

UNIVERSIDAD NACIONAL DE INGENIERÍA
FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA



**Proyecto: Sistema de Detección y Registro de
Movimientos en Partidas de Ajedrez Mediante YOLOv 8**

Integrantes:

- Cardenas Tintaya Diego (20221165G)
- Rodriguez Vasquez Brando Kevin (20160359A)

Curso:

- ingeniería de Software

Docente:

- Yury Oscar Tello Canchapoma

LIMA, PERÚ

2025

Índice

1.- Introducción:.....	3
2.- Otros proyectos similares:.....	3
3.- Objetivos:	3
3.1.- Objetivo general:.....	3
3.2.- Objetivos específicos:.....	3
4.- Cronograma:	4
5.- Diseño del proyecto:	6
5.1.- Diagrama De flujo:	6
7.- Conclusiones	8

Sistema de Detección y Registro de Movimientos en

Partidas de Ajedrez Mediante YOLOv 8

1.- Introducción:

El ajedrez es un juego milenario que combina estrategia, lógica y habilidad táctica. En el contexto de la inteligencia artificial y la visión por computadora, la detección automática de piezas sobre un tablero físico representa un desafío interesante y útil para sistemas de análisis, entrenamiento o transmisión en tiempo real.

Este proyecto tiene como propósito principal el desarrollo de una aplicación que permita capturar video desde una cámara o archivo, detectar automáticamente las piezas de ajedrez utilizando un modelo entrenado con YOLOv8 y transformar sus coordenadas al plano del tablero para representar la partida de manera digital. A través de la biblioteca python-chess, se valida el estado del juego y se genera un tablero virtual que refleja fielmente la posición real de las piezas.

La solución implementada integra técnicas de procesamiento de video, transformación de perspectiva, segmentación del tablero y validación de estabilidad para asegurar la consistencia de las detecciones. Todo el flujo está automatizado y visualizado en tiempo real, lo que permite un análisis práctico de partidas físicas sin necesidad de intervención manual.

2.- Otros proyectos similares:

Para el desarrollo de este sistema de detección y validación de movimientos de ajedrez en un tablero físico, tomé como referencia el repositorio “raspberry-chess” del usuario ale93111 (<https://github.com/ale93111/raspberry-chess?tab=readme-ov-file>), ya que presenta una arquitectura funcional y modular orientada al reconocimiento visual de piezas y la sincronización con la librería python-chess.

3.- Objetivos: .

3.1.- Objetivo general:

- Desarrollar un sistema de reconocimiento automático de piezas de ajedrez en video mediante visión artificial, capaz de proyectar la posición de las piezas a un tablero virtual validado con reglas del juego.

3.2.- Objetivos específicos:

- Implementar la detección de piezas sobre un tablero de ajedrez físico usando un modelo YOLOv8 previamente entrenado.
- Aplicar una transformación de perspectiva para mapear las posiciones detectadas al espacio lógico del tablero (casillas A1–H8).
- Estabilizar las detecciones mediante un filtro temporal que reduzca falsos positivos.
- Visualizar simultáneamente el tablero físico con las detecciones y un tablero digital representando el estado de la partida.

4.- Cronograma:

Semana 1: Captura de Video desde IP Webcam

Durante la primera semana se estableció la base del sistema visual mediante la implementación de un módulo de captura de video en tiempo real a partir de una cámara IP. Se desarrolló un script capaz de abrir el stream de video, visualizar los frames capturados junto con una métrica de FPS para evaluar la eficiencia..

Semana 2: Detección de Piezas

En la segunda semana se integró un modelo de detección de objetos (YOLOv8) entrenado específicamente para identificar piezas de ajedrez en imágenes. Inicialmente, se realizaron pruebas con imágenes estáticas para validar la precisión del modelo y su capacidad de diferenciar tipos de piezas. Luego, se integró con el módulo de captura para operar sobre video en tiempo real. Se implementó un sistema de dibujo de los cuadros delimitadores sobre las piezas detectadas, así como un mecanismo de control de velocidad con delay para asegurar una tasa de FPS aceptable. Se validó que las detecciones alcanzaran una confianza suficiente, haciendo uso del archivo best.pt, y se logró una integración funcional entre la detección y la visualización, sentando las bases del seguimiento de piezas en movimiento.

