



Resolver un reto real de IA y accesibilidad: mi experiencia en el Hackathon Kiro AWS × UCJC



El martes 29 de septiembre me levanté antes de lo habitual. No por los nervios del examen, que llevo tiempo aprendiendo a gestionar, sino por algo diferente: iba a pasar el día construyendo tecnología real para personas que la necesitan de verdad.

El Hackathon Kiro , organizado por AWS junto con la UCJC en el marco de la Cátedra AWS Emilio Lora-Tamayo, y con Capgemini como socio técnico, reunió en el Campus Castellana a más de 200 estudiantes con un único objetivo : resolver en una jornada un reto propuesto por la ONCE. El objetivo: hacer que la ONCE lleve el ajedrez a personas con discapacidad visual utilizando inteligencia artificial.

El reto de la ONCE: inteligencia artificial al servicio del ajedrez accesible

El auditorio del Campus Castellana tiene esa energía particular que solo se da cuando hay muchos ordenadores encendidos y la misma cantidad de personas con ganas de hacer algo. Éramos más de 200 estudiantes de grados de informática, robótica, teleco y de FP.

En mi caso, participaba como alumno de la de la UCJC, una combinación académica que en ese momento me parecía más útil que nunca.

La apertura fue directa. El Rector, Jaime Olmedo, inauguró la jornada destacando el valor de enfrentarnos a problemas reales: “ la tecnología en mejor versión: resolver los problemas de las personas ”. Enseguida, el equipo de AWS presentó Kiro , su entorno de desarrollo con IA integrada , y se anunció oficialmente el reto de la ONCE: entrenar un sistema capaz de reconocer un diagrama de ajedrez a partir de una imagen y traducir la posición a un formato que una persona ciega pueda entender y usar

No se buscaba una descripción genérica de la imagen, sino una traducción técnica y estructurada al lenguaje propio del ajedrez : casillas, piezas, notación algebraica y una salida compatible con



lectores de pantalla y sistemas de síntesis de voz.

El reto sonaba claro. En cuanto empecé a pensar en la arquitectura, me di cuenta de que no lo era tanto.

El desafío técnico: por qué no basta con describir la imagen con IA

La trampa en la que caería cualquier solución perezosa era evidente: llamar a un modelo de visión genérico, pedirle una descripción global del tablero y entregar eso como resultado.

Esa aproximación no resulta útil para una persona ciega que quiere seguir una partida posición a posición. Lo que necesita es saber exactamente dónde está cada pieza, qué jugada acaba de ocurrir (si es que puede deducirse) y recibir esa información en un formato estructurado, sin ambigüedad, que su tecnología de asistencia pueda leer en el orden correcto.

El reto técnico planteaba tres capas clave:

Modelado e identificación : Definir las variables del tablero (64 casillas, hasta 32 piezas de dos colores) e interpretar correctamente las distintas notaciones.

Reconocimiento visual : Lograr que la visión por computador identifique con precisión las piezas sobre fuentes heterogéneas: capturas de Lichess, fotos de libros impresos o imágenes de torneos.

Traducción e interfaz accesible : Convertir la posición a notación algebraica de ajedrez mediante la signografía braille de la Comisión Braille Española y generar una narración por voz clara para lectores de pantalla.

Esta tercera capa de accesibilidad e impacto social era la que marcaba la diferencia técnica y el aspecto con mayor peso en la evaluación del jurado.

Sprint 1: Arquitectura con Kiro AWS como copiloto de desarrollo

Las 11:00 marcaron el inicio del Sprint 1: dos horas dedicadas a la ideación, diseño de arquitectura y preparación del código . Antes de programar, utilicé Kiro AWS para estructurar el proyecto mediante su sistema de specs (requirements, design, tasks) y su mecanismo de steering, el cual permite establecer reglas claras que guían al agente de IA durante el desarrollo. El jurado valoraba el proceso tanto como el resultado

La arquitectura del sistema quedó configurada de la siguiente forma:

Entrada de datos : Procesamiento de archivos en formato PNG, JPG o PDF con diagramas de ajedrez.

Procesamiento visual : Pipeline en Python empleando OpenCV para localizar el tablero, corregir la perspectiva (transformación homográfica), recortar las 64 casillas y clasificar las piezas.

Lógica de ajedrez : Integración con la librería python-chess para validar posiciones, verificar jugadas legales y generar la notación estándar.



Canales de salida accesibles Braille Unicode : Aplicación del bloque U+2800 según el Documento Técnico B8 de la Comisión Braille Española.

Síntesis de voz : Generación de audio mediante pyttsx3 integrado en una interfaz web construida con Streamlit.

