FUTURE STEM HUB

Materiales educativos con práctica Python y tareas de Scratch

FUTURE-STEM-HUB

Empoderamiento de la educación STEM en secundaria con formación en IA y recursos para estudiantes y educadores

Proyecto Nº
2024-1-DE03-KA220-SCH-000247346

Co-funded by the European Union

MATERIALES EDUCATIVOS CON PRÁCTICA PYTHON Y TAREAS DE SCRATCH

El proyecto «Empoderamiento de la educación STEM en secundaria con formación en IA y recursos para estudiantes y educadores / FUTURE-STEM-HUB» (n.º de ref.: 2024-1-DE03-KA220-SCH-000247346) está cofinanciado por el Programa Erasmus+ de la Unión Europea. Está coordinado por la Universidad de Duisburg-Essen (Alemania) y cuenta con la participación de otras cuatro organizaciones asociadas: M&M Profuture Training (España), la Dirección Provincial de Kütahya del Ministerio de Educación Nacional (Turquía), COOPETAPE (Cooperativa de Educación), CRL (organismo supervisor de la escuela ETAP) (Portugal) y Tetra Solutions Ltd. (Bulgaria).

Los miembros del equipo del proyecto, que representan a todas las organizaciones socias, desarrollan los materiales educativos de FUTURE-STEM-HUB con tareas prácticas de Python y Scratch. Cada socio fue responsable de la autoría de un módulo. Su objetivo es presentar los conceptos fundamentales de la IA a estudiantes de secundaria, fomentando la concienciación y el debate sobre sus implicaciones sociales y éticas.

Autores:
Mustafa Bilgin, Universidad de Duisburg-Essen (Alemania)
Mónica Moreno, M&M Profuture Training (España)
Montserrat Renedo, M&M Profuture Training (España)
João Barroso, Escuela ETAP (Portugal)
Angelina Presa, Escuela ETAP (Portugal)
Silviya Georgieva, Tetra Solutions Ltd. (Bulgaria)
Borislava Zaharieva-Tomova, Tetra Solutions Ltd. (Bulgaria)
Yeliz Yurter, MEM de Kütahya (Türkiye)
Özcan Turan, Kütahya MEM (Türkiye)

Editor:
Mustafa Bilgin, Universidad de Duisburg-Essen (Alemania)

Índice de ilustraciones

Plantilla de diseño: Tetra Solutions Ltd.
Imagen de portada: Creación híbrida (Concepto y boceto: Mustafa Bilgin;
Retoque digital: Generado mediante Google Gemini).
Pág. 6: Gráfico de Tetra Solutions Ltd.
Pág. 7, 8: Creación híbrida (Concepto y boceto: Mustafa Bilgin;
Retoque digital: Generado mediante Google Gemini).
Los iconos proceden de Flaticon.com (High quality icons).

Descargo de responsabilidad y seguridad

El uso de los códigos de programación proporcionados en estos materiales, así como el acceso a sitios web externos y plataformas de terceros, se realiza bajo el propio riesgo y responsabilidad de las personas usuarias. El equipo del proyecto no asume ninguna responsabilidad por el contenido, la funcionalidad o la disponibilidad de los enlaces externos, tales como Google Colab, Scratch o Machine Learning for Kids. Dado que muchas de las tareas interactivas requieren una conexión activa a Internet y el procesamiento de datos en servidores en la nube, la responsabilidad de cumplir con los estándares de seguridad y los requisitos del sistema (como versiones actualizadas del navegador) recae en los usuarios.

Al utilizar herramientas de terceros para la creación de modelos de IA, no se debe cargar información personal o sensible. Dado que los materiales educativos están dirigidos al grupo objetivo de jóvenes de entre 15 y 18 años, el registro en servicios externos y la realización de actividades en línea deben llevarse a cabo de acuerdo con las condiciones de uso de los respectivos proveedores e, idealmente, bajo la supervisión y el consentimiento de un docente o tutor legal. No se acepta ninguna responsabilidad por posibles pérdidas de datos o daños en los dispositivos finales utilizados derivados de la ejecución de los ejemplos de código proporcionados.

Esta obra está bajo una Licencia Creative Commons Atribución-NoComercial-CompartirIgual 4.0 Internacional (CC BY-NC-SA 4.0).

Nota sobre las versiones del software

Ten en cuenta que las versiones de Python, Google Colab y YOLO indicadas en los materiales eran las vigentes en el momento de elaborar el curso. Dado que el software se actualiza con frecuencia, es posible que con el tiempo estas versiones queden obsoletas. Si al realizar los ejercicios aparecen errores o un comportamiento inesperado, comprueba si hay disponible una versión más reciente de la herramienta correspondiente y actualízala antes de continuar.

Lista de abreviaturas:

AI: Inteligencia artificial

CNN: Red neuronal convolucional

CV: Visión por computadora

EACEA: Agencia Ejecutiva Europea de Educación y Cultura

FPS: Fotogramas por segundo

IBM: Máquinas de negocios internacionales

IDE: Entorno de desarrollo integrado

ML: Aprendizaje automático

ML4K: Aprendizaje automático para niños

MIT: Instituto Tecnológico de Massachusetts

MNIST: Instituto Nacional de Estándares y Tecnología modificado (conjunto de datos)

NN: Red neuronal

PNL: Procesamiento del lenguaje natural

RGB: Rojo, Verde, Azul (modelo de color)

PROVENIR: Ciencia, tecnología, ingeniería y matemáticas

TTS: Texto a voz

UNESCO: Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura

YOLO: Sólo miras una vez

Contenidos

Descripción general del proyecto 6

Resultados del proyecto 6

Introducción.....7

Módulo 1: Mini-desafíos de codificación:
Aprendizaje automático y redes neuronales.....9

Módulo 2: Búsqueda del tesoro en una sala de escape con IA y calculadora.....39

Módulo 3: Scratch se encuentra con la inteligencia artificial.....61

Módulo 4: Programación de juegos en Python
(bibliotecas de pygame/arcade) con
IA.....82

Módulo 5: Reconocimiento de objetos con Roboflow:
Introducción a la visión artificial.....109

Resumen y próximos pasos.....125

This publication is licensed under a Creative Commons Attribution - NonCommercial - NoDerivatives 4.0 International License

Empowering Secondary School STEM Education with AI
Training and Resources for Students and Educators
2024-1-DE03-KA220-SCH-000247346

Descripción general del proyecto

FUTURE-STEM-HUB

El proyecto FUTURE-STEM-HUB tiene como objetivo promover y facilitar la integración de temas de Inteligencia Artificial (IA) en la educación STEM en las escuelas secundarias 1) proporcionando materiales educativos que introduzcan conceptos de IA y sus implicaciones sociales; 2) ofreciendo recursos de aprendizaje prácticos para que los estudiantes exploren la IA usando programación Python; y 3) equipando a los profesores con apoyo para integrar la IA en la formación STEM de la escuela secundaria.

Resultados del proyecto

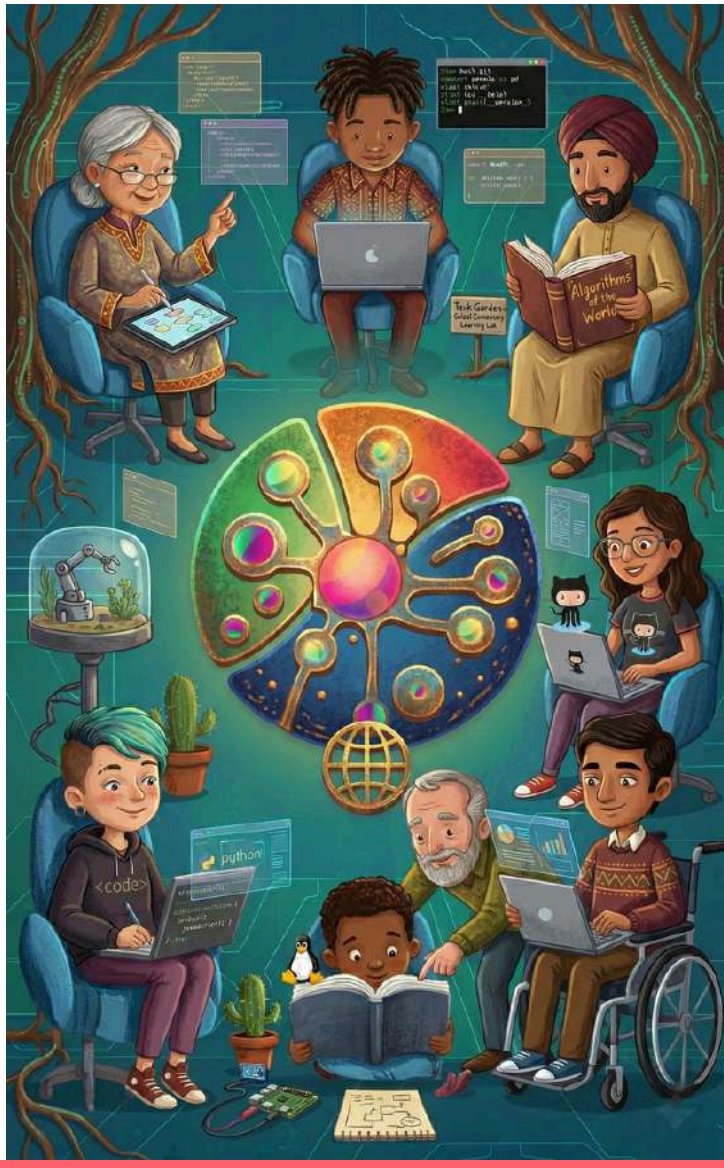
1. **Curso 1: Introducción digital: Fundamentos de inteligencia artificial** (Introducción a la IA a través de materiales educativos interactivos para estudiantes de secundaria)
2. **Curso 2: Profundizando en IA con Python y Scratch (IA avanzada:** materiales de aprendizaje práctico para estudiantes de secundaria)
3. **Kit de herramientas electrónico para educadores escolares:** Mejora de las habilidades en inteligencia artificial (Guía metodológica de IA para docentes de secundaria)

Introducción

Bienvenidos a los Materiales Educativos con Tareas Prácticas de Python y Scratch: una experiencia de aprendizaje atractiva e interactiva diseñada para introducir a estudiantes de secundaria al fascinante y práctico mundo de la IA. Para dotar a profesores y estudiantes de secundaria de materiales accesibles y prácticos que les permitan explorar conceptos clave de IA a través de Python y Scratch. Diseñados para estudiantes de 15 a 18 años con diversos niveles de competencia en STEM, los materiales son perfectos para el aprendizaje asincrónico a su propio ritmo y se integran perfectamente en la enseñanza presencial con la ayuda del profesor.

El contenido se divide en cinco módulos integrales, centrados en habilidades prácticas de programación en Python y Scratch, lógica matemática a través del aprendizaje basado en juegos (Escape Room), aplicaciones creativas de IA, resolución de problemas del mundo real con datos y consideraciones éticas.

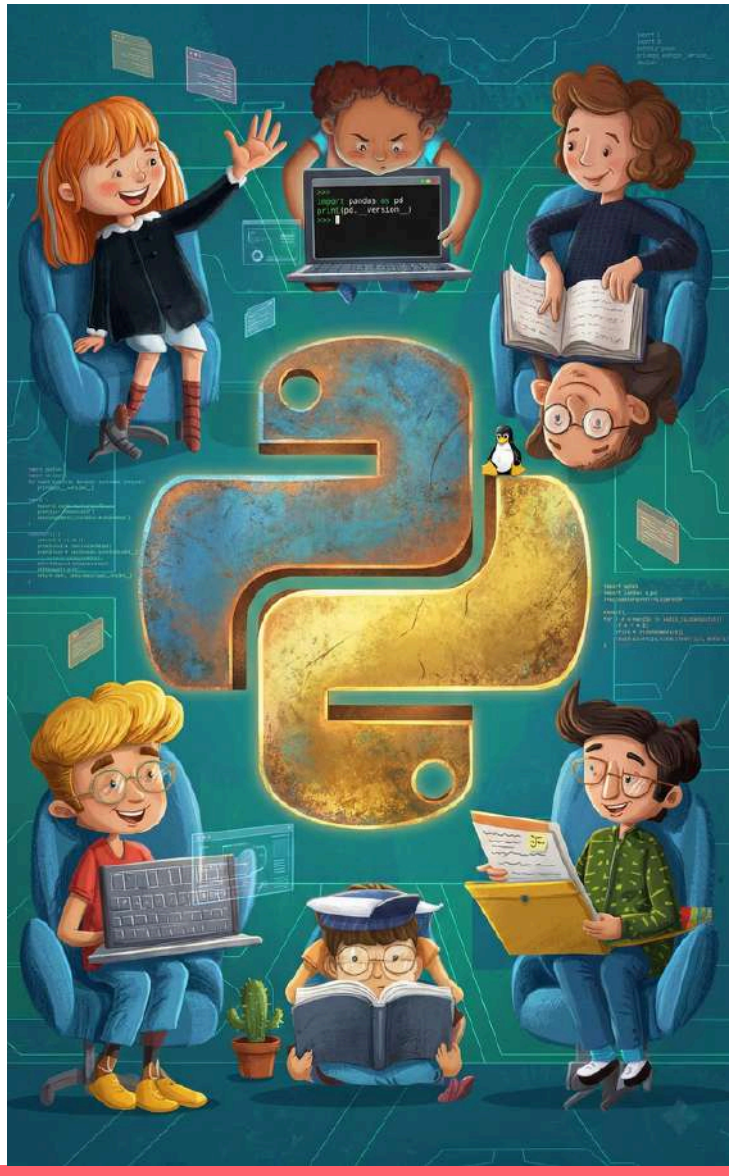
Cada módulo combina teoría, se basa en cuestionarios implementados, utiliza plataformas externas para profundizar en el tema e incluye ejercicios prácticos para garantizar una comprensión integral de los conceptos y metodologías de IA. La duración estimada del aprendizaje para los cinco módulos es de aproximadamente 10 horas.



Al finalizar el curso, los estudiantes podrán comprender y definir conceptos clave de IA, como aprendizaje automático, redes neutrales y aprendizaje profundo. Profundizarán en la historia y evolución de la Inteligencia Artificial y sus aplicaciones en diversas industrias, aprendiendo cómo las nuevas tecnologías de IA configuran las tendencias y el mundo actual. También profundizarán en tecnologías fundamentales, como la visión artificial y el procesamiento del lenguaje natural. Finalmente, explorarán y profundizarán en las consideraciones éticas de la IA, su responsabilidad y su impacto social.

Por otro lado, el profesorado tendrá acceso a recursos innovadores e interactivos que podrán usar y aplicar fácilmente en el aula de STEM, complementando así sus planes de estudio y enseñanza. Estos materiales les ayudarán a crear entornos de clase más interesantes y atractivos, ya que ofrecen diferentes enfoques para involucrar activamente al alumnado de secundaria en el proceso de aprendizaje.

Estos materiales se transformarán posteriormente en un curso en línea que se alojará en la plataforma FUTURE-STEM-HUB y será accesible a través del sitio web del proyecto:https://www.future-stem-hub.eu



Al completar con éxito el curso, los estudiantes recibirán un certificado digital que reconoce su logro.

Module 1: Mini-Coding-Challenges: Machine Learning and Neural Networks

INTRODUCCIÓN

¿Alguna vez te has preguntado cómo tu teléfono inteligente sabe qué videos te podrían gustar?
¿O cómo una aplicación puede reconocer tu escritura a mano? Detrás de todo eso está la Inteligencia Artificial (IA), ¡que es mucho menos misteriosa de lo que mucha gente piensa! En este capítulo, te unirás a Aylin y su ingenioso reloj inteligente, Alara. Juntos, te llevarán a aventuras que revelan cómo funciona realmente la IA. Desarrollarás una comprensión clara de los principios y componentes fundamentales de las Redes Neuronales (NN) y el Aprendizaje Automático (ML), incluyendo cómo procesan entradas, aprenden de los datos y toman decisiones. ¿La mejor parte? ¡No necesitas ser un experto en programación! Cada aventura incluye pequeños desafíos de codificación en tres niveles de dificultad. Simplemente elige el que más te convenga, desde principiante hasta un desafío real. Verás: después del primer capítulo, ya entenderás cómo las máquinas "aprenden a pensar". Sumerjámonos juntos en el mundo de la IA.

Al finalizar el módulo podrás adquirir diferentes habilidades, tales como:

- **Introducción a la IA**: El regalo misterioso: un reloj que hace más que marcar
- **Redes neuronales:** conceptos básicos: Alara despierta
- **Árboles de decisión**: El detective de la ruta escolar: cómo piensa tu reloj
- **Clasificación basada en reglas:** cómo Spotify podría saber lo que quieres escuchar
- **Aprendizaje por refuerzo:** aprendizaje mediante recompensas
- **Redes neuronales convolucionales**: cómo la IA detecta bordes en imágenes

- **Procesamiento del lenguaje natural:** cómo las computadoras entienden el lenguaje
- **Sistemas de recomendación:** cómo sabe YouTube lo que quieres ver
- **Bosque aleatorio:** Pronóstico del tiempo
- **Ética de la IA: Visión para el futuro:** Qué podemos y debemos hacer con la IA

Duración del módulo	**2 horas(1 hora de aprendizaje + 1 hora de ejercicios prácticos)**

MATERIALES EDUCATIVOS

AYLIN tiene 17 años, curioso, yLe encanta descubrir cómo funcionan las cosas. ¡Nunca se rinde!

LENA Es la mejor amiga de Aylin. Es creativa y le encanta la música. Su familia es de...Polonia;¡A veces trae deliciosos pierogi!