Semana 3: Detección de Esquinas

Para permitir el mapeo correcto del tablero real a coordenadas virtuales, en la tercera semana se desarrolló una parte del código para designar las esquinas del tablero, el usuario puede marcar manualmente las esquinas visibles del tablero en un frame capturado. Esto permitió mantener la flexibilidad en escenarios con tableros físicos de diferentes tamaños o perspectivas, garantizando precisión en la alineación espacial del sistema. La validación se centró en asegurar que los puntos se almacenen correctamente y correspondan con esquinas reales del tablero físico.

Semana 4: Transformación de Coordenadas

Durante la cuarta semana se abordó la transformación de coordenadas desde el espacio de píxeles al sistema lógico del ajedrez (casillas de A1 a H8). A partir de las esquinas detectadas.

Semana 5: Tablero Virtual

En la quinta semana se implementó un tablero virtual que replica el estado del tablero físico, utilizando el mismo OpenCV. Se desarrolló un módulo que permite visualizar la ubicación actual de las piezas detectadas en una representación digital. El sistema se diseñó para actualizarse dinámicamente en función de los datos generados por el modelo de detección y la transformación de coordenadas. Esta funcionalidad fue clave para empezar a realizar un seguimiento lógico del juego.

Semana 6: Mejora del Tablero Virtual con python-chess

La sexta semana estuvo enfocada en optimizar el tablero virtual sustituyendo la visualización artesanal basada en OpenCV por una representación formal basada en python-chess. Esta decisión permitió aprovechar la lógica del motor de ajedrez para validar legalidad de movimientos, actualizar el FEN automáticamente y facilitar la sincronización entre detecciones físicas y la lógica del juego. La integración con python-chess también permitió una mayor precisión en la representación del estado del tablero, además de facilitar tareas como la validación automática de jugadas legales e ilegales, característica clave para la consistencia del sistema.