Entorno : Proyecto desplegado en Python 3.11 con control de versiones en GitHub.

Gracias a la definición previa del steering en Kiro , fue posible descomponer la arquitectura en tareas concretas y priorizadas en menos de 15 minutos, acelerando los tiempos habituales de desarrollo en hackathons.

Almuerzo y networking: mentoría directa con AWS, Capgemini y profesores UCJC

A las 13:00 se realizó la pausa para el almuerzo. Durante 45 minutos, los Technical Account Managers de AWS , especialistas de Capgemini y profesores de la UCJC compartieron impresiones con los equipos. No como figuras de autoridad evaluando tu código, sino como ingenieros con ganas de ver qué estabas construyendo y de contarte cómo lo harían ellos

Una de las conversaciones clave con los mentores abordó la gestión de errores en la entrada de datos : cómo proceder cuando la imagen carece de indicación de jugada previa. La recomendación técnica fue garantizar siempre una salida útil y parcial (la descripción de la posición actual) antes que bloquear el sistema, una lección práctica en el diseño de productos accesibles.

Sprint 2 y Sprint 3:

Visión por computador con OpenCV, Streamlit y signografía braille

Las dos horas del Sprint 2 fueron las más intensas.

El flujo básico de imagen → posición braille → voz funcionó en los primeros 40 minutos sobre capturas digitales limpias de Lichess. Sin embargo, las fotografías de libros impresos presentaron retos adicionales: ángulos oblicuos, sombras e iluminación irregular. OpenCV necesitaba más trabajo del esperado para corregir la homografía del tablero en esas condiciones y Kiro fue clave aquí.

Para solucionar la deformación de la imagen , se configuró el módulo de visión por computador describiendo el problema a Kiro («detectar esquinas del tablero en perspectiva oblicua y aplicar transformación homográfica»), generando el algoritmo basado en contornos y cv2.getPerspectiveTransform, lo que resolvió el encuadre en las pruebas de validación.

Durante el Sprint 3 (15:45 a 16:45) el foco se centró en la usabilidad y la accesibilidad web:

Navegación completa por teclado en la interfaz Streamlit (teclas Tab, Enter y cursores).

Contraste cromático elevado para usuarios con baja visión o magnificadores de pantalla.

Verificación del audio con el Narrador de Windows.



Adaptación del orden de la narración a las normas de la Comisión Braille Española: pieza, letra de columna y número de fila en posición baja.

Presentaciones finales: demostraciones en vivo y criterios de evaluación

A las 17:00, después de la presentación de Capgemini sobre oportunidades laborales y programas de incorporación para recién graduados en tecnología, comenzaron los pitches finales . Los 10 proyectos seleccionados dispusieron de tres minutos para realizar una demo en directo ante el jurado.

Escuchar las demos de otros fue tan instructivo como construir la mía.

Las propuestas mostraron distintos enfoques de ingeniería : soluciones con gran precisión en visión por computador pero menor estructuración de datos para lectores de pantalla, frente a proyectos centrados en la experiencia de usuario con procesamiento de imágenes más simplificado.

El jurado evaluó las propuestas bajo cuatro criterios : innovación, viabilidad técnica, impacto social e integración del entorno Kiro AWS.

Aprendizajes: la ingeniería con propósito y el valor de la formación práctica en la UCJC

Hay algo que no aparece en ninguna ficha de sprint ni en ningún criterio de evaluación: el hecho de que el destinatario de lo que construíamos era una persona real, con una necesidad concreta , y que el éxito no se medía en líneas de código sino en si esa persona podría escuchar una jugada de ajedrez y entenderla.

Eso cambia la forma de tomar decisiones técnicas : cuando el usuario final requiere herramientas de accesibilidad, las decisiones de diseño priorizan la usabilidad real y el cumplimiento de estándares por encima de soluciones innecesariamente complejas.

La formación que recibo en el Doble Grado en Ingeniería Telemática e Ingeniería Robótica e Inteligencia Artificial de la UCJC me aportó una visión global para abordar el reto : desde la arquitectura de comunicación entre módulos y la integración de la visión por computador, hasta la entrega de datos en formatos accesibles.

Si hubiera que resumir el día en una frase , sería esta: aprendí más sobre para qué sirve la tecnología en ocho horas de hackathon que en muchas horas de clase . Y eso, precisamente, es lo que hace que valga la pena estudiar en la UCJC

Participar en iniciativas como el Hackathon Kiro AWS demuestra el valor de aplicar los conocimientos del aula a proyectos reales de Inteligencia Artificial con impacto social directo.