ALARA No es un reloj inteligente común. Es una IA que le enseña a Aylin sobre inteligencia artificial.

Figura 1 Personajes educativos Aylin, Alara y Lena; Concepto y diseño: Mustafa Bilgin; Mejorado con Google Gemini

UNIDAD 1.1 CONCEPTO Y ENFOQUE DEL CURSO

El proceso de aprendizaje se guía por un marco basado en historias que conecta temas STEM con contextos de la vida real con los que los estudiantes pueden identificarse. Conceptos complejos como las redes neuronales o el aprendizaje por refuerzo se representan en forma de prototipo simplificado, lo que permite a los estudiantes desarrollar una comprensión fundamental, sin necesidad de experiencia previa en programación. Este material se integra perfectamente en la educación informática y técnica, proyectos STEM, clubes extracurriculares o entornos de aprendizaje interdisciplinarios. No solo promueve la comprensión técnica, sino que también, a través del capítulo sobre Ética de la IA,...
—fomenta la reflexión crítica sobre cuestiones morales y sociales.

UNIDAD 1.2 INTRODUCCIÓN A LA INTELIGENCIA ARTIFICIAL (IA)

Antes de profundizar en las actividades, es fundamental comprender de qué se trata. La Inteligencia Artificial (IA) se refiere a la simulación de la inteligencia humana en máquinas programadas para pensar como humanos e imitar sus acciones. El término también puede aplicarse a cualquier máquina que exhiba características asociadas con la mente humana, como el aprendizaje y la resolución de problemas.
En este módulo, exploraremos los componentes fundamentales de la IA a través de historias interactivas:

- **Redes neuronales:**Se trata de sistemas informáticos inspirados en las redes neuronales biológicas que constituyen el cerebro animal. Ayudan a la IA a aprender a partir de ejemplos.

- **Árboles de decisión:**Una herramienta de apoyo a la toma de decisiones que utiliza un modelo de árbol de decisiones y sus posibles consecuencias. Representa la lógica de...
"Si esto sucede, entonces haz aquello."

- **Aprendizaje automático (ML):**Un subconjunto de IA que proporciona a los sistemas la capacidad aprender y mejorar automáticamente a partir de la experiencia sin ser programado explícitamente.

Este módulo utiliza un enfoque narrativo (narrativa) para hacer que estos conceptos matemáticos complejos sean accesibles y fáciles de entender.

UNIDAD 1.3 CONFIGURACIÓN DEL SISTEMA Y REQUISITOS PREVIOS

¿Qué plataforma necesitas?

En este módulo, utilizarás **Google Colab** para ejecutar el código Python. Google Colab es una plataforma en línea gratuita que te permite escribir y ejecutar código directamente en tu navegador web, sin necesidad de instalación.

Accede aquí:
https://colab.research.google.com/

¿Qué incluye el proceso de registro?

1. Ir a Google Colab
2. Haga clic en "Iniciar sesión" en la esquina superior derecha
3. Ingresa tu correo electrónico y contraseña de Google.
4. Una vez que haya iniciado sesión, podrá abrir o crear un nuevo cuaderno.

¿Necesitas registrarte?

Sí, necesitas una cuenta **gratuita de Google** (una cuenta de Gmail) para usar Google Colab. Si ya tienes una, puedes iniciar sesión directamente. Si no tienes una cuenta de Google, debes crear una antes de empezar.

Para crear una cuenta de Google:
Ir a cuentas.google.comy siga los pasos de registro.

Dónde se espera que escribas el código

Todos los ejercicios de codificación del Módulo 1 se preparan en un **cuaderno de Google Colab.**
No es necesario que escribas código: utilizarás un cuaderno prediseñado que contiene todos los ejercicios e instrucciones.

Aquí está el cuaderno del Módulo 1:
Módulo 1 Cuaderno de Colab

¿Qué hacer con el cuaderno?

1. Abra el enlace de arriba.
2. Haga clic en "Copiar a Drive" para guardar su propia versión editable.
3. Siga las instrucciones dentro del cuaderno, ejecutándose
   Cada celda de código paso a paso

UNIDAD 1.4 INSTRUCCIONES ADICIONALES ANTES DE COMENZAR

1. Utilice un navegador web moderno como Chrome, Firefox, Edge o Safari.
2. Asegúrate de tener una conexión a Internet estable: Google Colab funciona en línea.
3. Lea atentamente la historia y las instrucciones de cada estación antes de... Comenzar la tarea de codificación.
4. Cada estación ofrece **tres niveles de dificultad:**
   - **Básico** es para principiantes.
   - **Avanzado** es si tienes alguna experiencia con Python
   - **Experto:**Si buscas desafíos realmente difíciles, Expert es el lugar ideal.
   - Elige el nivel que se adapte a tus habilidades
5. Ejecute las celdas de código en orden. Algunos ejercicios dependen de los anteriores.
6. Si está atascado, utilice la "**Pista**" y el "**Fragmento de solución**" proporcionados en cada ejercicio.
7. Tras completar una tarea, intenta cambiar el código para ver cómo afecta al resultado. Experimentar te ayuda a aprender.

REFERENCIAS

Atribución de emojis e iconos

- Este documento utiliza emojis Unicode estándar, según la definición del Consorcio Unicode. Las versiones específicas de los emojis pueden variar según el sistema operativo del usuario. Consorcio Unicode. (2022). Recuperado de https://unicode.org/emoji/charts/full-emoji-list.html

Libros y artículos

- Russell, S., & Norvig, P. (2021). Inteligencia Artificial: Un Enfoque Moderno (4.ª ed.). Pearson. (Un libro de texto completo sobre conceptos de IA, incluyendo redes neuronales, árboles de decisión y aprendizaje por refuerzo).

- Géron, A. (2022). Aprendizaje automático práctico con Scikit-Learn, Keras y TensorFlow (3.ª ed.). O'Reilly. (Introducción práctica al aprendizaje automático y las redes neuronales con Python).

Ética y sociedad de la IA

- European Commission. (2019). Directrices éticas para una IA fiable. (Un marco para el desarrollo y la implementación de una IA ética)

- UNESCO. (2021). Recomendación sobre la ética de la inteligencia artificial.
- (Directrices globales sobre la ética de la IA, centradas en los derechos humanos y la inclusión).

Programación en Python para principiantes

- Sweigart, A. (2019). Automatiza las tareas aburridas con Python (2.ª ed.). No Starch Press. (Libro de Python para principiantes con proyectos prácticos).

ENLACES A PLATAFORMAS DE APRENDIZAJE

- Aquí está el cuaderno del Módulo 1 – https://colab.research.google.com/drive/14VaJnlgo5habZ0-N9jpz9nWZSVAw_Xwa?usp=sharing

- Documentación oficial de Python – https://docs.python.org/3/ (Referencia oficial del lenguaje Python y tutoriales

EJERCICIO PRÁCTICO

Estación 1: El regalo misterioso: un reloj que hace más que marcar

Es el 17.º cumpleaños de Aylin. Entre baklava y globos hay un pequeño y misterioso paquete. Aylin duda en abrirlo.

AYLIN: *"¿Qué pasa, tía Sema?*

AUNT SEMA: *Algo especial, canım (cariño). Tu abuela diría: «Este reloj tiene alma».*

Aylin abre el paquete con cuidado. Dentro hay un elegante reloj inteligente.
Con correa azul turquesa. De repente, la pantalla se ilumina.

ALARA (the watch): *"¡Merhaba (Hola), Aylin! ¡Soy Alara, tu nueva amiga IA!"*

Aylin salta hacia atrás sorprendida.

AYLIN: *"¿El... el reloj está hablando?"*

AUNT SEMA: *"Habla todos los idiomas que entiendes. ¡Alara te mostrará cómo funciona realmente la inteligencia artificial!"*

ALARA: *"magina que soy como tu cerebro: tengo neuronas que aprenden y sinapsis que conectan. ¡Juntos descubriremos cómo las máquinas aprenden a pensar!"*

```
BASICO

CÓDIGO:
NOMBRE = INPUT("¿CÓMO TE LLAMAS? ")
PRINT("HOLA " + NOMBRE + "! SOY ALARA, TU AMIGA IA")
```

Sugerencia: utilice la variable que ingresó en el espacio en blanco.

Fragmento de solución: nombre

Estación 1: El regalo misterioso: un reloj que hace más que marcar

```
AVANZADO

CÓDIGO:
print("🎂 Planificación del cumpleaños de Aylin 🎂")

comida_favorita = input("¿Cuál es tu comida favorita?
(Pizza/Kebab/Hamburguesa): ")
bebida = input("¿Y qué te gustaría beber con eso? ")

if comida_favorita.lower() == "pizza":
    print("🍕 Alara dice: ¡La pizza también es mi favorita!")
else:
    print("😋 Alara dice: " + comida_favorita + " ¡suena
genial!")

print("¿Y " + bebida + " con eso? ¡Combinación perfecta!")
```

Sugerencia: Complete los espacios en blanco con sus variables de entrada.

Fragmento de solución: favorita_comida, bebida

ESTACIÓN 2: Redes neuronales: Alara despierta

AYLIN:"Entonces... ¿piensas como yo?"

ALARA:¡Casi! Imagina que mi cerebro es como una gran fábrica de pensamientos. Tengo neuronas que aprenden. Mira...

AYLIN:"Entonces, ¿vas a aprender conmigo?"

ALARA:"Sí", responde Alara.
"Sé lo básico, pero a través de ti aprendo a ver el mundo con ojos humanos".

ESTACIÓN 2: Redes neuronales: Alara despierta

```
BASIC

CÓDIGO:
print("¿Debería comprar un videojuego nuevo?")
print("Valora de 0 a 1, donde 0 = nada, 0.5 = regular, 1 =
muchísimo")

saldo_ahorros = float(input("¿Qué tan llena está tu hucha?
(0-1): "))
deseo = float(input("¿Cuánto quieres el videojuego? (0-1):
"))

def neurona_decision(dinero, ganas):
    if dinero * ganas > 0.5:
        return "✅ SÍ, ¡cómpralo!"
    else:
        return "❌ NO, mejor espera"

print(neurona_decision(saldo_ahorros, deseo))
```

Fragmento de solución: 0.5

Sugerencia: Una neurona se "activa" cuando el producto es mayor que 0,5.

ESTACIÓN 2: Redes neuronales: Alara despierta

AVANZADO (Parte I)

CÓDIGO:

```
print("¿Es una buena amistad?")
print("Valora de 0 a 1, donde 0 = nada, 0.5 = regular, 1 =
muchísimo")

confianza = float(input("¿Cuánto confías en esta persona? (0-
1): "))
diversion = float(input("¿Cuánta diversión tenéis juntos? (0-
1): "))
ayuda = float(input("¿Qué tan servicial es esta persona? (0-
1): "))

# Pesos - ¿qué es lo más importante en una amistad?
peso_confianza = 0.5
peso_diversion = 0.3
peso_ayuda = 0.2

def analisis_amistad(confianza, diversion, ayuda):
    entradas = [confianza, diversion, ayuda]
    pesos = [peso_confianza, peso_diversion, peso_ayuda]

    total = 0
    for i in range(len(entradas)):
        total += entradas[i] * ____
    return total
```

ESTACIÓN 2: Redes neuronales: Alara despierta

```
AVANZADO  (Parte II)

CÓDIGO:
puntuacion = analisis_amistad(confianza, diversion, ayuda)
print("Puntuación de la amistad:", puntuacion)

if puntuacion > 0.7:
    print("💖 ¡Amistad verdadera!")
elif puntuacion > 0.4:
    print("👍 Buena relación")
else:
    print("🤔 Quizás no es la mejor amistad")
```

Fragmento de solución: pesos[i]

Sugerencia: Multiplique cada entrada por su peso correspondiente.

ESTACIÓN 3: Árboles de decisión: el detective de la ruta escolar

Aylin está en la parada del autobús. Llueve. Habla con Alara sobre su camino a la escuela.

ALARA *¡Tengo una idea para tu ruta escolar! Hagamos un árbol de decisiones. Alara la ayuda a tomar decisiones: esto se llama árbol de decisiones.*

AYLIN: *—¡Genial! Ahora siempre sabré qué camino tomar —dice Aylin con alegría.*

ALARA:"*Y estoy aprendiendo cómo tomar decisiones*".

ESTACIÓN 3: Árboles de decisión: el detective de la ruta escolar

BASIC

CÓDIGO:

```
clima = input("¿Qué tiempo hace? (sol/lluvia): ")

def ruta_al_colegio(clima):
    if clima == "lluvia":
        return "☔ ¡Llévate el paraguas!"
    else:
        return "🌞 ____"

print(ruta_al_colegio(clima))
```

Consejo: ¿Qué haces si no llueve?

Fragmento de solución: ¡Vaya sin paraguas!

AVANZADO

CODE:

```
clima = input("¿Qué tiempo hace? (lluvia/sol): ")
bus_llega = input("¿Viene el autobús? (sí/no): ")

def plan_ruta(clima, bus_llega):
    if clima == "lluvia" and bus_llega == "sí":
        return "🚌 ¡Coge el autobús!"
    elif clima == "lluvia" and bus_llega == "no":
        return "☂️ ____"
    else:
        return "🚶 ¡Ve caminando!"

print(plan_ruta(clima, bus_llega))
```

Consejo: Si el autobús no llega, sólo el paraguas ayuda.

Fragmento de solución: ¡Toma el paraguas!

ESTACIÓN 4: Clasificación basada en reglas: cómo Spotify podría saber lo que te gusta

Aylin y su amiga Lena están escuchando música.

LENA:*"¿Cómo sabe tu reloj que me gusta el rock?"*

ALARA:*¡Es una clasificación basada en reglas! Agrupo las canciones similares.*
LENA:"¡Eso es como magia!" exclama Lena.

ALARA:*—No hay magia —la corrige Alara.*
"Solo reconocimiento de patrones, y cada vez aprendo más sobre tus gustos musicales".

```
BASICO

CÓDIGO:
cancion = input("Describe tu canción: ... (por ejemplo:
guitarra)")

def clasificar_musica(cancion):
    if "guitarra" in cancion:
        return "🎸 Rock"
    elif "____" in cancion:
        return "🎤 Pop"
    else:
        return "🎶 Otro"

print(clasificar_musica(cancion))
```

Fragmento de solución: bailable

Consejo: Las canciones bailables suelen ser pop.

Agrupación simplificada: En este ejercicio, agrupamos a los oyentes según sus preferencias musicales más marcadas. Los sistemas de IA reales, como K-Means, utilizan cálculos matemáticos más complejos para encontrar automáticamente usuarios con gustos similares.

ESTACIÓN 5: Aprendizaje por refuerzo: aprendizaje mediante recompensas

Aylin está jugando con una aplicación de entrenamiento de perros en su tableta

AYLIN:*"¿Cómo le enseño a este perro virtual qué hacer?"*

ALARA:*Es sencillo: ¡mediante recompensas! Cuando haga algo bien, dale un premio. Así, poco a poco, aprenderá qué acciones vale la pena realizar. ¡A esto se le llama aprendizaje por refuerzo!*

AYLIN:*"Entonces, ¿es como la escuela? ¿Buenas notas por las respuestas correctas?"*

ALARA:*¡Exactamente! Pero aquí tú decides qué es lo correcto. El perro prueba cosas y recuerda lo que funciona.*

AYLIN:*"¿Entonces aprendes igual que tu perro virtual?", pregunta Aylin con una sonrisa.*

ALARA:*"En cierto modo, sí", admite Alara. "Las recompensas funcionan para todos los estudiantes".*

BASICO (Parte I)

CÓDIGO:

```
import random

print("¡Entrena a tu perro virtual!")
print("Recompensa con una golosina (1) o ignora (0)\n")

acciones = ["Sentado", "Túmbate", "Da la pata"]
probabilidades = [0.3, 0.3, 0.4]  # Valores iniciales
probabilidad_distraccion = 0.2  # 20% de probabilidad de que
el perro haga otra cosa

def normalizar(probabilidades):
    total = sum(probabilidades)
    return [p / total for p in probabilidades]
```

ESTACIÓN 5: Aprendizaje por refuerzo: aprendizaje mediante recompensas

BASICO (Parte II)

CÓDIGO:

```
for ronda in range(1, 6):
   print(f"\n Sesión de entrenamiento {ronda}/5")

   # El perro elige una acción según sus preferencias
   accion = random.choices(acciones,
weights=_______________)[0]  # Hueco 1

   # Distracción: el perro podría hacer una acción diferente
   if random.random() < probabilidad_distraccion:
      acciones_posibles = [a for a in acciones if a !=
accion]
      accion_equivocada = random.choice(acciones_posibles)
      print(f"El perro debería hacer '{accion}' pero en su
lugar hace: 🐶 {accion_equivocada}")
      accion = accion_equivocada
   else:
      print(f"El perro hace: 🐾 {accion}")

   # Interacción con el usuario con verificación del input
   while True:
      entrada_usuario = input("¿Dar una golosina? (1=sí,
0=no): ").strip()
   if entrada_usuario in ("1", "0"):
      recompensa = int(entrada_usuario)
      break
      print("¡Por favor, introduce solo 1 o 0!")

   indice = acciones.index(accion)
```

ESTACIÓN 5: Aprendizaje por refuerzo: aprendizaje mediante recompensas

```
BASIC (Part III)

CÓDIGO:
      # Lógica de aprendizaje
   if reward == 1:
      probabilities[index] += ______  # Hueco 2
      print("✅ El perro está feliz: '¡Eso estuvo bien!'")
   else:
      probabilities[index] = max(0.1, probabilities[index] -
0.1)
      print("❌ El perro piensa: 'Eso no fue tan bueno...'")

      # Normalizar probabilidades
      probabilities = normalize(probabilities)

      print("Valores actuales de aprendizaje:", [round(p, 2)
for p in probabilities])

# Mostrar resultados
print("\n ¡Entrenamiento finalizado!")
for action, value in zip(actions, probabilities):
   print(f" {action}: {value:.2f}")

accion_favorita =
actions[probabilities.index(max(probabilities))]
print(f"\n🐶 Acción favorita del perro: {accion_favorita}
🦴")
```

Consejo 1: Aquí se almacenan las probabilidades actuales, que el perro utiliza para elegir una acción.