Semana 7: Documentación y Preparación de Entrega

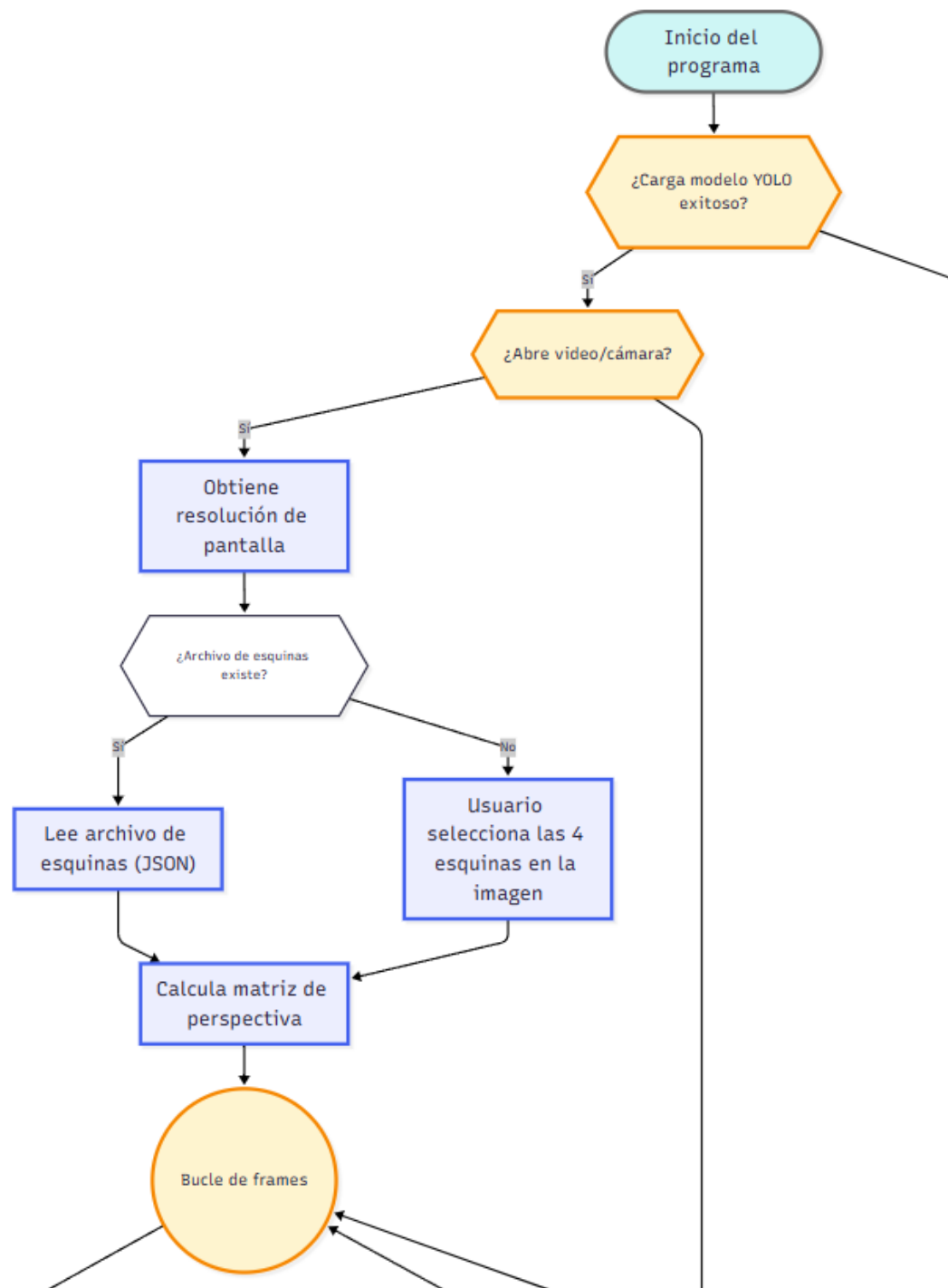
La séptima semana se dedicó a la elaboración y organización de la documentación técnica. Se documentaron todos los módulos desarrollados, incluyendo los scripts de captura, detección, marcaje de esquinas, transformación de coordenadas, y generación del tablero virtual.

