + *Python* es el lenguaje principal utilizado en el desarrollo del proyecto. Python se destaca en el campo de la inteligencia artificial y el aprendizaje automático debido a su amplia gama de bibliotecas y frameworks disponibles.
  + Flutter es un marco de desarrollo de código abierto de Google que utiliza el lenguaje de programación Dart para crear aplicaciones multiplataforma con interfaces de usuario atractivas y fluidas. Permite desarrollar aplicaciones para Android, iOS, web y escritorio, todo desde una base de código común.
  1. Plan de trabajo

El plan para nuestro proyecto será el siguiente:

* Adquisición de Datos y preprocesamiento de Datos:

Recopilación de una amplia variedad de imágenes de hojas de partituras musicales fotocopiadas.

Aplicación de técnicas de preprocesamiento de imágenes para mejorar la calidad y claridad de las partituras.

* Diseño de la Arquitectura, Entrenamiento y Evaluación del Modelo:

Diseño de la arquitectura del modelo para el reconocimiento de figuras musicales en las imágenes de las partituras.

Entrenamiento del modelo utilizando conjuntos de datos de entrenamiento y validación.

Evaluación del rendimiento del modelo y métricas de reconocimiento de figuras musicales.

* Diseño de la Interfaz de Usuario:
* Desarrollo de una interfaz de usuario amigable que permita a los usuarios cargar imágenes de partituras y ver los resultados del reconocimiento de figuras musicales.
* Pruebas y Validación:

Realización de pruebas exhaustivas para garantizar la precisión y confiabilidad del sistema en la identificación de figuras musicales en hojas fotocopiadas.

* Despliegue y Monitoreo:

Despliegue del sistema en un entorno operativo.

Establecimiento de un sistema de monitoreo para asegurar el correcto funcionamiento continuo del sistema y la detección temprana de posibles problemas.