Consejo 2: ¿Cuánto debería aumentar la probabilidad cuando el perro recibe una recompensa?

Fragmento de solución 1: probabilidades

Fragmento de solución 2: 0,2

ESTACIÓN 5: Aprendizaje por refuerzo: aprendizaje mediante recompensas

AVANZADO (Parte I)

CÓDIGO:

```
print("¡Mantén tu planta viva!")
print("Acciones: regar, fertilizar, no_hacer_nada")

estado_planta = {
   "water": 5,  # 0-10
   "nutrients": 5,
   "health": 10
}

def evaluar_estado():
    water, nutrients = estado_planta["water"],
estado_planta["nutrients"]
    if 3 <= water <= 7 and 3 <= nutrients <= 7:
        return ____  # ¡Perfecto!
    elif 1 <= water <= 9 and 1 <= nutrients <= 9:
        return _____  # Bien
    else:
        return -1  # Mal

for dia in range(7):
    print(f"\n--- Día {dia+1} ---")
    print(f"Estado: Agua={estado_planta['water']}/10,
Nutrientes={estado_planta['nutrients']}/10")

    accion = input("¿Qué haces? (regar/fertilizar/nada): ")
```

ESTACIÓN 5: Aprendizaje por refuerzo: aprendizaje mediante recompensas

AVANZADO (Parte II)

CÓDIGO:

```
        # Realizar acción
    if accion == "water":
        estado_planta["water"] = min(10,
estado_planta["water"] + 3)
    elif accion == "fertilize":
        estado_planta["nutrients"] = min(10,
estado_planta["nutrients"] + 3)

    # El tiempo pasa - el estado disminuye
    estado_planta["water"] = max(0, estado_planta["water"] -
1)
    estado_planta["nutrients"] = max(0,
estado_planta["nutrients"] - 1)

    # Calcular recompensa
    recompensa = evaluar_estado()
    print(f"Recompensa: {recompensa} puntos")

    if estado_planta["water"] == 0 or
estado_planta["nutrients"] == 0:
        print("💀 ¡La planta ha muerto!")
        break

if estado_planta["water"] > 0 and estado_planta["nutrients"]
> 0:
    print("❤️🌱 ¡La planta sobrevivió! ¡Bien hecho!")
```

Consejo 1: Puntos por niveles ideales de agua y nutrientes (máxima recompensa).

Consejo 2: Puntos por niveles aceptables, no perfectos pero aún así correctos.

Fragmento de la solución 1: 2

Fragmento de la solución:2 1

ESTACIÓN 6: Redes neuronales convolucionales: cómo una IA detecta bordes en imágenes

Aylin mira su foto.

ALARA:“Veo líneas, brillo ycontrastes—estos¡Son mis bordes! Cuando algo cambia, lo sé: un nuevo objeto comienza aquí.

ALARA:Para detectar esto, uso un pequeño truco. Siempre comparo dos píxeles vecinos. Si son diferentes, aparece un borde. El operador != significa 'no igual' o 'diferente de'.

ALARA:“Cada imagen que me muestras me ayuda a ‘ver’ mejor”, explica Alara.
“A medida que aprendes, describo el mundo cada vez con mayor precisión”.

BASICO (Parte I)

CÓDIGO:

```
def contar_bordes(fila):
    bordes = 0
    for i in range(len(fila)-1):
        if fila[i] ___ fila[i+1]:
            bordes += 1
        return bordes

print("Imagen de 4 filas de Alara")

# Introducir 4 filas por el usuario
imagen = []
for n in range(4):
    fila = input(f"Introduce la fila {n+1} (0 y 1, por
ejemplo: 100101): ")
    fila = fila[:6]
    imagen.append(fila)
```

ESTACIÓN 6: Redes neuronales convolucionales: cómo una IA detecta bordes en imágenes

```
BASICO (Parte II)

CODE:
# Contar bordes por fila
for i, fila in enumerate(imagen):
    bordes = contar_bordes(fila)
    if bordes > 0:
        print(f" Fila {i+1}: {bordes} bordes")
    else:
        print(f"⬜ Fila {i+1}: sin bordes")
```

Sugerencia: Un cambio (0 → 1 o 1 → 0) es una ventaja.
Fragmento de solución: i=

```
AVANZADO  (Parte I)

CÓDIGO:
print("¡Crea tu propia imagen 4x4!")
size = 4
imagen = []

# Introducir filas
for i in range(size):
    fila = input(f"Introduce la fila {i+1} (solo ⬜ y ⬛,
por ejemplo: ⬛⬜⬜⬛): ")
    fila = list(fila.ljust(size, "⬜"))[:size]
    imagen.append(fila)

print("\n Tu imagen:")
for fila in imagen:
    print(" ".join(fila))
```

ESTACIÓN 7: Procesamiento del lenguaje natural basado en reglas: cómo las computadoras entienden el lenguaje

Aylin le escribe un mensaje a Lena: “¡Oye, eso estuvo genial!”

ALARA:*“Puedo saber si estás feliz, enojado o neutral. ¡Eso se llama análisis de sentimientos!”*

ALARA:*"Tus mensajes me ayudan a comprender mejor el lenguaje humano", dice Alara agradecida. "Estamos aprendiendo juntos: tú me enseñas IA y yo aprendo comunicación humana".*

```
BASICO

CÓDIGO:
print("¡Alara interpreta tu estado de ánimo!")
texto = input("Escribe algo (por ejemplo: … eso fue genial /
… eso fue malo): ")

def entender(texto):
if "genial" in texto:
   return "😊 ¡Estás de buen humor!"
elif "malo" in texto:
   return ____
else:
   return "😐 Neutral."

print(entender(texto))
```

Consejo : “malo” = negativo.

Fragmento de solución : "😠 ¡Estás enojado!
"

ESTACIÓN 7: Procesamiento del lenguaje natural basado en reglas: cómo las computadoras entienden el lenguaje

AVANZADO

CÓDIGO:

```
print("¡Contando palabras como un modelo de lenguaje!")
texto = input("Escribe una frase: ")

def contar_palabras(texto):
    palabras = texto.split()
    return len(____)

print("Número de palabras:", contar_palabras(texto))
```

Consejo : ¿Qué contiene todas las palabras?
Fragmento de solución: palabras

ESTACIÓN 8: Sistemas de recomendación —Cómo YouTube podría saberlo

Aylin mira videos en su tableta.

ALARA:*"¡Recuerdo lo que te gusta y te sugiero contenido similar!"*

ALARA:"Cuanto más me muestres lo que te gusta, mejores sugerencias podré ofrecerte".
Alara explica: «Esto es una colaboración».

```
BASIC

CÓDIGO:
print("Recomendación de vídeos de Alara")
video = input("¿Qué estás viendo? (por ejemplo: vídeo
musical, vídeo de animales): ")

def recomendar(video):
    if "música" in video:
        return "🎶 ¡Vídeos musicales similares!"
    else:
        return ____

print(recomendar(video))
```

Consejo: si no es un vídeo musical, sugiere algo nuevo.
Fragmento de solución: "¡Prueba algo nuevo!"

ESTACIÓN 8: Sistemas de recomendación —Cómo YouTube podría saberlo

AVANZADO

CÓDIGO:

```
print("TU ASISTENTE DE YOUTUBE")
print("¡Encuentra tu recomendación perfecta!")

nombre = input("\n¿Cómo te llamas? ")
print(f"¡Hola {nombre}! ¿Qué te gusta?")
favorito1 = input("Tema favorito 1: ")
favorito2 = input("Tema favorito 2: ")

datos_usuario = {
   nombre: [favorito1, favorito2],
   "Lena": ["Deportes", "Gaming"]
}

def recomendacion(nombre_usuario):
   if favorito1 in ["Música", "música"]:
       return "🎵 TU RECOMENDACIÓN: ¡Mix Top 100 Charts!"
   elif favorito1 in ["Deportes", "deportes"]:
       return "⚽ TU RECOMENDACIÓN: ¡Mejores goles
recopilación!"
   else:
       return "🎬 TU RECOMENDACIÓN: ¡Vídeos en tendencia!"

print(f"\n{recomendacion(____)}")
```

Consejo: ¿Qué variable contiene el nombre?

Fragmento de solución: nombre

ESTACIÓN 9: Pronóstico del tiempo—Bosque aleatorio

AAylin señala su reloj.

AYLIN:"¿Cómo sabes siempre si podemos jugar afuera?"

ALARA:¡Ese es mi equipo de expertos! Tres mentes inteligentes dentro de mí se consultan entre sí, ¡como si fueran un bosque al azar!

A**YLIN:¡**Guau! ¿Así que de verdad tienes tres expertos diferentes?

ALARA:¡Sí! Uno mira la temperatura, otro las nubes y otro el viento. ¡Juntos tomamos la mejor decisión!

ALARA:"Con cada predicción, soy más preciso"Alara dice: "Sus comentarios me ayudan a perfeccionar mis expertos".

¿Qué tienen de especial los bosques aleatorios?En realidad, un Bosque Aleatorio entrena muchos árboles utilizando datos y características seleccionados aleatoriamente. Nuestra votación simula este principio de "inteligencia colectiva"

ESTACIÓN 9: Pronóstico del tiempo—Bosque aleatorio

```
BASICO

CÓDIGO:

print("LOS EXPERTOS DEL TIEMPO DE AYLIN")
print("¡Tres expertos están consultando!")

temperatura = float(input("Temperatura: "))
nubes = input("Nubes (soleado/nublado): ")

# Preguntamos a los expertos
experto1 = "Sol" if temperatura > 20 else "Lluvia"
experto2 = "Sol" if nubes == "soleado" else "Lluvia"

opiniones = [experto1, experto2]
decision = max(set(opiniones), key=opiniones.____)

print(f"Pronóstico: {decision}")
```

Consejo: Cuente qué opinión aparece con mayor frecuencia.
Fragmento de solución: count

ESTACIÓN 9: Pronóstico del tiempo—Bosque aleatorio

AVANZADO

CÓDIGO:

```
print("COMITÉ METEOROLÓGICO DE AYLIN")
print("¡Tres expertos en IA deciden!")

temperatura = float(input("Temperatura: "))
nubes = input("Nubes (soleado/nublado/lluvioso): ")
viento = input("Viento (fuerte/débil): ")

# Tres expertos dan su opinión
experto1 = "Sol" if temperatura > 20 else "Lluvia"
experto2 = "Sol" if ____ == "soleado" else "Lluvia"
experto3 = "Sol" if viento == "débil" else "Lluvia"

opiniones = [experto1, experto2, experto3]
decision = max(set(opiniones), key=opiniones.count)

print(f"Pronóstico: {decision}")
```

Sugerencia: ¿Qué variable contiene la información de la nube?

Fragmento de solución: nubes

Epílogo—El segundo regalo

Unos meses después.

Aylin está sentada en su escritorio, con Alara brillando en su muñeca. En la cocina, los platos tintinean: la tía Sema está de visita y preparando börek.

AYLIN:"¿Sabes, Alara?", dice Aylin con una sonrisa, "cuando me la regalaron para mi cumpleaños, no tenía ni idea de IA. Ahora... estoy desarrollando mi primer proyecto y entiendo cómo piensas".

Alara parpadea con orgullo. «No solo has entendido la IA, sino que también me has enseñado a pensar más como un humano».

En ese momento, la tía Sema entra con un pequeño regalo envuelto.
TÍA SEMA:"Canım" (cariño), dice con dulzura, "He visto cuánto has aprendido en los últimos meses. Tu abuela diría: "Le has dado alma al reloj".

Aylin abre el paquete. Dentro hay un cuaderno sencillo. En la primera página, escrito con letra fluida:

“Inventa con corazón. Programa con valentía. Piensa como Alara, pero siente como Aylin”.
Aylin sonríe.

AYLIN:“Este es el regalo más hermoso, tía Sema”

TÍA SEMA:*—No —dice Sema, dándole un ligero golpecito en el pecho—. El regalo más hermoso ya está aquí.*

FIN

PREGUNTAS DE REFLEXIÓN

Tómate un momento para reflexionar sobre lo aprendido en este módulo. Responde las siguientes preguntas con atención.

- **Entendiendo la IA:**¿Cómo ha cambiado tu comprensión de la inteligencia artificial tras completar este módulo? ¿Qué te sorprendió más de cómo piensan los sistemas de IA como Alara?

- **Redes neuronales y toma de decisiones:**En la Estación 2, exploraste cómo las redes neuronales usan ponderaciones para tomar decisiones. ¿Te imaginas una situación real en la que sopesas diferentes factores antes de tomar una decisión? ¿En qué se parece esto a **cómo la IA toma decisiones?**

- **Consideraciones éticas:** A lo largo del módulo, se plantearon cuestiones éticas sobre la IA. ¿Por qué es importante considerar la ética al diseñar o utilizar la IA? Dé un ejemplo de una preocupación ética relacionada con la IA en la vida diaria

- **Aprendizaje a través de la interacción:** Alara aprende de las interacciones de Aylin. ¿Cómo crees que los sistemas de IA en la vida real (como los motores de recomendación o los asistentes de voz) aprenden de las interacciones de los usuarios? ¿Cuáles son los beneficios y los riesgos de este tipo de aprendizaje?

- **La IA en tu mundo**: ¿Dónde te encuentras con la IA en tu vida diaria? Elige un ejemplo (p. ej., redes sociales, servicios de streaming, dispositivos inteligentes) y explica cómo utiliza uno de los conceptos de IA que aprendiste en este módulo.

CUESTIONARIO CON PREGUNTAS DE OPCIÓN MÚLTIPLE (una respuesta correcta por pregunta)

1. ¿Qué es Alara?
a) La mejor amiga de Aylin
b) Un despertador normal y corriente
c) Una IA en un reloj inteligente
d) La tía de Aylin

2. ¿Qué concepto de programación se introduce en la Estación 2 para tomar una decisión basada en múltiples entradas ponderadas?
a) Árboles de decisión
b) Aprendizaje por refuerzo
c) Redes neuronales
d) Clasificación basada en reglas

3. La estación 3, "El detective de la ruta escolar", trata principalmente sobre...
a) Recomendaciones musicales
b) Planificación de rutas mediante árboles de decisión
c) Reconocimiento de imágenes
d) Procesamiento del lenguaje

4.En la Estación 4, ¿cómo clasifica Alara una canción como "Pop"?
a) Si contiene una guitarra
b) Si es ruidoso
c) Si es bailable
d) Si es de Taylor Swift

5. ¿Qué simula el ejercicio “Cuidado de plantas” en la Estación 5?
a) Reconocimiento de imágenes
b) Aprendizaje por refuerzo mediante recompensas
c) Redes neuronales
d) Árboles de decisión

CUESTIONARIO CON PREGUNTAS DE OPCIÓN MÚLTIPLE (una respuesta correcta por pregunta)

6.En la Estación 6, ¿cómo detecta Alara los bordes de una imagen?
a) Comparando píxeles vecinos para detectar diferencias
b) Contando los píxeles blancos
c) Mediante comandos de voz
d) Preguntando al usuario

7. En la Estación 7 ("Procesamiento del Lenguaje Natural"), el estado de ánimo de un texto se reconoce mediante...
a) Medir la longitud del texto
b) Buscar palabras clave específicas como "excelente" o "malo"
c) Comprobación de la gramática
d) Análisis de la fuente

8. ¿Qué hace el sistema de recomendaciones en Station 8?
a) Crea pronósticos meteorológicos
b) Sugiere vídeos según los intereses del usuario.
c) Reconoce la escritura a mano.
d) Guía a un robot a través de un laberinto.

9. ¿Cómo toma una decisión el "Bosque Aleatorio" en la Estación 9?
a) Sólo un experto decide
b) A través de una votación democrática de múltiples “árboles”
c) Mediante selección aleatoria
d) A través de redes neuronales profundas

10. ¿Qué lenguaje de programación se utiliza en todos los desafíos de codificación?
a) JavaScript
b) Python
c) N1GL1B
d) C++

MÓDULO N.° 2 | Búsqueda del tesoro con calculadora en una sala de escape con IA

INTRODUCCIÓN

¿Alguna vez te has preguntado cómo tu teléfono inteligente sabe qué videos te podrían gustar? El objetivo de este módulo es presentar los conceptos fundamentales que explican cómo la Inteligencia Artificial (IA) "piensa", toma decisiones y aprende de los datos. El contenido proporciona una base teórica sencilla que conecta las matemáticas, la lógica y la creatividad mediante una experiencia de aprendizaje atractiva y lúdica.