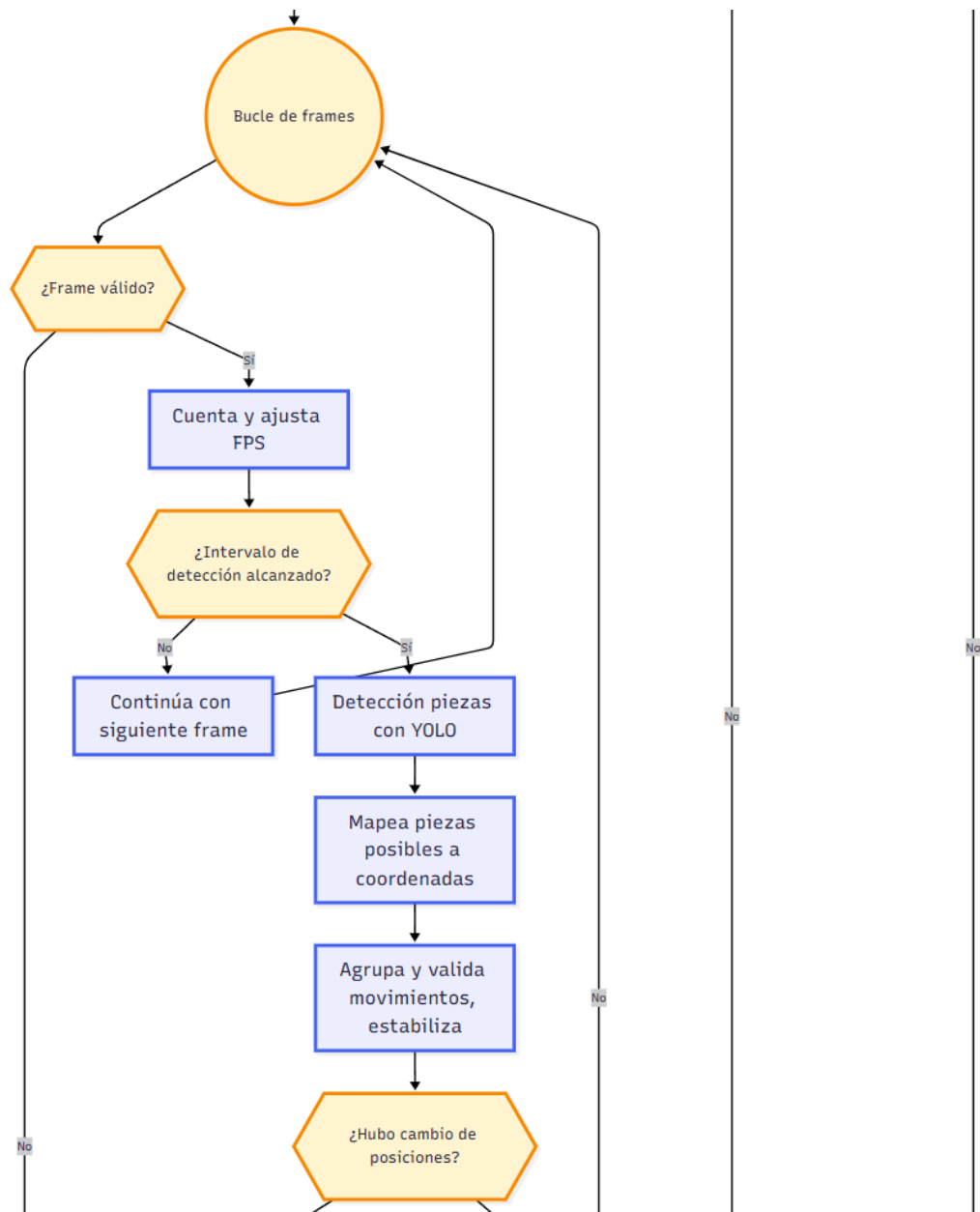
Semana 8: Entrega Final y Revisión

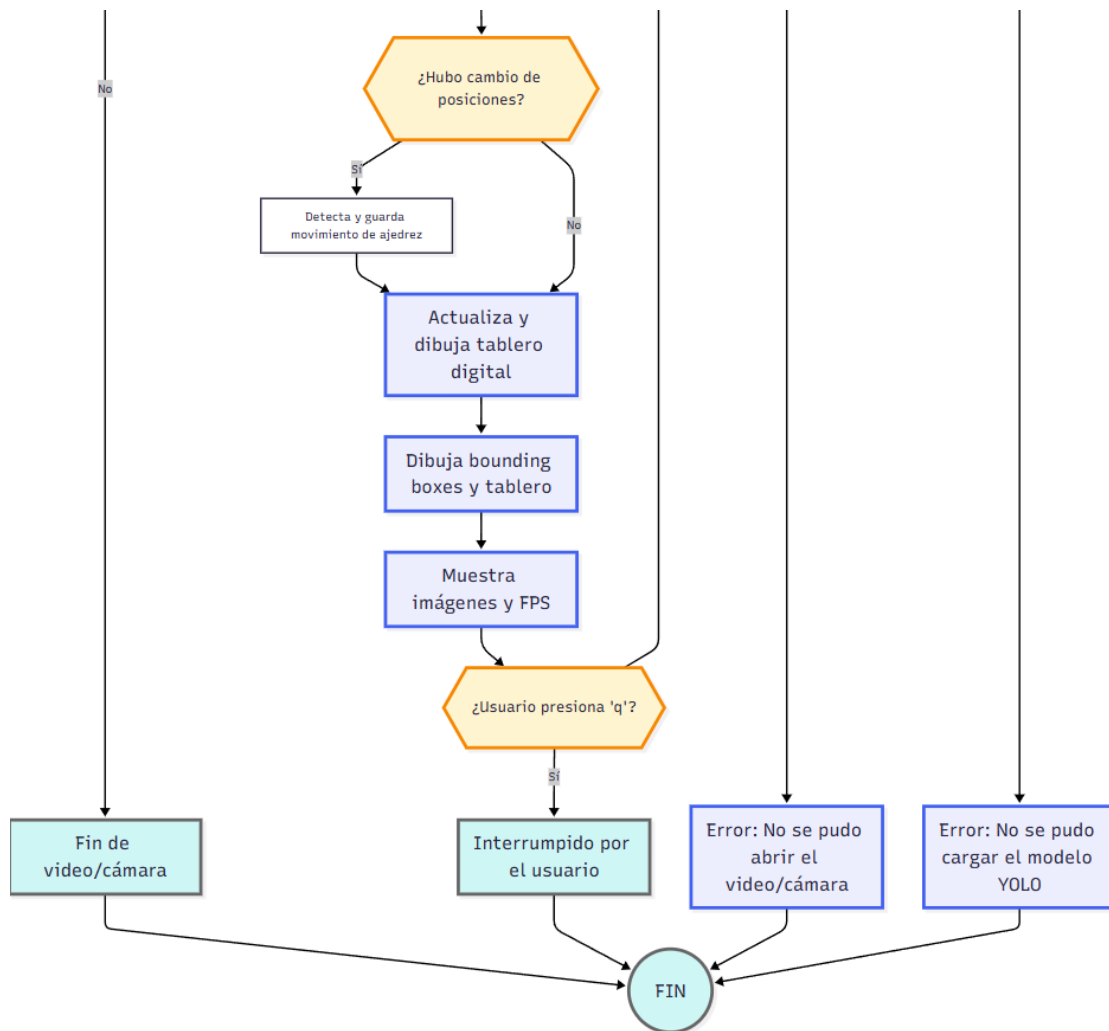
En la última semana se preparó y realizó la entrega formal del proyecto. Esto incluyó una revisión completa de todo, pruebas funcionales del sistema completo y preparación de una presentación para demostrar su funcionamiento. Se validó el comportamiento del sistema en tiempo real, desde la detección de piezas hasta la actualización lógica del tablero digital. Se incorporaron ajustes menores en función de la retroalimentación obtenida y se dejó el sistema en un estado estable y documentado, listo para ser evaluado por revisores o usuarios finales.

5.- Diseño del proyecto:

5.1.- Diagrama De flujo:







7.- Conclusiones

1. El desarrollo del sistema de seguimiento de ajedrez en tiempo real permitió integrar múltiples tecnologías de manera efectiva, demostrando la viabilidad de combinar visión por computadora y lógica ajedrecística en una sola aplicación funcional.
2. Durante el proyecto se logró capturar video desde una cámara IP con buena estabilidad, lo cual fue clave para el procesamiento en tiempo real. La integración del modelo YOLO permitió detectar las piezas con un nivel de precisión aceptable, marcando un punto de partida sólido para aplicaciones de visión artificial aplicadas a juegos de tablero.
3. En términos generales, el proyecto alcanzó un nivel funcional avanzado, superando desafíos técnicos relacionados con estabilidad, precisión de detección y sincronización. La modularidad del sistema y el uso de bibliotecas como OpenCV, YOLOv8 y python-chess facilitaron el desarrollo progresivo y validado por etapas.

4. Finalmente, se demostró que es posible construir un sistema capaz de asistir, monitorear o incluso arbitrar una partida de ajedrez en un entorno físico, abriendo posibilidades para desarrollos futuros como análisis automático, interfaces de asistencia o entrenamiento inteligente.