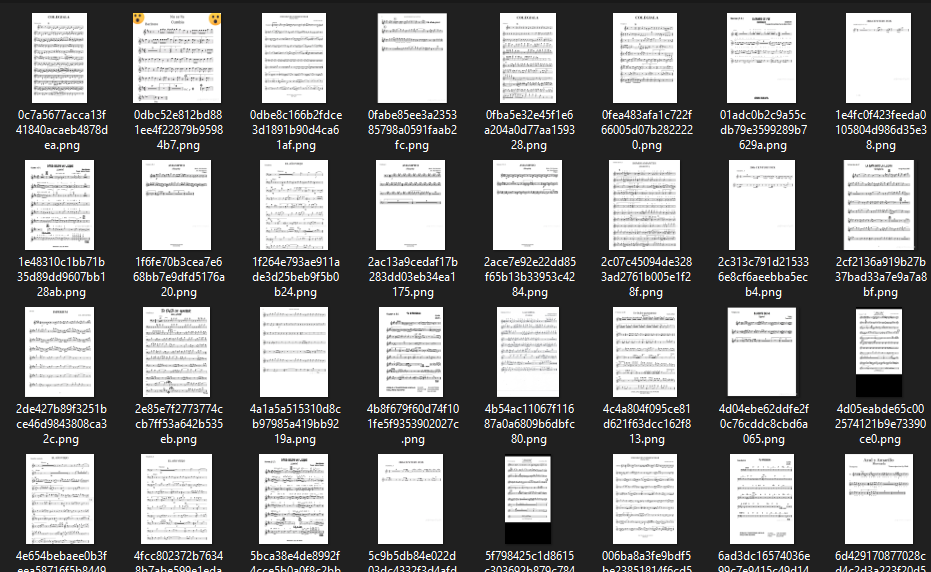
* 1. Cronograma

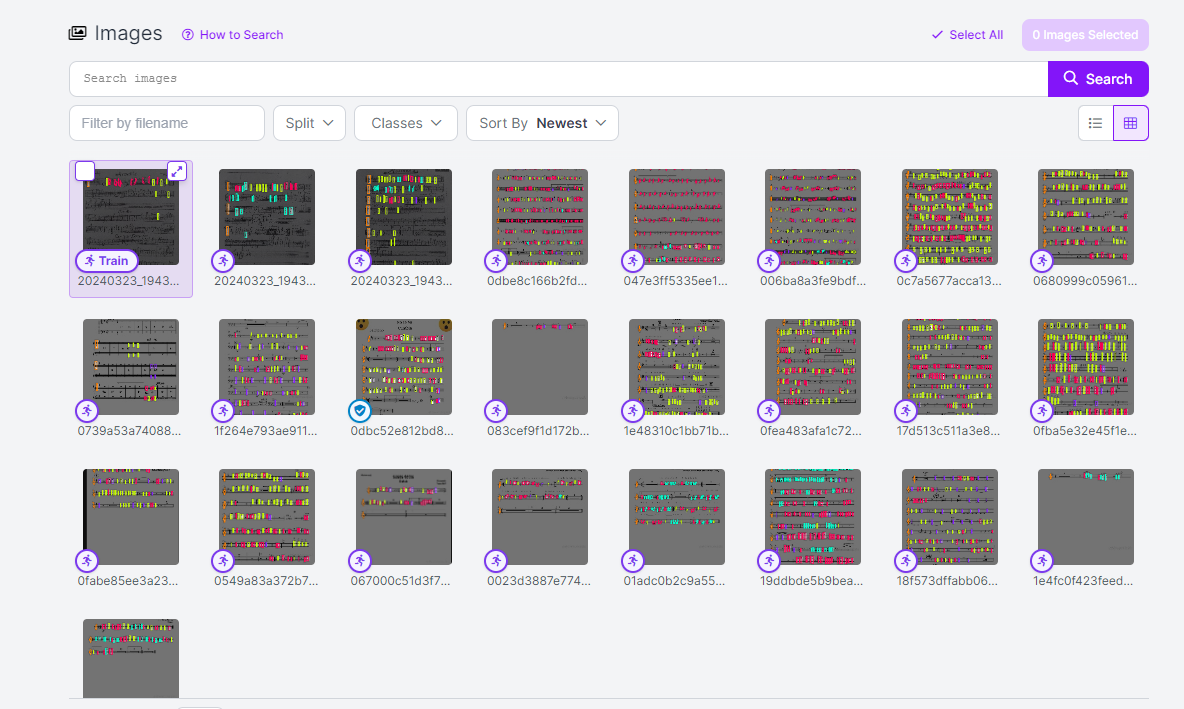
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Fases del Proyecto** | **Se1** | **Se2** | **Se3** | **Se4** | **Se5** | **Se6** | **Se7** |
| Desarrollar el Dataset de Figuras  Musicales | X |  |  |  |  |  |  |
| Preparación de Datos |  | X |  |  |  |  |  |
| Implementación del Modelo  YoloV8 para el Entrenamiento |  |  | X |  |  |  |  |
| Realización de Pruebas al Modelo  Entrenado |  |  |  | X |  |  |  |
| Diseño de una Interfaz en VSCode para Android |  |  |  |  | X |  |  |
| Creación e Implementación de una API con almacenamiento para posteriormente manipular en MuseScore |  |  |  |  |  | X |  |
| Realización de Pruebas en la APP  Desarrollada |  |  |  |  |  |  | X |

# **RESULTADOS**

* 1. Dataset
     1. Descripción y preprocesamiento realizado (Evidencia del antes y del después)

El preprocesamiento de los datos se llevó a cabo en varias etapas. Inicialmente, se realizó una exhaustiva investigación para identificar las figuras musicales en un pentagrama, que formarían parte del dataset. Posteriormente, se emprendió la tarea de adquirir al menos 40 imágenes donde se tratarían de implementar 1000 por cada clase de figuras necesarias para el proyecto cabe recalcar que las clases son: Clave de sol,blanca,negra,,corchea,semicorchea.  
  
  
  
Una vez recopiladas las imágenes, se procedió a su clasificación según las diferentes clases de figuras musicales, lo que marcó el inicio del proceso de preprocesamiento de los datos.

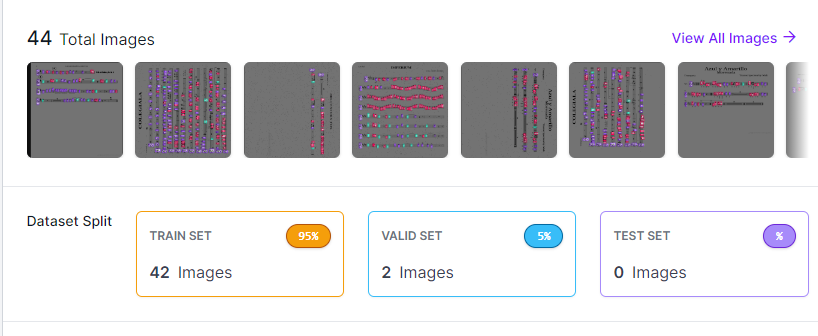
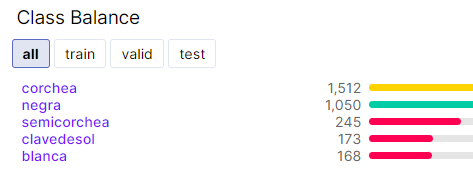
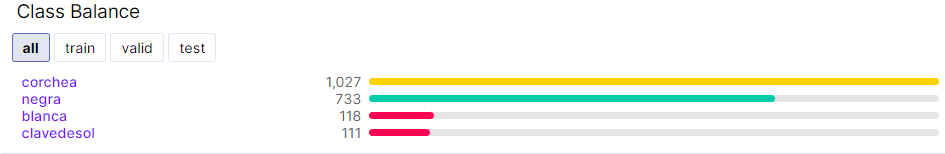