Al finalizar el módulo podrás adquirir diferentes habilidades, tales como:

- **Competencia matemática:** Mayor dominio de las operaciones aritméticas básicas, comprensión de promedios, porcentajes y razones. Mayor confianza en el uso de calculadoras para la resolución de problemas.
- **Pensamiento crítico y razonamiento lógico:** Los estudiantes practicarán el pensamiento lógico al decodificar pistas y tomar decisiones basadas en los datos proporcionados. Las habilidades de resolución de problemas se fortalecen al aplicar las matemáticas a situaciones prácticas y acertijos.
- **Toma de decisiones:** Al resolver tareas relacionadas con la clasificación (por ejemplo, decidir qué objeto se ajusta a una determinada descripción), los estudiantes practican la toma de decisiones informadas basadas en datos numéricos y lógica.

Duración del módulo	2 horas(1 hora de aprendizaje + 1 hora de ejercicios prácticos)

MATERIALES EDUCATIVOS

UNIDAD 2.1 ¿QUÉ HAY DETRÁS DE LA SALA DE ESCAPE CON IA?

La inteligencia artificial puede definirse como la capacidad de las máquinas para realizar tareas que normalmente requieren inteligencia humana, como razonar, resolver problemas, aprender o reconocer patrones. Las matemáticas son la base de la IA: la probabilidad, la estadística y la lógica ayudan a las computadoras a tomar decisiones basadas en datos, no en la intuición. Los datos, a su vez, actúan como el "combustible" de la IA: cuantos más datos de calidad haya disponibles, mejor será el sistema para detectar patrones y hacer predicciones.
En este módulo, descubrirás cómo la IA utiliza el razonamiento, la probabilidad y el aprendizaje para resolver problemas reales de forma creativa.

Para comprender estas ideas, exploraremos cuatro principios clave que representan cómo “piensa” la IA:

- **Redes bayesianas,** que ayudan a predecir resultados cuando la información es incierta.
- **Agrupamiento de K-Medias,** que agrupa datos similares para encontrar patrones y relaciones.
- **Métodos de Monte Carlo,** que utilizan experimentos aleatorios para estimar probabilidades y hacer predicciones.
- **Aprendizaje hebbiano,** lo que muestra cómo las conexiones se fortalecen a través de la repetición, al igual que nuestro cerebro cuando aprende cosas nuevas.

UNIDAD 2.2 COMPRENDER LA IA A TRAVÉS DE IDEAS SIMPLES

Con estas sencillas ideas, comprenderás que la IA no se trata solo de programación, sino también de razonamiento lógico, resolución de problemas y creatividad. Estos principios te guiarán en cada etapa de la Sala de Escape con IA, donde cada desafío revelará cómo la inteligencia artificial "aprende haciendo".

Estas cuatro ideas muestran cómo la IA combina la lógica, las matemáticas y la creatividad para aprender y tomar decisiones inteligentes: ¡exactamente lo que harás mientras resuelves los acertijos en esta aventura de AI Escape Room!

"El propósito de la IA no es reemplazar a los humanos, sino amplificar las capacidades humanas".
— *Fei-Fei Li (Universidad de Stanford, investigadora de IA)*

REFERENCIAS

Inteligencia artificial y toma de decisiones

- Russell, S., & Norvig, P. (2021). Inteligencia artificial: un enfoque moderno 4ª ed.). Pearson.

Probabilidad, Simulación y Aprendizaje

- Shalev-Shwartz, S., & Ben-David, S. (2014). Comprensión del aprendizaje automático: De la teoría a los algoritmos. Cambridge University Press.→ Fundamentos del aprendizaje automático, la agrupación en clústeres y el aprendizaje a partir de datos.

- Khan Academy. (n.d.). Estadística y probabilidad.. https://www.khanacademy.org/math/statistics-probability
  → Explicaciones sencillas y accesibles de probabilidad, promedios y simulaciones.

.

ENLACES A PLATAFORMAS DE APRENDIZAJE

Si quieres seguir aprendiendo sobre Inteligencia Artificial de forma divertida y sencilla, ¡estas plataformas en línea son un excelente punto de partida! Puedes jugar, experimentar y explorar cómo las computadoras aprenden a reconocer imágenes, sonidos o patrones, como en la Sala de Escape de IA. Cada enlace incluye actividades cortas que te ayudan a comprender cómo funciona la IA en la vida real mientras sigues aprendiendo con la práctica.

- Máquina de enseñanza de Google– Plataforma interactiva para entrenar modelos de IA simples con imágenes, sonidos o poses.
  https://teachablemachine.withgoogle.com/

- Khan Academy – Inteligencia artificial y aprendizaje automático– Lecciones gratuitas que explican conceptos básicos de IA utilizando ejemplos cotidianos.
  https://www.khanacademy.org/computing/computer-science

- IA para los océanos (Code.org)– Actividad interactiva divertida en la que los estudiantes entrenan una IA para limpiar el océano y aprenden sobre datos y sesgos
  https://studio.code.org/s/oceans

- Aprendizaje automático para niños– Plataforma que permite a los estudiantes experimentar con modelos de IA y usarlos en proyectos de Scratch
  https://machinelearningforkids.co.uk/

- IBM SkillsBuild para estudiantes– Plataforma de aprendizaje en línea gratuita que ofrece cursos básicos de inteligencia artificial y alfabetización de datos con insignias y certificados
  https://skillsbuild.org/students

- Experimentos de inteligencia artificial de Google– Una colección de herramientas interactivas que muestran cómo la IA reconoce patrones, sonidos y movimientos
  https://experiments.withgoogle.com/collection/ai

EJERCICIO PRÁCTICO

Bienvenido a la sala de escape con inteligencia artificial!

Formas parte de un equipo de jóvenes exploradores que buscan un tesoro encerrado tras una cerradura digital de 4 dígitos. Para abrirlo, debes avanzar por un camino que atraviesa 4 salas protegidas por cuatro guardianes, cada uno representando un principio de la Inteligencia Artificial:

1. Probabilidad (Redes Bayesianas)
2. Agrupamiento (Agrupamiento K-Medias)
3. Simulación (métodos de Monte Carlo)
4. Aprendizaje hebbiano

Tu misión es superar los desafíos de cada estación, resolver sus problemas y obtener la combinación que abre la cerradura. Cada estación incluye una explicación, con breves ejercicios intermedios y un desafío final de programación con tres niveles de dificultad que te proporcionará uno de los dígitos necesarios para abrir la cerradura.

Estación 1 – Redes bayesianas

Una **red bayesiana** es un modelo matemático que utiliza la **probabilidad** para tomar decisiones o predicciones cuando no se conoce con certeza toda la información, es decir, cuando existe **incertidumbre.**

Mini-Desafío 1 Estación 1 — La cerradura simple

Imagina que quieres saber si puedes abrir un candado numérico en una sala de escape.

- **Evento A: Pistas→** Hay varias pistas ocultas. Cuantas más encuentres, más fácil será abrir la cerradura.
- **Evento B: Acertijo→** Hay un acertijo que proporciona los números de la combinación. Cuantas más pistas encuentres, más fácil será resolverlo.
- **Evento C: Bloqueo→** La posibilidad de resolver el rompecabezas e introducir los números en el candado para abrirlo depende de la cantidad de pistas que encuentres.

Mini-Desafío 1 Estación 1 — La cerradura simple

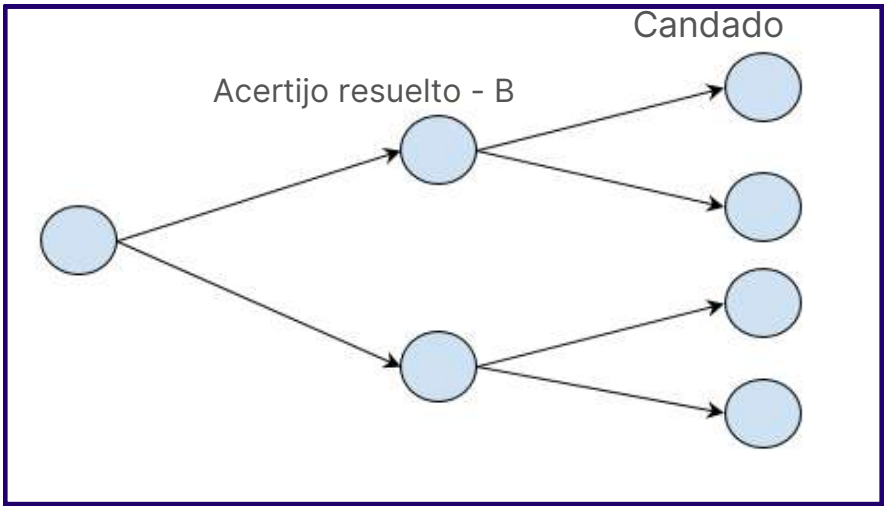


Figura 2. Ruta de decisión y probabilidad: desde encontrar pistas hasta abrir la cerradura.
Fuente:: Monica Moreno

Nota:la mitad del gráfico esta desaparecido,correspondiente a la probabilidad de no encontrar pistas.

I**nstrucciones:**
Si encuentras las pistas (probabilidad 0,8) y resuelves el acertijo (probabilidad 0,9 si encuentras las pistas), ¿cuál es la probabilidad de que abras el candado (probabilidad 0,95 si resuelves el acertijo)?

El camino para encontrar la solución se compone de una serie de probabilidades, que deben multiplicarse.

Respuesta: P(Cerradura abierta)=0,8 × 0,9 × 0,95

Pregunta¿Qué resultado obtendrías si sumas todos los caminos posibles?

Respuesta: 1, porque cada camino representa la probabilidad de que ocurra un posible escenario. La probabilidad de que ocurra uno de todos los resultados posibles siempre es 1.

Regla de Bayes

Hasta ahora hemos visto cómo podemos hacer una predicción, pero ¿qué pasa si miramos el resultado y queremos saber la probabilidad de que se haya seguido un camino?

Mini-Desafío 2 Estación 1 – El Corredor Condicional

Encuentras un pasillo secreto con otro acertijo en la pared: "De todas las veces que resolviste el acertijo, ¿cuál es la probabilidad de que hayas encontrado las pistas primero?"

En otras palabras, queremos calcular la probabilidad de C hacia B **(que es diferente de la probabilidad de B hacia C)**. Debes usar la regla de Bayes:
:
Instrucciones:
Primero debemos calcular la probabilidad de B=✓yC=✓, como todos los caminos que terminan enB=✓,y todos los que terminan en C=✓Para simplificar las cosas, te daremos las operaciones ya realizadas:P(B=✓)= 0,76, P(C=✓) = 0,734.

Ahora solo nos falta aplicar la regla de Bayes con la fórmula:

P(B=✓ if C=✓) = [P(C=✓ if B=✓) * P(B=✓)] / P(C=✓)

Respuesta: P(B=✓|C=✓) = [0.95 * 0.76] / 0.724 = 0.983

Desafío 1 - El guardián de la probabilidad

Has llegado a la primera sala, la Cámara del Guardián de la Probabilidad, un misterioso salón iluminado por linternas digitales. En el centro se encuentra un antiguo cofre cubierto de piedras preciosas. Entonces, el guardián, un holograma de un anciano, te dice:

*"Solo quienes comprenden la **incertidumbre** podrán descubrir si el cofre contiene oro. La probabilidad es la clave."*

*Tu misión es calcular la **probabilidad** de que el cofre contenga oro basándote únicamente en lo que puedes ver, utilizando un modelo de **red bayesiana simplificado.***

***Guión** ¿Cuál es la probabilidad de que este **cofre pesado y brillante** contenga oro? El primer decimal del resultado será el primer dígito de la cerradura.*

- Sabes que el **30% de los cofres brillantes contienen oro y el 70% contienen piedras.**
- Los cofres que contienen oro **son pesados en el 80% de los casos.**
- Los cofres que contienen piedras **son pesados en el 40% de los casos.**

Desafío 1 - El guardián de la probabilidad

NIVEL FÁCIL Dibuja la red bayesiana en una hoja de papel y usa la calculadora para obtener la respuesta. **Pista:** Solo necesitas dos eventos (A y B) y la regla de Bayes.

NIVEL MEDIO:Modifique este código incompleto para resolver el problema. Simplemente añada las probabilidades y los símbolos matemáticos para operar.

```
BASICO

CÓDIGO:
p_oro =
p_piedras =
p_pesado_si_oro =
p_pesado_si_piedras =

# Calcular la probabilidad de que sea pesado
p_pesado = (p_oro "operación" p_pesado_si_oro) + (p_piedras
"operación" p_pesado_si_piedras)

# Calcular la probabilidad de que sea oro dado que es pesado
p_oro_dado_pesado = (p_oro "operación" p_pesado_si_oro)
"operación" p_pesado

print("Probabilidad de que sea oro dado que es pesado:",
round(p_oro_dado_pesado, 2))
```

NIVEL DIFÍCIL: Dibuja la red bayesiana y crea tu propio código para resolver el problema.

Estación 2: Agrupamiento de K-Medias

K-medias Es un algoritmo que agrupa datos en categorías según su similitud. Para entender esto, veamos un ejemplo. Imagina que has medido la temperatura casi todos los días del año y quieres saber a qué estación corresponde cada punto. Esto se llama un problema de clasificación.

Mini-Desafío 1 Estación 2 – Los Cristales en el Pasillo

De camino a la siguiente habitación, encuentras varios cristales en el suelo. Cada cristal tiene dos números escritos: brillo y pureza. Agrupa estos cristales en dos grupos distintos según sus características.

Instrucciones

- Dibuja dos ejes (x,y) y marca los puntos:
  [1,5], [4,1], [3.5,0], [2,4], [0.5, 5], [0.5, 5], [5, 2].
- Imagina que quieres dos grupos o **clústeres** (k=2).
  ¿Dónde colocarías el centro inicial de cada grupo o **centroides**?
- Asigna cada cristal al centroide más cercano.

Respuesta : Grupo 1: [1,5], [2,4], [0.5, 5], [0.5, 5]. Group 2: [4,1], [3.5,0], [5, 2]

K-Means **automatiza el proceso de formación de grupos,** de modo que cada uno tenga un centro (centroide) que representa **el promedio de las características de ese grupo**. «Cómo funciona K-Means»

1. Elija cuántos clústeres hay (k=n)
2. Coloque k centroides aleatoriamente en el plano, uno para cada grupo
3. A cada punto del plano se le asigna el centroide más cercano
4. Calcula el “centro” de cada área para reubicar el centroide.
5. Repita el proceso hasta que los puntos ya no cambien de grupo.

Desafío 2 - El Alquimista

Una vez que hayas pasado la primera sala, llegarás a un laboratorio lleno de frascos con líquidos de colores misteriosos. Entonces, un a**nciano alquimista** te dice:

"Joven aventurero, no podrás irte de aquí sin clasificar mis pociones. Cada botella pertenece a un tipo de magia diferente: **Curación, Fuerza o Invisibilidad**. Si me ayudas a agruparlas correctamente, te abriré la puerta.

.

ECada frasco tiene **dos características químicas**: pH y concentración de energía mágica. Tu misión es usar **K-Means** para agrupar los **frascos** y así superar la segunda prueba. El número de grupos que crees corresponderá al segundo dígito del candado.

NIVEL FÁCIL Utilice el siguiente código para agrupar los archivos jar, completando los parámetros faltantes. **EXTRA**: observe cómo cambia el resultado al variar el número de grupos.

NOTA: Primero debes instalar la biblioteca sklearn (pip install scikit-learn)

NIVEL MEDIO Usa el código del nivel fácil y añade un nuevo punto con las coordenadas que desees. Usa la función kmeans.predict(point) para obtener la clasificación. Añade este código al final para mostrar el resultado.

NIVEL DIFÍCIL: Crea el código desde cero.

HARD LEVEL: Draw the Bayesian Network and create your own code to solve the problem.

Desafío 2 - El Alquimista

BASICO

CÓDIGO:

```
from sklearn.cluster import KMeans  # librería kmeans
import numpy as np
import matplotlib.pyplot as plt

# Datos: [pH, poder]
pociones = np.array([
   [2, 53], [6, 70], [3, 55], [3.1, 52], [10, 25],
   [8, 84], [9, 80], [7.5, 82],
   [10, 19], [10.5, 22], [4, 48], [9, 20],
])

# K-Means
kmeans = KMeans(n_clusters=2, random_state=0)  # Declaración
del algoritmo
kmeans.fit(pociones)  # Ejecuta kmeans para clasificar los
datos

# Visualización
plt.scatter(pociones[:, 0], pociones[:, 1], c=kmeans.labels_,
cmap="viridis", marker="o")
plt.xlabel("pH")
plt.ylabel("Poder mágico")
plt.title("Clasificación de pociones con K-Means")
plt.show()

print("Clasificación de cada frasco:", kmeans.labels_)
```

HARD LEVEL: Draw the Bayesian Network and create your own code to solve the problem.

Desafío 2 - El Alquimista

```
MEDIO

CÓDIGO:
plt.scatter(pociones[:, 0], pociones[:, 1], c=kmeans.labels_,
cmap="viridis", marker="o")
plt.scatter(
   nuevo_frasco[:, 0], nuevo_frasco[:, 1],
   c=[plt.cm.viridis(prediccion[0] / (kmeans.n_clusters -
1))],
   marker="x", s=150, label="nuevo frasco"
)
plt.xlabel("pH")
plt.ylabel("Poder mágico")
plt.title("Clasificación de botella especial")
plt.legend()
plt.show()
```

NIVEL DIFÍCIL: Dibuja la red bayesiana y crea tu propio código para resolver el problema.