Una vez obtenidas las imágenes, se optó por emplear la herramienta de **Roboflow** para llevar a cabo la segmentación de las mismas. Esta elección se basó en su capacidad para facilitar el proceso de anotación de objetos en las imágenes, permitiendo una segmentación precisa de las diferentes partes de las plantas medicinales garantizando que el conjunto de datos estuviera adecuadamente etiquetado y listo para su uso en el entrenamiento del modelo de visión por computadora.

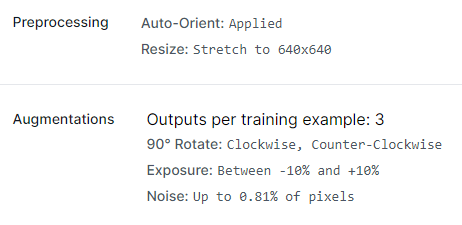
* + 1. Conjunto de entrenamiento, evaluación y validación

En este proceso se hizo 2 formas de desarrollo donde 26 imágenes se haga la forma de segmetacion y el resto de tipo object detection y posteriormente se aplico Data Augmentation, para la segmetacion tiene 72 train y 1 valid y el object detection tiene 42 train y 2 valid



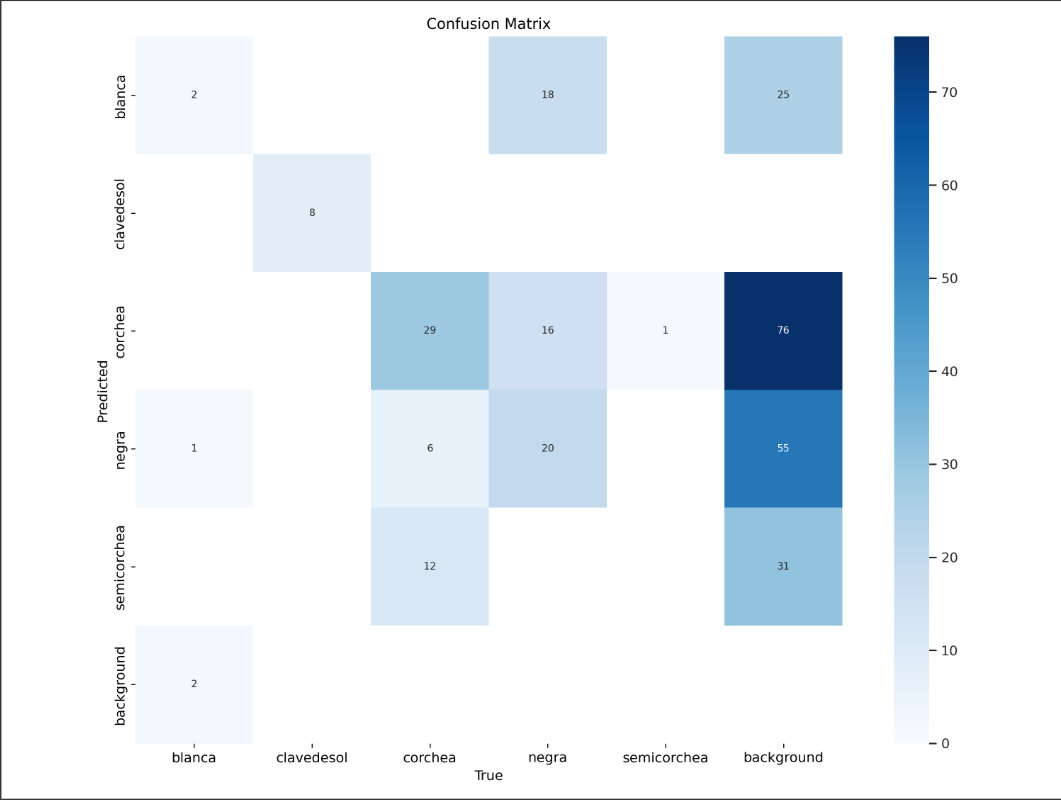
* + 1. Técnicas, criterio y/o métodos aplicados para la conformación de los conjuntos de datos de entrenamiento, prueba y validación.

Se incluyeron el uso de las opciones de Roboflow para realizar el aumento de datos. A través de la versión gratuita de esta herramienta, logramos duplicar nuestro conjunto de datos.

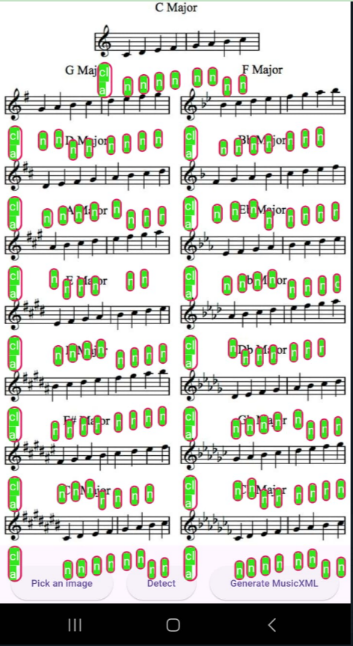
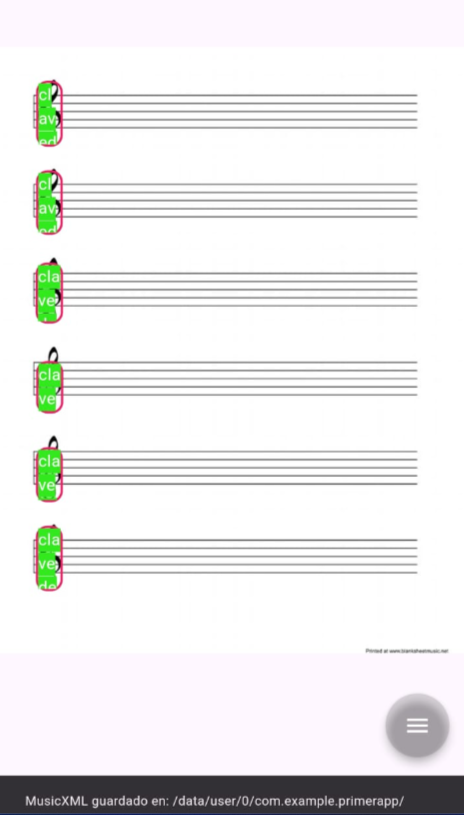


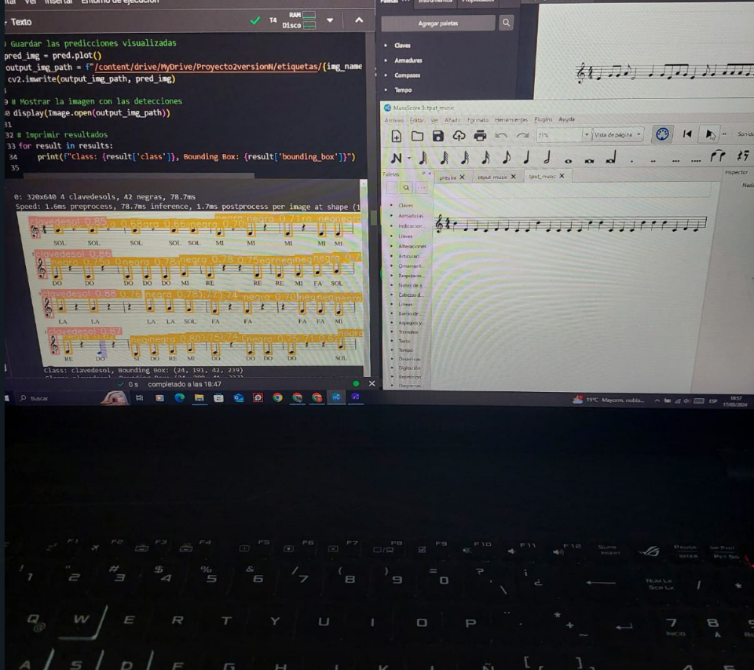
* + 1. Información adicional que considere importante incluir.