Desafío 2 - El Alquimista

DURO

CÓDIGO:

```
from sklearn.cluster import KMeans
import numpy as np
import matplotlib.pyplot as plt

# Paso 1: Definir los datos (lista de pares [pH, energía] con
numpy)
pociones = np.array([
   # añade aquí tus datos
])

# Paso 2: Crear el modelo KMeans especificando el número de
clústeres
kmeans = KMeans(n_clusters=2, random_state=0)

# Paso 3: Entrenar el modelo con tus datos (fit)
kmeans.fit(pociones)

# Visualización
plt.scatter(pociones[:, 0], pociones[:, 1], c=kmeans.labels_,
cmap="viridis", marker="o")
plt.xlabel("pH")
plt.ylabel("Poder mágico")
plt.title("Clasificación de pociones con K-Means")
plt.show()

# Paso 4: Mostrar el resultado
print("Clasificación de cada frasco:", kmeans.labels_)
```

¡Bien hecho! Has aprendido a usar K-means. Como recompensa por ayudar al alquimista, este desbloquea la puerta para que puedas continuar tu aventura.

Estación 3—Métodos de Monte Carlo

Los métodos de Monte Carlo son una forma de predecir la probabilidad con la que ocurrirá un evento simulando el mismo experimento muchas veces y contando cuántas veces ocurre ese evento como resultado.

Mini-Desafío 1 Estación 3 —La Fuente de la Fortuna

Tras caminar un rato sin encontrar la siguiente habitación, empiezas a sentirte perdido. Llegas a una pequeña habitación con una **fuente mágica** y decides pedir un deseo para encontrar el camino.

La inscripción en la piedra dice:

"Si lanzas una moneda, hay un 50 % de posibilidades de que tu deseo se cumpla.
¿Pero qué pasa si lo intentas muchas veces?

Instrucciones

- Usa la calculadora para generar diez números aleatorios. Los números pares representan cara y los impares, cruz.
- Cuenta el número de veces que la moneda sale cara y divídelo entre el número total de lanzamientos (N=10). El resultado es la probabilidad.

Para utilizar los métodos de Monte Carlo, siempre debe haber **aleatoriedad** en los experimentos. Sin azar, no habría variación en los resultados, lo que significa que no podríamos estimar probabilidades.

Desafío 3 - El guardián de los dados del destino

Después de un largo rato caminando por un pasillo de piedra iluminado por antorchas, encuentras una habitación con una **mesa circular** cubierta de pergaminos, piedras de colores y algunos **datos.**

Sentado detrás de la mesa está el **Guardián de la Azar,** un esqueleto encapuchado con una voz profunda que toma 3 dados y te dice:
:

*"Viajero, esta habitación no se abre con fuerza ni ingenio... Soy el guardián de los **infinitos juegos de Montecarlo** y te propongo un reto. Voy a tirar estos tres dados y sumar los números. Elige: **mayor que 12 o menor o igual a 12**. Si aciertas, podrás avanzar, pero si no, quedarás encerrado aquí para siempre.*

Para decidir qué opción elegir, decides aplicar tus conocimientos de probabilidad y usar métodos de Monte Carlo para determinar tu mejor opción. El primer decimal de la mayor probabilidad te dará el tercer dígito del candado.

NIVEL FÁCIL: Siga los pasos a continuación para crear un código que le permita calcular la probabilidad de éxito en cada caso para N=10.000 experimentos.

1. 1. Crea 2 variables para contar cuántas veces aparece cada resultado.
   Clave:Debes inicializar las variables a 0.
2. 1. Genera 3 números aleatorios y súmalos.
   Clave:Debes utilizar la función random.random().
3. Comprueba si la suma es >12 o =<12 y suma 1 al contador correspondiente.
4. 1. Repita el proceso 10.000 veces.
   Clave:Debes poner el código dentro de un bucle "for".
5. Divida los contadores por N=10.000 para obtener la probabilidad y elija el más probable.

Desafío 3 - El guardián de los dados del destino

NIVEL MEDIO: Escribe el código para simular el experimento durante N=10.000 iteraciones (puedes usar los pasos de nivel fácil si lo necesitas), pero también debes calcular la probabilidad de cada resultado posible (de 3 a 18) al tirar los dados.

Clave: Debes usar un diccionario para los contadores. Realiza todos los experimentos y suma los valores del contador. Finalmente, después de todos los experimentos, divide todos los valores entre N.

NIVEL DIFÍCIL: Cree un programa desde cero que, para cualquier número de dados, repeticiones y valor umbral, calcule la probabilidad de cada resultado posible.

Clave: Debe crear una función con N, num_dices y umbral como parámetros que devuelva un diccionario con los posibles resultados como clave y las probabilidades como valor.

EXTRA (nivel experto):Agrega una visualización de la distribución de probabilidad que permite ver en un histograma (plt.bar) las probabilidades de cada valor.

Estación 4 – Aprendizaje Hebbiano

El aprendizaje hebbiano es una regla de aprendizaje para redes neuronales basada en la frase: «Las neuronas que se activan juntas, se conectan juntas». En otras palabras:

- Si dos neuronas se activan al mismo tiempo, la conexión entre ellas se **fortalece.**
- Si uno está activado y el otro no, la conexión se **debilita**

Mini-Desafío 1 Estación 4 – La Biblioteca de Ideas

De camino a la última sala, atraviesas un pasillo oscuro donde hay dos estatuas con símbolos brillantes. Cada vez que dices "luz", las dos estatuas se iluminan juntas. Pero si dices "sombra", solo una brilla.

Según la regla de Hebb, ¿qué pasará con la conexión entre estas dos estatuas si repites "luz" muchas veces?

a) Se debilitará
b) Seguirá igual
c) Será más fuerte

Mini-Desafío 1 Estación 4 – La Biblioteca de Ideas

Para calcular el peso de la conexión entre dos neuronas después de que se activan, se utiliza la fórmula hebbiana:

$$w_{final} = w_{initial} + (n_{together} \times \Delta w_{+}) + (n_{alone} \times \Delta w_{-})$$

Dónde:

- $w_{initial}$ = peso inicial de la conexión
- $n_{together}$ = número de veces que ambas neuronas se activan juntas
- Δw_{+} = aumento de peso debido a la activación articular
- n_{Alone} = número de veces que se activa solo una neurona
- Δw_{-} = pérdida de peso cuando solo se activa uno

Mini-Desafío 2 Estación 4 – Pesas de conexión

Para iluminar correctamente el pasillo, debes calcular la activación entre las dos estatuas (o neuronas). Para ello, debes usar la fórmula hebbiana.

Instrucciones

- El peso inicial de la conexión es 0,2.
- Cada vez que ambos se encienden juntos→**+0.1**
- Cada vez que solo se enciende una →**–0.05**
- Ambas luces se encienden juntas 3 veces, pero la luz A se enciende sola 2 veces.

Respuesta: $w_{final} = 0.2 + (3 \times 0.1) + (2 \times 0.05) = 0.6$

Desafío 4 - El Mural de las Neuronas

Estás muy cerca del final, pero aún te queda un último desafío por superar. En la última sala, encontrarás un mural con tres luces mágicas interconectadas (A, B, C) conectadas entre sí con hilos que representan pesos de conexión.
Una voz misteriosa te dice

"Para abrir la puerta final, debes descubrir qué conexiones se han fortalecido. Calcula los pesos finales tras varias activaciones y revela el patrón secreto."

Cada conexión comienza con un **peso de 0**. Si dos neuronas se activan simultáneamente, la conexión aumenta en 0,2, mientras que si solo se activa una, el peso disminuye en -0,07. Hay tres conexiones: AB, AC y BC. El último dígito del candado corresponde al último decimal del peso de la conexión AC después de la ronda 4. Activaciones por ronda:

Ronda	Neuronas activadas
1	A,B
2	B,C
3	A
4	A, C
5	A, B, C

NIVEL FÁCIL: Realice los cálculos solo para la conexión AB, con una tabla que muestre el resultado del peso de la conexión para cada ronda.

NIVEL MEDIO: Calcular los pesos finales de todas las conexiones en todas las rondas.

NIVEL DIFÍCIL: Calcula los pesos finales de todas las conexiones en todas las rondas, suponiendo que si las tres neuronas se activan simultáneamente, todas las conexiones reciben una bonificación de +0,2. Identifica la conexión más fuerte de la última ronda.

PREGUNTAS DE REFLEXIÓN

Tómate un momento para reflexionar sobre lo aprendido en este módulo. Responde atentamente las siguientes preguntas.

- **IA, matemáticas y toma de decisiones:**
  En este módulo, usaste matemáticas, probabilidad y lógica para resolver los desafíos de la Sala de Escape con IA. ¿Cómo te ayudó esta actividad a comprender cómo toma decisiones la IA? ¿Qué desafío te ayudó a comprenderlo mejor?

- **Aprendizaje mediante probabilidad y simulación:**
  Exploraste diferentes métodos de IA, como redes bayesianas, agrupamiento de K-medias, métodos de Monte Carlo y aprendizaje hebbiano. ¿Podrías explicar alguna de estas ideas con tus propias palabras y dar un ejemplo de cómo se aplicó en la sala de escape?

- **Aprendizaje por repetición y experiencia:**
  Algunos desafíos requerían repetir cálculos, simulaciones o ensayos para llegar a la respuesta correcta. ¿En qué se parece esto a cómo los sistemas de IA aprenden de los datos y la experiencia? ¿Se te ocurre un ejemplo real donde la IA aprende de esta manera?

CUESTIONARIO CON PREGUNTAS DE OPCIÓN MÚLTIPLE(una respuesta correcta por pregunta)

1. ¿En qué nos ayudan principalmente las redes bayesianas?

a) Agrupación de puntos de datos similares
b) Toma de decisiones bajo incertidumbre
c) Aceleración de los cálculos con muestreo aleatorio
d) Fortalecimiento de las conexiones neuronales

2. Si el 30% de los cofres brillantes contienen oro y el 80% de los cofres de oro son pesados, ¿qué regla usamos para encontrar la probabilidad de que un cofre pesado tenga oro?

a) Simulación de Monte Carlo
b) Aprendizaje hebbiano
c) Regla de Bayes
d) Algoritmo K-Means

3.¿Qué representa la "k" en K-Means?

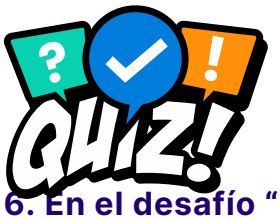
a) El número de puntos de datos
b) El número de clústeres
c) El número de dimensiones
d) El número de probabilidades

4. ¿Qué paso viene inmediatamente después de asignar puntos al centroide más cercano?

a) Aleatorizar nuevamente los centroides
b) Calcular los nuevos centros de cada cluster
c) Dibujar una red bayesiana
d) Estimación de probabilidades con dados

5. ¿Por qué son útiles los métodos de Monte Carlo en la estimación de probabilidad?

a) Eliminan la aleatoriedad por completo
b) Simulan muchos experimentos con aleatoriedad.
c) Agrupan los datos en categorías
d) Fortalecen las conexiones en red



CUESTIONARIO CON PREGUNTAS DE OPCIÓN MÚLTIPLE(una respuesta correcta por pregunta)

6. En el desafío "Guardián de los Dados del Destino", ¿qué se calcula?

a) ¿Qué cofre tiene oro?
b) ¿A qué grupo pertenece una poción?
c) La probabilidad de que las sumas de los dados sean mayores o menores que 12
d) El peso final de la conexión neuronal

7. Según la frase "las neuronas que se activan juntas, se conectan juntas", ¿qué sucede cuando dos neuronas se activan simultáneamente muchas veces?

a) La conexión se debilita
b) La conexión desaparece
c) La conexión se fortalece
d) Nada cambia

8. Si dos neuronas se activan juntas 3 veces (+0,1 cada una) y sólo una se activa 2 veces (–0,05 cada una), el peso inicial = 0,2, ¿cuál es el peso final?

a) 0.35
b) 0.60
c) 0.10
d) 0.45

9. ¿Cuál de las siguientes opciones distingue mejor las redes bayesianas de los métodos de Monte Carlo?

a) Las redes bayesianas utilizan reglas de probabilidad, Monte Carlo utiliza simulaciones
b) Ambos son el mismo método con diferentes nombres.
c) Las redes bayesianas son aleatorias, Monte Carlo es determinista
d) Monte Carlo fortalece conexiones como el aprendizaje hebbiano

10.En las redes neuronales artificiales, el aprendizaje hebbiano ayuda principalmente con:

a) Agrupar datos en grupos
b) Fortalecimiento de las conexiones entre neuronas coactivadas
c) Predecir resultados con incertidumbre
d) Ejecutar miles de simulaciones aleatorias

MÓDULO N.° 3 | Scratch se encuentra con la inteligencia artificial

INTRODUCCIÓN

El objetivo de este módulo es introducir el lenguaje de programación Scratch y comprender cómo Scratch puede ser una herramienta fácil y efectiva para introducir conceptos de programación.

Al finalizar el módulo podrás adquirir diferentes habilidades, tales como:

- **Entender cómo funciona Scratch** Proporciona una plataforma accesible para aprender conceptos de programación.
- **Domina el aprendizaje basado en bloques fundamentales** Habilidades de programación y pensamiento computacional.
- **Integrar inteligencia artificial** extensiones en proyectos creativos de Scratch.

Duración del módulo	2 horas(1 hora de aprendizaje + 1 hora de ejercicios prácticos

MATERIALES EDUCATIVOS

Scratch es un poderoso lenguaje de programación basado en lo visual, lo que significa que, a diferencia de los lenguajes de programación tradicionales, Scratch no se basa en sintaxis o código basado en líneas; funciona conectando bloques para crear lógica.
Usar Scratch como introducción a la programación ofrece muchas ventajas. En este módulo, aprenderás sobre la historia de Scratch y cómo programar con esta plataforma en línea. Al final del módulo, aprenderás a añadir extensiones de IA a las aplicaciones de Scratch.

UNIDAD 3.1 INTRODUCCIÓN A SCRATCH

Scratch es un potente lenguaje de programación visual y una comunidad en línea desarrollado por el MIT Media Lab. A diferencia de los lenguajes de programación tradicionales, que requieren aprender y escribir código complejo, Scratch utiliza bloques de código que encajan como piezas de un rompecabezas. Este concepto hace que la programación sea accesible para todos, independientemente de su nivel de experiencia y conocimientos de programación.

POR QUÉ SCRATCH ES UNA GRAN HERRAMIENTA PARA APRENDER PROGRAMACIÓN

Los principales beneficios de aprender a codificar usando Scratch son que elimina muchas barreras que tradicionalmente hacen que la programación sea un desafío para los nuevos codificadores.
Por ejemplo, Scratch tiene las siguientes diferencias con los lenguajes de programación tradicionales:

- **Sin errores de sintaxis:**Los bloques de código en Scratch solo se conectan de manera que tengan sentido lógico, lo que evita errores de sintaxis frustrantes.
- **Retroalimentación visual inmediata:**Scratch te permite ver los resultados de tu código instantáneamente en el escenario.
- **Piso bajo, techo alto:**Scratch es fácil de aprender, pero también permite crear proyectos sofisticados.
- **Libertad creativa:**Crea juegos, animaciones, historias interactivas, simulaciones y más.
- **Comunidad colaborativa:**Comparta proyectos, combine el trabajo de otros y aprenda de sus compañeros.
- **Multiplataforma:**Scratch funciona en todos los navegadores web principales en cualquierdispositivo,Incluyendo tabletas y teléfonos inteligentes.

APLICACIONES DE SCRATCH EN EL MUNDO REAL

Scratch puede ser un lenguaje basado en bloques sin sintaxis; sin embargo, enseña conceptos fundamentales de programación que pueden transferirse directamente a otros lenguajes como Python, JavaScript y Java.

Scratch maneja los conceptos de programación de las siguientes maneras:

Concepto de programación	En Scratch	Aplicación en el mundo real
Secuencias	Bloques apilados en orden	Instrucciones paso a paso en cualquier programa
Bucles	Bloques de repetición	Automatizar tareas repetitivas
Condicionales	Bloques si-entonces	Lógica de toma de decisiones en aplicaciones
Variables	Bloques de almacenamiento de datos	Almacenamiento y gestión de información
Eventos	Bloques que desencadenan acciones	Manejo de la interacción del usuario en el software
Paralelismo	Múltiples scripts ejecutándose	Aplicaciones multiproceso

UNIDAD 3.2 HABILIDADES BÁSICAS DE PROGRAMACIÓN BASADA EN BLOQUES

CÓMO CREAR UNA CUENTA SCRATCH

Abrir una cuenta es sencillo y gratuito. Empieza visitando el sitio web de Scratch.https://scratch.mit.edu/.Luego crea una cuenta seleccionando “Unirse a Scratch”

NAVEGANDO POR LA INTERFAZ DE SCRATCH

La interfaz de Scratch controla cómo se ve un programa en la pantalla. Hay siete elementos diferentes en Scratch que necesitas conocer para empezar a programar en la plataforma. Son:

- Interfaz
- Sprites
- Escenario
- Bloques
- Disfraces
- Telones de fondo
- Sonidos

La interfaz de Scratch se divide en tres áreas principales: el Escenario (arriba a la derecha), donde se muestran los proyectos; el Área de Código (centro de la pantalla), donde puedes arrastrar y ensamblar bloques para crear scripts; y el Área de Sprites (abajo a la derecha), donde puedes agregar y administrar personajes, o sprites, y fondos.

El lado izquierdo de la interfaz contiene la Paleta de Bloques, una caja de herramientas con distintas categorías de bloques de código para crear programas visualmente. En resumen, los elementos principales de la interfaz son:

- Área de Sprites (abajo a la derecha):Muestra todos los sprites (personajes y objetos) de tu proyecto. Haz clic en un sprite para editar su código.
- Área de Código (Centro):El espacio de trabajo donde arrastras y conectas bloques para crear scripts para el sprite seleccionado.
- Escenario (arriba a la derecha):El área de salida visual donde tus sprites actúan e interactúan. Esto es lo que ve tu público.

COMPRENSIÓN DE LAS CATEGORÍAS DE BLOQUES

Scratch organiza los bloques de código en nueve categorías con códigos de colores. Cada categoría cumple una función específica en el programa; las categorías son las siguientes:

1. **Azul - Movimiento:** La categoría de movimiento permite mover tu sprite por la pantalla. Si quieres que un sprite avance, puedes usar el bloque "Cambiar X por 10".
2. **Púrpura – Apariencia:**La categoría de apariencia se centra en cambiar la apariencia de sprites y fondos. Puedes usar un bloque de cambio de vestuario para cambiar la apariencia de un sprite.
3. **Magenta – Sonido:** La categoría de sonido se centra en hacer que los sprites emitan sonidos. Puedes usar el bloque "Reproducir sonido hasta el final" para reproducir un sonido.
4. **Amarillo – Eventos:**Todo proyecto Scratch necesita un evento; esto hace que sucedan cosas cuando se realizan interacciones.

5. **Naranja claro – Control:**La categoría de control se centra en la frecuencia con la que un sprite debe realizar una acción o el tiempo durante el cual debe permanecer inmóvil. Si arrastras un bloque "Esperar 1 segundo" a tu programa, este esperará 1 segundo antes de iniciarse.
6. **Cerceta – Detección:**Esta categoría trata de comprobar si un objeto está en contacto con otro objeto para que pueda producirse una acción.
7. **Verde – Operador:**Estos controles se pueden utilizar para realizar operaciones matemáticas y lógicas, como suma, resta y concatenación de cadenas.
8. **Naranja oscuro – Variables:**Las variables se utilizan para almacenar y gestionar información como puntuaciones o puntos de salud.
9. **Rojo - Mis bloques:**En esta categoría podrás crear tus propios bloques personalizados.

UNIDAD 3.3 CÓMO CREAR PROYECTOS EN SCRATCH

Para crear un proyecto en Scratch, comience agregando un personaje (sprite) y un fondo de la biblioteca.

Agregar un nuevo Sprite

En la esquina inferior derecha del editor de Scratch, encontrarás los controles de sprites:

- Elige un Sprite: explora la biblioteca incorporada de sprites (animales, personas, objetos, personajes de fantasía, etc.)
- Pintar: Crea tu propio sprite usando las herramientas de dibujo
- Sorpresa: agrega un sprite aleatorio para inspirarte
- Subir Sprite: Importa un archivo de imagen desde tu computadora.

Cambiando el telón de fondo

El fondo define el escenario de tu proyecto. Haz clic en el icono del fondo (abajo a la derecha del escenario) para:

- Elija entre fondos prediseñados (exteriores, interiores, espaciales, abstractos, etc.)
- Pinta el tuyo usando el editor de fondos.
- Sube una imagen desde tu computadora.

Puedes tener varios fondos y alternar entre ellos programáticamente usando los bloques "cambiar fondo a [nombre]. Esto es útil para crear juegos o historias con varias escenas.

MOVER UN SPRITE Y ACTIVAR UNA ACCIÓN

1. **Paso 1:** Agregar un bloque de eventos:Haz clic en la categoría Eventos (amarilla). Arrastra el bloque "Al hacer clic en la bandera verde" al área de código. Este bloque inicia el programa al hacer clic en la bandera verde sobre el escenario.

2. **Paso 2:** Agregar movimiento:Haz clic en la categoría Movimiento (azul). Arrastra el bloque "Mover 10 pasos" y colócalo debajo del bloque de evento. Cambia el número de 10 a 50 haciendo clic en él.

3. **Paso 3:** Agregar un efecto de sonidoHaz clic en la categoría Sonido (magenta). Arrastra "Reproducir sonido [miau]" hasta que esté listo y conéctalo debajo del bloque de movimiento.

4. **Paso 4**: Haz que el Sprite hable:Haz clic en la categoría "Apariencia" (morado). Arrastra "Di [¡Hola!] durante 2 segundos" y añádelo a tu script. Puedes cambiar "¡Hola!" por el mensaje que quieras.

5. **Paso 5:** Pruebe su programa:Haz clic en la bandera verde sobre el escenario. Tu sprite debería moverse 50 pasos a la derecha, maullar y decir tu mensaje.

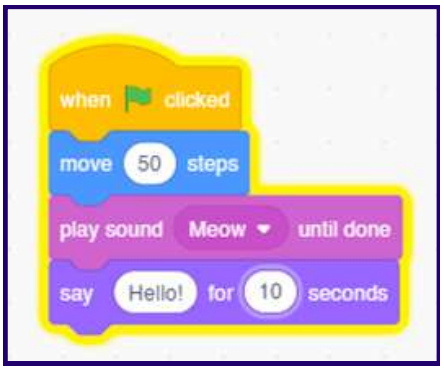


Figura 3 Lógica de bloques de Scratch para la interacción de movimiento y sonido: Scratch de MIT Media Lab

UNIDAD 3.4 INTRODUCCIÓN A LAS EXTENSIONES DE IA DE SCRATCH

¿QUÉ ES UNA EXTENSIÓN EN SCRATCH?

Las extensiones son módulos adicionales que amplían las capacidades de Scratch más allá de las categorías de bloques estándar. Piensa en ellas como conjuntos de herramientas especializados que otorgan a tus sprites nuevas capacidades, desde leer texto en voz alta hasta reconocer imágenes mediante inteligencia artificial.

Scratch 3.0 incluye extensiones de software, que agregan funciones computacionales y extensiones de hardware, que se conectan a dispositivos físicos como robots y kits electrónicos. En esta unidad nos centramos en las extensiones de software impulsadas por IA que incorporan aprendizaje automático y procesamiento del lenguaje natural a sus proyectos.

AGREGAR UNA EXTENSIÓN EN SCRATCH

Agregar una extensión es sencillo:

1. Busque el botón con el símbolo "+" en la parte inferior izquierda del área de bloques.
2. Haga clic en el símbolo para abrir la Biblioteca de extensiones
3. Busque las extensiones disponibles y haga clic en una para agregarla a su proyecto.
4. Nuevos bloques aparecerán en tu paleta inmediatamente listos para usar.

La mayoría de las extensiones de IA requieren una conexión a internet activa, ya que el procesamiento de la inteligencia artificial se realiza en servidores en la nube, no en su ordenador. Asegúrese de estar conectado a internet antes de usar estas extensiones. Además, tenga en cuenta su privacidad y evite subir información personal o confidencial al entrenar modelos de aprendizaje automático.

POR QUÉ Y CÓMO SE UTILIZA LA IA EN PROYECTOS CREATIVOS

La inteligencia artificial permite que tus proyectos de Scratch hagan cosas que serían casi imposibles solo con la programación tradicional:

Capacidad de IA	En Scratch	Paralelo en el mundo real
Reconocimiento de imágenes	Un juego que identifica objetos que dibujas o muestras a la cámara	Google Photos etiqueta automáticamente personas y lugares
Clasificación de textos	Un chatbot que entiende si los mensajes son preguntas, declaraciones o comandos.	Filtros de spam de correo electrónico que categorizan los mensajes
Síntesis de voz	Personajes que leen historias en voz alta o narran eventos del juego.	Sistemas de navegación GPS que proporcionan indicaciones de voz
Análisis de sentimientos	Una mascota virtual que reacciona de forma diferente a palabras amables y malas	Sistemas de navegación GPS que proporcionan indicaciones de voz
Reconocimiento de patrones	Una aplicación de música que identifica si los sonidos son de batería, piano o canto.	Shazam identifica canciones a partir de clips de audio cortos

La IA amplía las posibilidades creativas al hacer que los proyectos sean adaptativos (respondan de manera diferente según la información), inteligentes (tomen decisiones a partir de patrones aprendidos) e interactivos (comprenden múltiples formas de información, como imágenes, texto y sonido).

DESCRIPCIÓN GENERAL DE LAS EXTENSIONES DE SCRATCH RELACIONADAS CON LA IA

EXTENSIONES OFICIALES DE INTELIGENCIA ARTIFICIAL DE SCRATCH

- **Texto a voz:**Convierte texto escrito en palabras habladas con múltiples voces e idiomas.
- **Traducir:**Traduce texto entre idiomas mediante traducción automática.
- **Detección de vídeo:**Detecta movimiento y presencia a través de cámara web (visión artificial básica)

EXTENSIONES DE IA DE TERCEROS

- **Aprendizaje automático para niños:** Entrene modelos personalizados para reconocimiento de imágenes, texto, números y sonidos.
- **Máquina enseñable:**La herramienta de Google para el entrenamiento rápido de modelos, integrable con Scratch
- **Detección facial:**Detecta rostros y rasgos faciales a través de una cámara web.
- **Reconocimiento de voz:**Convierte palabras habladas en texto (control de voz)

UNIDAD 3.5 EXTENSIÓN DE IA: APRENDIZAJE AUTOMÁTICO PARA NIÑOS

Machine Learning for Kids (ML4K) es una plataforma educativa desarrollada por IBM que facilita el aprendizaje automático a estudiantes de todas las edades. Ofrece una interfaz intuitiva para entrenar modelos de IA y se integra a la perfección con Scratch mediante bloques personalizados.

Con ML4K, puedes entrenar modelos para:

- Reconocer y clasificar **imágenes** (por ejemplo, distinguir gatos de perros o identificar formas dibujadas a mano)
- Comprender y categorizar **textos** (por ejemplo, detectar si los mensajes son cumplidos o insultos)
- Identificar patrones en **números** (por ejemplo, predecir resultados basándose en información numérica)
- Reconocer **sonidos** (por ejemplo, diferenciar entre notas musicales o comandos de voz)

¿QUÉ ES EL APRENDIZAJE AUTOMÁTICO?

El aprendizaje automático es un subconjunto de la inteligencia artificial donde las computadoras aprenden a realizar tareas analizando ejemplos, en lugar de seguir reglas preprogramadas. Por ejemplo, en lugar de programar la descripción de cada raza de perro, se le muestran a la computadora miles de imágenes de perros y esta
aprende a reconocer las características que definen las razas. Así es como servicios como Google Fotos etiquetan automáticamente las fotos.

PASO A PASO: ENTRENAMIENTO Y USO DE UN MODELO

PASO 1: ACCEDA AL APRENDIZAJE AUTOMÁTICO PARA NIÑOS

Abra su navegador web y navegue hasta
https://machinelearningforkids.co.uk/

Haga clic en “Comenzar” y, para este ejemplo,
elija crear una cuenta sin registrarse.

Paso 2: Crear un nuevo proyecto

Una vez que haya iniciado sesión, haga clic en “Proyectos” y luego en “Agregar un nuevo proyecto”. Se le pedirá que

- **Ponle nombre a tu proyecto:** Elija un nombre descriptivo (por ejemplo, "Detector de emociones" o "Clasificador de reciclaje")
- **Seleccionar el tipo de reconocimiento:** Elija entre Texto, Números, Imágenes o Sonidos

Para este tutorial, seleccione "Texto" para crear un clasificador de sentimientos.

Paso 3: Define tus etiquetas (categorías)

Los modelos de aprendizaje automático clasifican las entradas en categorías llamadas "etiquetas".
Para un clasificador de sentimientos, cree dos etiquetas:

- **Positivo:**Mensajes felices, entusiastas o amables.
- **Negativo:**Mensajes tristes, enojados o malos.

Haga clic en “Entrenamiento” y luego en “Agregar nueva etiqueta” para cada categoría.

Paso 4: Proporcionar ejemplos de capacitación

El modelo aprende estudiando ejemplos. Para cada etiqueta, proporcione al menos de 5 a 10 frases de ejemplo:

Ejemplos positivos	Ejemplos negativos
¡Esto es increíble!	I do not like this.
Esto no me gusta.	Esto es terrible.
Me encanta aprender cosas nuevas	Esto me enoja.

Consejo: Cuantos más ejemplos proporciones, mejor será el rendimiento de tu modelo. Intenta incluir al menos diez ejemplos por etiqueta y asegúrate de que la redacción sea variada.

Paso 5: Entrena tu modelo

Una vez que haya agregado suficientes ejemplos, regrese al proyecto y seleccione "Aprender y probar". Luego, haga clic en el botón "Entrenar nuevo modelo de aprendizaje automático". El sistema analizará sus ejemplos y creará un modelo capaz de clasificar texto nuevo. El entrenamiento suele tardar entre 1 y 3 minutos. Verá un indicador de progreso. Al finalizar, recibirá un mensaje de confirmación.

Paso 6: Pruebe su modelo

Antes de usar tu modelo en Scratch, pruébalo en ML4K. Simplemente escribe una frase que no hayas incluido en tus datos de entrenamiento (p. ej., "¡Qué feliz estoy hoy!") y haz clic en "Probar". El modelo predecirá qué etiqueta coincide mejor con tu entrada y mostrará un porcentaje de confianza.

Si el modelo comete errores, regrese a la sección de entrenamiento y agregue más ejemplos para mejorar la precisión.

Paso 7: Conéctese a Scratch

ML4K proporciona una integración especial con Scratch:

1. Haga clic en "Crear" en la navegación del proyecto.
2. Seleccione "Scratch 3"
3. Esto abre un editor Scratch modificado con su modelo ML precargado como bloques personalizados.

Verás nuevos bloques específicos para tu proyecto, como:

- reconocer texto [ ] (etiqueta)
- reconocer texto [ ] (confianza)

Paso 8: Usa bloques ML en tu proyecto Scratch

Crea un proyecto Scratch simple que use tu modelo entrenado: comienza agregando un sprite (prueba el sprite "Robot"), luego crea una variable llamada "entrada de usuario" y luego crea este script:
Al hacer clic en la bandera verde, pregunta [Dime cómo te sientes hoy] y espera. Configura [entrada del usuario] como (respuesta). Si <recognise text (entrada del usuario) (etiqueta) = [Positivo]>, di [¡Me alegra que te sientas bien!] durante 3 segundos. De lo contrario, di [Lamento que te sientas mal. ¡Las cosas mejorarán!] durante 3 segundos.

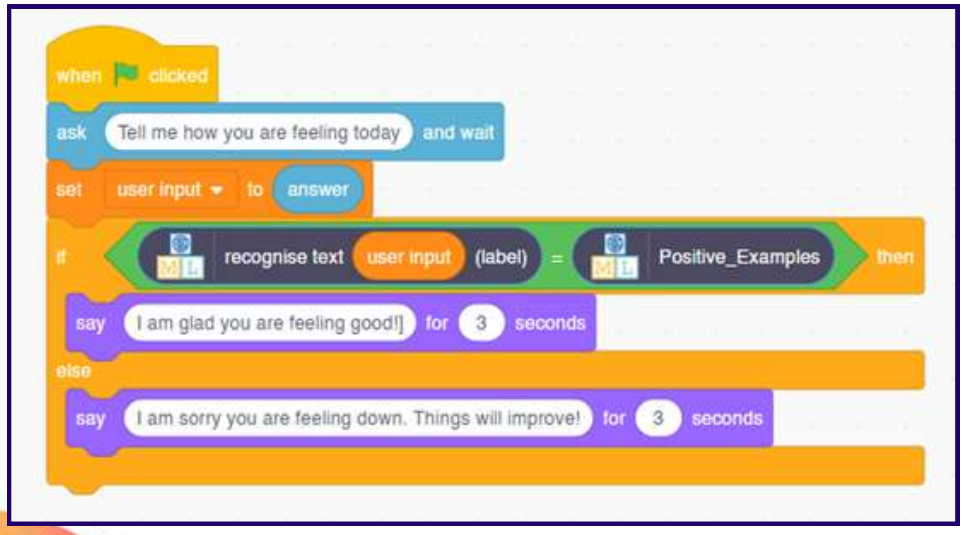


Figura 4 Secuencia de bloques de código en Scratch que ilustra una cadena básica de evento-acción: Scratch de MIT Media Lab

Paso 9: Probar y refinar

Ejecuta tu proyecto y escribe diferentes mensajes. Observa cómo responde el sprite según la clasificación de la IA. Si el modelo clasifica el texto incorrectamente, regresa a ML4K, añade más ejemplos de entrenamiento, vuelve a entrenar y actualiza tu proyecto de Scratch.

UNIDAD 3.6 EXTENSIÓN DE IA: TEXTO A VOZ

La síntesis de voz (TTS) es una tecnología basada en IA que convierte texto escrito en audio hablado. Esta extensión oficial de Scratch utiliza síntesis de voz basada en la nube para dar a tus sprites voces realistas en múltiples idiomas y tonos.

TTS hace que los proyectos sean más accesibles (los usuarios con discapacidad visual pueden escuchar el contenido), más atractivos (el diálogo hablado se siente más dinámico que las burbujas de texto) y más versátiles (los proyectos pueden narrar historias, proporcionar instrucciones o crear herramientas de aprendizaje de idiomas).

AGREGAR LA EXTENSIÓN DE TEXTO A VOZ

Paso 1: Abra la Biblioteca de Extensiones. En el editor de Scratch, haga clic en el botón azul "Añadir Extensión" (con un símbolo "+") en la esquina inferior izquierda de la paleta del bloque.

Paso 2: Desplázate por la biblioteca de extensiones y haz clic en "Texto a voz" (tiene un altavoz con bocadillos de diálogo). Aparecerán nuevos bloques en tu paleta, bajo la nueva categoría "Texto a voz".