Inicialmente, se recolectaron 347 imágenes para completar por cada clase de partituras musicales, conforme al plan establecido. Sin embargo, al analizar este conjunto inicial de imágenes, se determinó que la diversidad era insuficiente para garantizar un entrenamiento efectivo del modelo. Por consiguiente, se implementaron técnicas de Data Augmentation de manera uniforme en todas las clases. Este proceso permitió enriquecer significativamente para ciertas clases pero lastimosamente en las partituras la figuras que mas utilizan son negras corcheas y semicorcheas, son pocas veces utilizadas las blancas y claves de sol ya que tienen un cierto punto de tiempo en el pentagrama

* 1. Resultados de entrenamiento y prueba (Métricas de rendimiento)
     1. Matriz de confusión  
        La **matriz de confusión** muestra cómo de bien clasifica el modelo las imágenes según las clases que tienen. Cada clase tiene una fila y una columna en la matriz. Los valores de la diagonal son los que el modelo acierta, y los valores fuera de la diagonal son los que el modelo se equivoca.

## Resultados de aplicación y utilización





# CONCLUSIONES

El desarrollo de este proyecto se enfocó en alcanzar cuatro objetivos fundamentales:

* **Estudio del estado del arte en la Identificación de figuras musicales**

Durante la investigación y revisión del estado del arte en la identificación de notas musicales, se ha logrado varias formas para una buena comprensión de la aplicación de inteligencia artificial para este propósito.

* **Creación de un Dataset**

La creación de un conjunto de datos especifico de figuras musicales que están en conjunto de hojas de papel ha sido muy escaso. La recopilación y preparación de las imágenes sentó las bases para el entrenamiento del modelo.

* **Desarrollo del modelo de detección de Figuras musicales con YOLOv8**

La implementación y entrenamiento del modelo YOLOv8 para la detección de figuras musicales es un paso significativo en el proyecto. El modelo ha demostrado su capacidad para procesar imágenes y realizar tareas precisas de identificación siempre y cuando la imagen este bien clara y no muy borrosa y solo este la hoja de papel enfocado.

* **Validación y evaluación del modelo**

Los resultados obtenidos al evaluar el modelo son prometedores. La métrica de precisión muestra un alto rendimiento, destacando la efectividad del modelo en su tarea.

Sin embargo, el enfoque basado en YOLO, nos ha permitido entender el trabajo que conlleva un proyecto de investigación. Para el desarrollo del mismo se han investigado más a fondo varios conceptos explicados en la carrera y otros no contemplados, los cuales he tenido que entender e implementar para replicar YOLO basándonos en la publicación oficial. Como conclusión, aunque no haya resultados tangibles del mismo, hemos adquirido diferentes conocimientos.

# RECOMENDACIONES

Como trabajo futuro, en primer lugar, se deberá revisar si la función de pérdida implementada está funcionando de forma correcta en YOLO. Además, otros aspectos que podrían tratarse en el enfoque YOLO serían los siguientes:

• Investigar diferentes configuraciones para el proyecto como puede ser aplicar diferentes funciones de pérdida customizadas.

• Realizar cambios significativos a nivel de red como reducción de capas convolucionales para intentar obtener un resultado con un dataset más pequeño.

• Aplicar aumentado de datos con el fin de entrenar la red con mayor cantidad de datos.

• Ajustar parámetros distintos a los usados para comprobar cómo se comportan diferentes modelos desconocidos.

# REFERENCIAS BIBLIOGRAFICAS (Harvard o APA)

* Ultralytics. Uso de la interfaz de línea de comandos. Creado 2023-11-12, Actualizado 2024-03-01, <https://docs.ultralytics.com/es/usage/cli/>
* Ultralytics. Modelo de entrenamiento con Ultralytics YOLO. Creado 2023-11-12, Actualizado 2024-03-31, <https://docs.ultralytics.com/es/modes/train/>
* Ultralytics. Configuración. Creado 2023-11-12, Actualizado 2024-03-31, <https://docs.ultralytics.com/es/usage/cfg/>
* Flutter, Start building Flutter Android apps on Windows. <https://docs.flutter.dev/get-started/install/windows/mobile?tab=first-start>
* Flutter, flutter\_vision 1.1.4. 19-9-2023. <https://pub.dev/packages/flutter_vision>
* CLAUDIA PAOLA TORRES RODRÍGUEZ  y JUAN CAMILO MONTAÑEZ OSORIO  
  https://view.genially.com/61854bad138f420d9d988f1c/interactive-content-reconocimiento-de-partituras
* Método de identificación automática para reconocimiento de elementos musicales presentes en imágenes 2D de partituras usando técnicas de aprendizaje profundo.  
  https://repository.ucatolica.edu.co/entities/publication/287128c1-475d-4c22-a8fe-dfd728ea3848

# ANEXOS