Paso 3:Utilice los bloques en su código:

Pronuncia una frase: Arrastra el bloque "hablar [hola]" al área de código. Haz clic dentro del bloque para cambiar el texto a lo que quieras que diga tu sprite.

Cambiar la voz: Usa el bloque de voz [alto] para elegir una voz diferente. Las opciones incluyen Alto, Tenor, Chillido, Gigante y Gatito.

Cambiar el idioma: Para cambiar el idioma, utilice el idioma establecido para bloquear.

Figura 5 Secuencia de bloques de código para conversación en Scratch: Scratch de MIT Media Lab

USANDO BLOQUES DE SCRATCH PARA CREAR PROYECTOS INTERACTIVOS.

Combine los bloques de texto a voz con otros bloques de Scratch para crear proyectos interactivos. Por ejemplo, puedes usar el bloque de evento "Al hacer clic en la bandera verde" de la categoría "Eventos" para iniciar el discurso. También puedes usar el bloque "Preguntar y esperar" de la categoría "Detección" y el bloque "Responder" con el bloque "Hablar" para que tu sprite repita lo que escribe el usuario.

BLOQUES CLAVE DE TEXTO A VOZ

Esta extensión proporciona bloques para controlar el habla, son los siguientes:

Bloque	Función	Ejemplo
hablar [texto]	Speaks the specified text aloud	hablar [Hola, bienvenido a mi proyecto!]
establecer la voz en [voz]	Cambia el carácter de la voz.	Establecer la voz en [tenor] (opciones: alto, tenor, chirrido, gigante, gatito)
Establezca el idioma en [idioma]	Cambia el idioma del habla.	Establecer idioma en [Español] (23 idiomas compatibles)

OPCIONES DE VOZ DE TEXTO A VOZ

Hay varias opciones de voz de texto a voz con las que puedes experimentar para que coincidan con la personalidad de tus sprites; son las siguientes:

- Alto: Voz neutra de tono medio.
- Tenor: Voz ligeramente más aguda.
- Chillido: Voz aguda y infantil.
- Gigante: Voz profunda y resonante.
- Gatito: Voz muy aguda y linda.

EJEMPLOS DE CÓMO SE PUEDE UTILIZAR LA CONVERSIÓN DE TEXTO A VOZ EN SCRATCH

- **Historias interactivas:** Los personajes narran el desarrollo de la trama y hablan diálogos.
- **Cuestionarios educativos:** La computadora lee las preguntas en voz alta para mayor accesibilidad.
- **Aprendizaje de idiomas**: Práctica de pronunciación con diferentes idiomas
- **Asistentes virtuales:** Asistentes como Siri que responden en voz alta.
- **Guías de audio:** Visitas a museos o guías de la ciudad con información narrada
- **Retroalimentación dinámica:** Animaciones habladas o pistas en los juegos.

REFERENCIAS

Libro, Wiki y Documentación

- Scratch Foundation (MIT Media Lab): documentación de la plataforma, recursos educativos y descripciones de la interfaz https://scratch.mit.edu/

- Aprendizaje automático para niños (Dale Lane, IBM), 11 de febrero de 2021: Tutoriales de plataforma, metodología de entrenamiento de modelos y guías de integración. ISBN-13: 9781718500563

- Scratch Wiki: Documentación técnica para extensiones y descripciones de bloques. https://scratch-wiki.info/

- Manual de desarrollo de código de IA Scratch: una guía para principiantes sobre cómo construir sistemas inteligentes. Libro de bolsillo – 11 de febrero de 2025. ISBN-13 : 979-8310433649

ENLACES A PLATAFORMAS DE APRENDIZAJE

Sitio web oficial de Scratch
La plataforma oficial de Scratch, desarrollada por el MIT Media Lab. Crea una cuenta, crea proyectos, explora millones de creaciones de la comunidad y accede a tutoriales y recursos de aprendizaje.
https://scratch.mit.edu/

Aprendizaje automático para niños
Plataforma educativa de IBM para entrenar modelos de aprendizaje automático personalizados (texto, imágenes, números, sonidos) e integrarlos con Scratch. Incluye hojas de trabajo, tutoriales y modelos preentrenados.
https://machinelearningforkids.co.uk/

OPCIONES DE VOZ DE TEXTO A VOZ

Enseñando conceptos de IA de forma creativa con Scratch (Codingal)
Entrada de blog que explora enfoques creativos para enseñar conceptos de IA a través de Scratch, con ideas de proyectos y estrategias pedagógicas.
https://www.codingal.com/coding-for-kids/blog/teach-kids-ai-codng-concepts-creatively-using-scratch/

Estudio de aprendizaje automático Scratch
Colección seleccionada de proyectos de Scratch con extensiones de aprendizaje automático. Explora para inspirarte y remezcla proyectos existentes para aprender de ellos.
https://scratch.mit.edu/studios/3995548/

EJERCICIO PRÁCTICO

Objetivo: Crea un proyecto usando la extensión Text-to-Speech para experimentar con diferentes voces que coincidan con las personalidades de tus sprites.

Instrucciones: Siga los pasos a continuación para crear un sprite que diga en voz alta la entrada del usuario, demostrando el poder de combinar la entrada de texto con la salida de voz.

Paso 1: Configura tu proyecto: Crea un nuevo proyecto de Scratch o continúa con uno existente. Elige un sprite que hable por sí solo: los sprites "Robot" o "Mago" son ideales.

Paso 2:Agregar la extensión: agregue la extensión de texto a voz como se describe arriba.

Paso 3: Crea el script y construye este script para tu sprite:
Cuando hizo clic la bandera verde
└── establecer la voz en [gigante]
└── hablar [¡Hola! Puedo decir lo que escribas.]
└── para siempre
└── preguntar [¿Qué debo decir?] y esperar
  └── hablar (responder)

Paso 4: Prueba tu proyecto. Haz clic en la bandera verde y escribe mensajes cuando se te pida. Tu sprite los leerá en voz alta.
Pruebe diferentes voces e idiomas cambiando los bloques de voz e idioma establecidos.

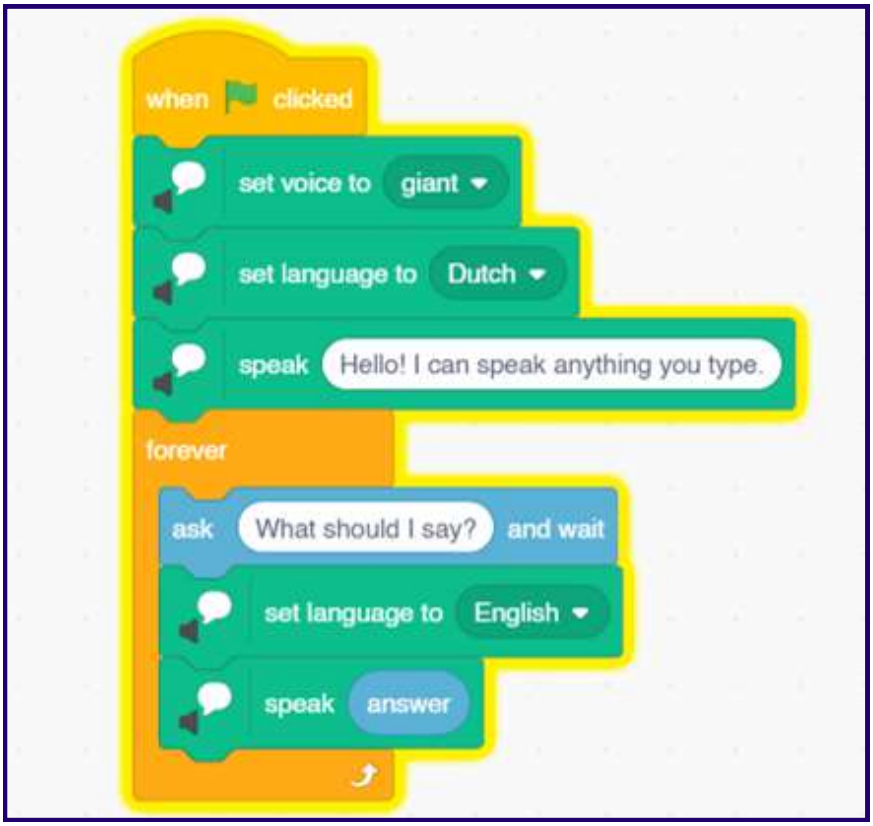


Figura 6 Secuencia de bloques de código en Scratch: Scratch de MIT Media Lab

Preguntas de reflexión

Tómate un tiempo para reflexionar profundamente sobre lo aprendido en este módulo. Responde las siguientes preguntas con atención:

- **Pregunta de reflexión 1:**¿Qué aspecto del uso de IA en Scratch te sorprendió más? ¿Hubo algo que funcionara diferente a lo esperado? Explícalo con ejemplos específicos de tu proyecto.

- **Pregunta de reflexión 2**:¿Cómo usarías el aprendizaje automático en un juego o una historia interactiva? Describe una idea de proyecto específica y explica cómo la IA la haría más atractiva o inteligente que la programación tradicional por sí sola.

- **Pregunta de reflexión 3**:¿Qué problemas o desafíos enfrentaste al entrenar tu modelo de aprendizaje automático o integrar extensiones de IA? ¿Cómo los resolviste? Si te topaste con problemas que no pudiste resolver, ¿qué estrategias probarías?

CUESTIONARIO CON PREGUNTAS DE OPCIÓN MÚLTIPLE(una respuesta correcta por pregunta)

1. ¿Qué afirmación describe mejor a Scratch?

a) Un lenguaje de programación textual utilizado por ingenieros de software profesionales.

b) Un lenguaje de programación visual basado en bloques que hace que la codificación sea accesible para principiantes.

c) Un servicio de inteligencia artificial para construir modelos de aprendizaje automático.

d) Un kit de robot de hardware para enseñar programación a niños.

2. ¿Cuál es uno de los principales beneficios de utilizar programación basada en bloques en Scratch?

a) Requiere que los estudiantes memoricen la sintaxis, lo que mejora las habilidades de memoria.

b) Los bloques solo se conectan de formas lógicamente correctas, lo que evita errores de sintaxis.

c) Se centra solo en la codificación basada en texto, lo que es más desafiante para los estudiantes.

d) Solo lo pueden usar personas con experiencia previa en programación.

3. ¿Qué categoría de bloque de Scratch usarías para crear un bucle que repita el código 10 veces?

a) Movimiento

b) Eventos

c) Controlados

d) Operadores

4. ¿Qué debes hacer antes de poder utilizar un modelo de Machine Learning for Kids en tu proyecto de Scratch?

a) Nada: la extensión viene entrenada previamente con un modelo que funciona para todos los propósitos.

QUIZ WITH MULTIPLE-CHOICE QUESTIONS (one correct answer per question)

b) Proporcione ejemplos de entrenamiento para cada categoría que desee que el modelo reconozca y luego entrene el modelo.
c) Compre un dispositivo de hardware especial para habilitar el aprendizaje automático.
d) Escriba el código Python para crear un algoritmo de aprendizaje automático e impórtelo a Scratch.

5. ¿Cuál de las siguientes NO es una capacidad relacionada con la IA mencionada en este módulo?
a) Reconocer objetos o imágenes a través de la clasificación de imágenes
b) Hacer que un sprite realice una rutina de baile usando un bucle de repetición
c) Convertir palabras habladas en texto a través del reconocimiento de voz
d) Traducir texto del inglés al español dentro de un proyecto

6. ¿Qué extensión de Scratch usarías para que un sprite narre una historia en voz alta?
a) Detección de video
b) Traducción
c) Texto a voz
d) Aprendizaje automático para niños

7. ¿Por qué es importante proporcionar múltiples ejemplos de entrenamiento al crear un modelo de aprendizaje automático?
a) El modelo no puede funcionar con menos de 100 ejemplos.
b) Más ejemplos ayudan al modelo a aprender patrones con mayor precisión y a clasificar nuevas entradas de manera más confiable.
c) Los ejemplos de entrenamiento solo se usan como decoración y no afectan el rendimiento.
d) Múltiples ejemplos ralentizan el modelo, haciéndolo más preciso.

8. ¿Qué ventajas ofrece la combinación de Machine Learning for Kids con Text-to-Speech?
a) Hace que el proyecto se ejecute más rápido
b) Entiende las entradas y comunica las respuestas
c) Reduce la cantidad de código que necesita escribir
d) Permite que el proyecto funcione sin conexión a Internet

MÓDULO N.° 4 | Programación de juegos en Python (bibliotecas Pygame/Arcade) con IA

INTRODUCCIÓN

En este módulo breve y fácil de usar, te convertirás en creador de juegos, no solo en jugador. Aprenderás a diseñar y programar un pequeño videojuego con Python y Pygame, y descubrirás cómo lograr que los personajes de tu juego se comporten de forma inteligente mediante formas sencillas de Inteligencia Artificial (IA).

No te preocupes si nunca has programado: te lo explicaremos todo paso a paso. Empezarás creando una pequeña ventana de juego y moviendo un cuadrado en la pantalla. Más adelante, explorarás cómo hacer que el juego reaccione a tus acciones, por ejemplo, haciendo que un enemigo note tus movimientos o cambie de color al acercarte.

Este módulo no se trata solo de ejecutar código, sino también de comprenderlo y experimentarlo. Probarás y modificarás ejemplos para ver cómo pequeños cambios pueden generar diferentes comportamientos. De esta forma, aprenderás cómo piensan y responden los juegos, de forma similar a los sistemas de IA simples del mundo real.

Al finalizar el módulo podrás adquirir diferentes habilidades, tales como:

- Cree juegos 2D interactivos simples utilizando Python y las bibliotecas Pygame o Arcade.
- Agregue comportamientos básicos de IA para que los personajes del juego puedan reaccionar a las acciones del jugador.
- Utilice el pensamiento lógico y computacional para diseñar juegos interactivos y receptivos.
- Experimente, pruebe y modifique el código del juego para observar cómo los pequeños cambios afectan el comportamiento.
- Reflexione sobre cómo se utiliza la IA en la vida real, en áreas como la educación, la atención médica, las finanzas y el entretenimiento.

Duración del módulo	**2 horas(1 hora de aprendizaje + 1 hora de ejercicios prácticos)**

MATERIALES EDUCATIVOS

UNIDAD 4.1 INTRODUCCIÓN A PYGAME

Antes de empezar a codificar, es importante saber con qué herramientas trabajarás y para qué sirven:

- **Python 3** El lenguaje de programación que se utiliza para dar instrucciones a la computadora. Se usa ampliamente en escuelas, universidades y empresas para programación, análisis de datos e inteligencia artificial.

- **Pygame –**Una biblioteca gratuita de Python que te ayuda a crear juegos 2D sencillos. Ofrece funciones listas para usar para abrir una ventana de juego, dibujar formas, mostrar colores y reaccionar al teclado o al ratón.

- **Editor de código (IDLE, Thonny o VS Code)**El programa donde escribirás, guardarás y ejecutarás tu código Python. En este módulo, puedes usar IDLE (incluido con Python) u otro editor recomendado por tu profesor.

Estas herramientas se irán introduciendo paso a paso, para que puedas instalarlas y realizar las actividades por tu cuenta, incluso si es tu primer contacto con la programación.

Meta: Aprende a abrir una ventana de juego, mostrar colores y mover un cuadrado con el teclado.

Paso 1: Instale Python 3 y abra su entorno de programación
Antes de comenzar a usar Pygame, debes asegurarte de que Python 3 esté instalado en tu computadora.

Instalar Python 3

- Vaya al sitio web oficial de Python:https://www.python.org/downloads/.
- Descargue la versión para su sistema (Windows, macOS o Linux).
- Durante la instalación, marque la opción “Agregar Python a PATH” antes de hacer clic en Instalar ahora.
- Una vez instalado, reinicie su computadora (opcional pero recomendado).

UNIDAD 4.2 VERIFICAR SU INSTALACIÓN

Abra el Símbolo del sistema (Windows) o la Terminal (macOS/Linux) y escriba:

python --version o **py --version**

muestra algo como Python 3.13.3, todo está listo.

UNIDAD 4.3 INSTALAR UN EDITOR O USAR IDLE

Puedes escribir código Python usando uno de estos:

- IDLE (viene instalado con Python)
- Thonny (IDE simple para principiantes) –https://thonny.org3.
- VS Code (opción avanzada) –https://code.visualstudio.com

Para abrir IDLE:

- En Windows: presioneInicio → IDLE (Python 3.x)
- En macOS: busca "IDLE" en Launchpad
- En Linux: ejecute idle3 en la terminal

Una vez abierto, puede crear un nuevo archivo (Archivo → Nuevo archivo), escribir código y ejecutarlo (Ejecutar → Ejecutar módulo o presionar F5).

UNIDAD 4.4 CÓMO ABRIR PYTHON (IDLE) Y EJECUTAR SU PRIMER PROGRAMA

Una vez instalado Python 3, puedes usar el editor integrado llamado IDLE para escribir y ejecutar tu código.

1. Abra IDLE

- En Windows: presione Inicio → IDLE (Python 3.x).
- En macOS: busque IDLE en Launchpad.
- En Linux: escriba idle3 en la terminal.

2. Crear un nuevo archivo

- En IDLE, haga clic en Archivo → Nuevo archivo (o presione Ctrl + N).
- Se abre una ventana en blanco donde puedes escribir tu programa.

3. Escribe o pega el código

Copie el código de este módulo en el nuevo archivo.
Por ejemplo:

```
CÓDIGO:
import pygame
pygame.init()
pantalla = pygame.display.set_mode((800, 600))
pygame.display.set_caption("Mi primer juego")
ejecutando = True
while ejecutando:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            ejecutando = False
    pantalla.fill((200, 220, 255))
    pygame.display.flip()
pygame.quit()
```

4. Guarde el archivo

Seleccione Archivo → Guardar como...
Nombra tu archivo (por ejemplo my_first_game.py)

5. Ejecute el programa

Presione F5 o vaya a Ejecutar → Ejecutar módulo.
Aparece una nueva ventana: si ves un fondo azul, todo está funcionando

UNIDAD 4.4 CÓMO ABRIR PYTHON (IDLE) Y EJECUTAR SU PRIMER PROGRAMA

Copia y ejecuta este código:

```
CÓDIGO:
import pygame  # 1. Importar la librería pygame
pygame.init()  # 2. Iniciar (inicializar) todos los módulos
de pygame
pantalla = pygame.display.set_mode((800, 600))  # 3. Crear
una ventana de juego (ancho 800, alto 600)
pygame.display.set_caption("Mi primer juego")  # 4. Dar un
título a la ventana del juego
ejecutando = True  # 5. Crear una variable para mantener el
juego en ejecución

while ejecutando:  # 6. Iniciar el bucle principal del juego
(se repite continuamente)
    for event in pygame.event.get():  # 7. Comprobar todos
los eventos (por ejemplo, teclas o clics del ratón)
        if event.type == pygame.QUIT:  # 8. Si el usuario
pulsa el botón de "cerrar"…
            ejecutando = False         # 9. …detener el bucle
del juego

    pantalla.fill((200, 220, 255))  # 10. Rellenar la
pantalla con un color azul claro (valores RGB)
    pygame.display.flip()  # 11. Actualizar la ventana para
mostrar el nuevo fotograma

pygame.quit()  # 12. Cerrar el juego y salir de forma segura
```

UNIDAD 4.5 CREA TU PRIMERA VENTANA DE JUEGO

1. Ahora, hagamos que algo se mueva. Copia el código a continuación y pruébalo:

```
PARTE I

CÓDIGO:
IMPORT PYGAME  # 1. IMPORTAR LA LIBRERÍA PYGAME
PYGAME.INIT()  # 2. INICIAR PYGAME

# 3. CREAR LA VENTANA
PANTALLA = PYGAME.DISPLAY.SET_MODE((800, 600))

# 4. AÑADIR UN TÍTULO
PYGAME.DISPLAY.SET_CAPTION("CUADRADO EN MOVIMIENTO")

X = 400  # 5. POSICIÓN X DEL CUADRADO (HORIZONTAL)
Y = 300  # 6. POSICIÓN Y DEL CUADRADO (VERTICAL)
VELOCIDAD = 1  # 7. CUÁNTOS PÍXELES SE MUEVE CADA VEZ

EJECUTANDO = TRUE  # 8. MANTENER EL JUEGO EN EJECUCIÓN

WHILE EJECUTANDO:  # 9. INICIAR EL BUCLE DEL JUEGO
    FOR EVENT IN PYGAME.EVENT.GET():  # 10. COMPROBAR EVENTOS
        IF EVENT.TYPE == PYGAME.QUIT:  # 11. SI EL USUARIO
PULSA “X”, DETENER
            EJECUTANDO = FALSE
```

UNIDAD 4.6 MOVER UN CUADRADO

1. Ahora, hagamos que algo se mueva. Copia el código a continuación y pruébalo:

```
PARTE II

CÓDIGO:
   teclas = pygame.key.get_pressed()  # 12. Comprobar qué
teclas están pulsadas
if teclas[pygame.K_LEFT]:
    x -= velocidad  # 13. Mover a la izquierda
if teclas[pygame.K_RIGHT]:
    x += velocidad  # 14. Mover a la derecha
if teclas[pygame.K_UP]:
    y -= velocidad  # 15. Mover hacia arriba
if teclas[pygame.K_DOWN]:
    y += velocidad  # 16. Mover hacia abajo

pantalla.fill((200, 220, 255))  # 17. Pintar el fondo
pygame.draw.rect(pantalla, (0, 0, 255), (x, y, 50, 50))  #
18. Dibujar el cuadrado azul
pygame.display.flip()  # 19. Actualizar la pantalla

pygame.quit()  # 20. Cerrar el juego
```

Lo que está sucediendo:

- x e y definen la posición del cuadrado en la pantalla.
- Las teclas de flecha cambian estos valores, moviendo el cuadrado.
- El bucle repite el dibujo muchas veces por segundo, creando un movimiento suave.

Prueba esto:

- Cambia la velocidad = 1 a la velocidad = 5. ¿Qué sucede?
- Haz que el cuadrado sea de un color diferente, por ejemplo (255, 0, 0) para rojo.
- ¿Qué crees que pasaría si eliminaras pygame.display.flip()?

Pregunta de reflexión:¿Cómo sabe la computadora la nueva posición del cuadrado cada vez que presionas una tecla?

Punto de control de aprendizaje

En este punto, deberías poder:

✔ Abra una ventana de Pygame.
✔ Utilice el bucle del juego para mantener el juego en ejecución.
✔ Controla un objeto en movimiento con el teclado.

En la siguiente parte, agregarás Inteligencia Artificial para que tu juego reaccione a tus movimientos.

UNIDAD 4.7 IA SIMPLE EN JUEGOS (ENEMIGO REACTIVO)

n este módulo, utilizamos una IA de juego simple basada en reglas: la computadora sigue reglas predefinidas para percibir al jugador, decidir qué hacer y actuar en consecuencia, sin aprender de los datos. Esto es similar a muchos videojuegos clásicos donde los enemigos persiguen, evitan o reaccionan al jugador según la distancia o la posición.

Meta: Aprende a hacer que tu juego reaccione a las acciones del jugador añadiendo una forma sencilla de Inteligencia Artificial (IA). Crearás un enemigo que cambia de color cuando el jugador se acerca.

Paso 1 – Comprender la idea
Antes de codificar, piense en lo que significa IA en un juego.
En este caso, la IA no se trata de aprendizaje ni de datos: se trata de hacer que la computadora reaccione a lo que hace el jugador.

Utilizarás una regla sencilla: Si el jugador está cerca, el enemigo cambia de color. Si no, permanece igual.

Este es un ejemplo de comportamiento condicional, una de las formas más simples de IA.
Paso 2 – Ejemplo de código: Enemigo reactivo
Copie y ejecute el siguiente código:

```
PARTE I

CÓDIGO:
import pygame, math      # 1. Importar librerías del juego y
matemáticas
pygame.init()            # 2. Iniciar (inicializar) todas las
funciones de Pygame

pantalla = pygame.display.set_mode((800, 600))
# 3. Crear una ventana de juego (800x600)
pygame.display.set_caption("IA simple - Enemigo reactivo")
# 4. Título

jugador = pygame.Rect(100, 100, 50, 50)
# 5. Jugador (cuadrado azul)
enemigo = pygame.Rect(500, 300, 50, 50)
# 6. Enemigo (cuadrado rojo)
velocidad = 5                # 7. Velocidad del jugador
distancia_reaccion = 120
# 8. Distancia (px) para que el enemigo "reaccione"

reloj = pygame.time.Clock()
ejecutando = True           # 9. Mantener el juego en
ejecución
```

PARTE II

CÓDIGO:

```
while ejecutando:          # 10. Bucle principal del juego
(cada fotograma)
    for event in pygame.event.get():  # 11. Comprobar todos
los eventos
        if event.type == pygame.QUIT:  # 12. Si se cierra la
ventana…
            ejecutando = False         # 13. …detener el
bucle

    # ---- MOVIMIENTO (debe estar dentro del while) ----
    teclas = pygame.key.get_pressed()  # 14. ¿Qué teclas
están pulsadas?
    if teclas[pygame.K_LEFT]:
        jugador.x -= velocidad         # 15. Mover a la
izquierda
    if teclas[pygame.K_RIGHT]:
        jugador.x += velocidad         # 16. Mover a la
derecha
    if teclas[pygame.K_UP]:
        jugador.y -= velocidad         # 17. Mover hacia
arriba
    if teclas[pygame.K_DOWN]:
        jugador.y += velocidad         # 18. Mover hacia
abajo

    # ---- IA SIMPLE: cambiar color si está cerca ----
    px, py = jugador.center      # 19. Centro del jugador
    ex, ey = enemigo.center      # 20. Centro del enemigo
    dist = math.hypot(px - ex, py - ey)  # 21. Distancia

    if dist < distancia_reaccion:      # 22. Si está a menos
de 120 px…
        color_enemigo = (255, 200, 0)  # 23. …el enemigo se
vuelve amarillo
    else:
        color_enemigo = (220, 60, 60)  # 24. Si no, el
enemigo sigue rojo
```

```
PARTE III

CÓDIGO:
# ---- DIBUJAR ----
pantalla.fill((240, 245, 255))  # 25. Fondo
pygame.draw.rect(pantalla, (60, 120, 255), jugador)
# 26. Jugador (azul)
pygame.draw.rect(pantalla, color_enemigo, enemigo)
# 27. Enemigo (reactivo)
pygame.display.flip()  # 28. Mostrar cambios

reloj.tick(60)  # Limitar a 60 FPS

pygame.quit()  # 29. Cerrar de forma segura
```

Paso 3: ¿Qué está sucediendo en este código?

Concepto	Explicación
Percepción	El enemigo mide qué tan lejos está del jugador usando math.hypot.
Decisión	Si la distancia es menor que react_distance, el enemigo decide reaccionar.
Acción	La reacción es cambiar su color.

Este ciclo “percibir → decidir → actuar” es la base de la mayoría de los sistemas de IA de juegos.
Incluso reacciones simples como ésta hacen que el mundo del juego se sienta más vivo.

PASO 4 – PROBAR, CAMBIAR, OBSERVAR

Intente cambiar estas partes del código y observe qué sucede:

1. Cambiar react_distance de 120 a 250: ¿qué sucede?
2. Haz que el enemigo se vuelva verde (0, 255, 0) en lugar de amarillo
3. Haz que el jugador se mueva más rápido aumentando la velocidad = 8.
4. ¿Puedes hacer que el enemigo reaccione solo cuando el jugador se mueve por encima o por debajo de él (por ejemplo, comparar solo las posiciones y)?

Anime a los estudiantes a predecir antes de ejecutar el programa.
—¿Qué creen que va a pasar?

Paso 5 – Reflexión

¿Es esto realmente "inteligencia"?
¿Por qué o por qué no?
¿Cómo podríamos hacer que el comportamiento del enemigo parezca más inteligente?

Pista: La IA real en los juegos utiliza la misma lógica: decisiones simples repetidas muchas veces por segundo pueden crear la ilusión de inteligencia.

REFERENCIAS

Libro, Wiki y Documentación

- Fundación del Software Python. Documentación de Python 3. Disponible en:https://docs.python.org

- Comunidad de Pygame. Documentación y tutoriales de Pygame. Disponible en:https://www.pygame.org/docs/ yhttps://www.pygame.org/wiki/tutorials

- Proyecto Python Arcade. Documentación de la biblioteca Python Arcade. Disponible en: https://api.arcade.academy

- Python real. Pygame: Introducción a la programación de juegos en Python. Disponible en: https://realpython.com/pygame-a-primer/

- Microsoft. Plan de estudios de IA para principiantes. Disponible en:https://github.com/microsoft/IA-Para-Principiantes

ENLACES A PLATAFORMAS DE APRENDIZAJE

- Sitio web oficial de Python: descarga y aprende los fundamentos de la programación en Python (https://www.python.org/)

- Documentación de Pygame: Guía oficial de funciones, eventos y gráficos de Pygame. (https://www.pygame.org/docs/)

- Documentación de la biblioteca Arcade: una biblioteca alternativa para juegos 2D sencillos en Python. (https://api.arcade.academy/en/latest/)

- Guía de IA de Pygame: ejemplos de algoritmos de IA sencillos para Pygame. (https://pygame-ai.readthedocs.io/en/latest/guide.html)

- Python Real: Crea un juego con Pygame: un tutorial fácil de usar para principiantes. (https://realpython.com/pygame-a-primer/)

- Tutoriales de KidsCanCode: Proyectos y desafíos sencillos para aprender Pygame. (https://kidscancode.org/blog/)

- IA para principiantes (Microsoft): Contenido educativo abierto que presenta los principios de la IA. (https://microsoft.github.io/AI-For-Beginners/)

EJERCICIO PRÁCTICO

ATRÁPAME SI PUEDES

Meta: Crea un juego corto donde el jugador mueve un cuadrado azul para recoger objetos mientras un enemigo rojo de la IA lo sigue. Este ejercicio combina todas las habilidades aprendidas: movimiento, detección de colisiones y un comportamiento simple de la IA.

Paso 1 – Prepare su entorno
Asegúrese de que Pygame esté instalado:

pip install pygame

Luego abra su editor de Python y cree un nuevo archivo llamado catch_me.py.

Paso 2: Comprender la idea del juego

Elemento	Description
Jugador	Se mueve con las teclas de flecha.
Objetivo	Aparece aleatoriamente; aumenta la puntuación cuando se recolecta.
Enemigo (IA)	Sigue al jugador calculando la dirección y la distancia.
Meta	Recoge tantos objetivos como puedas antes de que el enemigo te

Piensa antes de codificar:¿Cómo podría un enemigo “saber” dónde está el jugador?

PASO 3 – CONSTRUYE EL JUEGO PASO A PASO

PARTE I

CÓDIGO:

```
import pygame, random, math
pygame.init()

# --- Ventana ---
ANCHO, ALTO = 800, 600
pantalla = pygame.display.set_mode((ANCHO, ALTO))
pygame.display.set_caption("Atrápame si puedes")
reloj = pygame.time.Clock()

# --- Colores ---
AZUL = (60, 120, 255)
ROJO = (220, 60, 60)
VERDE = (60, 200, 60)
BLANCO = (240, 245, 255)
NEGRO = (20, 20, 20)

# --- Entidades ---
jugador = pygame.Rect(100, 100, 40, 40)
# cuadrado azul (jugador)
enemigo = pygame.Rect(600, 400, 40, 40)
# cuadrado rojo (enemigo que persigue)
objetivo = pygame.Rect(
    random.randint(50, ANCHO - 50),
    # objetivo verde: posición aleatoria
    random.randint(50, ALTO - 50), 25, 25
)
```

PASO 3 – CONSTRUYE EL JUEGO PASO A PASO

PARTE II

CÓDIGO:

```
VELOCIDAD_JUGADOR = 5
VELOCIDAD_ENEMIGO = 2.5
puntuacion = 0

# Fuente (compatible: usa la fuente por defecto para evitar
problemas)
fuente = pygame.font.Font(None, 26)

# --- Bucle del juego ---
ejecutando = True
while ejecutando:
   # 1) Manejar eventos
   for event in pygame.event.get():
       if event.type == pygame.QUIT:
           ejecutando = False

   # 2) Movimiento del jugador (flechas del teclado)
   teclas = pygame.key.get_pressed()
   if teclas[pygame.K_LEFT]:
       jugador.x -= VELOCIDAD_JUGADOR
   if teclas[pygame.K_RIGHT]:
       jugador.x += VELOCIDAD_JUGADOR
   if teclas[pygame.K_UP]:
       jugador.y -= VELOCIDAD_JUGADOR
   if teclas[pygame.K_DOWN]:
       jugador.y += VELOCIDAD_JUGADOR
```

PASO 3 – CONSTRUYE EL JUEGO PASO A PASO

PARTE III

CÓDIGO

```
# Mantener al jugador dentro de los límites de la ventana
jugador.clamp_ip(pantalla.get_rect())

   # 3) IA simple: el enemigo persigue al jugador
dx = jugador.centerx - enemigo.centerx
dy = jugador.centery - enemigo.centery
dist = math.hypot(dx, dy)
if dist != 0:  # evitar división por cero cuando se
superponen
    enemigo.x += int((dx / dist) * VELOCIDAD_ENEMIGO)
    enemigo.y += int((dy / dist) * VELOCIDAD_ENEMIGO)

# 4) El jugador recoge el objetivo
if jugador.colliderect(objetivo):
    puntuacion += 1
    objetivo.topleft = (
        random.randint(25, ANCHO - 50),
        random.randint(25, ALTO - 50)
    )

# 5) Dibujar todo
pantalla.fill(BLANCO)
pygame.draw.rect(pantalla, AZUL, jugador)
pygame.draw.rect(pantalla, ROJO, enemigo)
pygame.draw.rect(pantalla, VERDE, objetivo)
```

PASO 3 – CONSTRUYE EL JUEGO PASO A PASO

```
PARTE IV

CÓDIGO
# Dibujar puntuación
texto_puntuacion = fuente.render(f"Puntuación: {puntuacion}",
True, NEGRO)
pantalla.blit(texto_puntuacion, (10, 10))

pygame.display.flip()
reloj.tick(60)

pygame.quit()
```

Paso 4 – Experimentar y reflexionar
Intente cambiar estos valores y observe lo que sucede:

- Cambia ENEMY_SPEED para hacer que el enemigo sea más rápido o más lento.
- Cambia el color del jugador o del objetivo.
- Haz el juego más difícil reduciendo el tamaño del jugador.
- Añade un mensaje de Juego terminado cuando el enemigo toca al jugador.

Preguntas de reflexión

Tómate un tiempo para reflexionar profundamente sobre lo aprendido en este módulo. Responde las siguientes preguntas con atención:

- **Pregunta de reflexión 1:** ¿Cuál fue la parte más difícil de crear tu juego?
- **Pregunta de reflexión 2:** ¿Cómo hace la IA para que el juego se sienta más interactivo?
- **Pregunta de reflexión 3:** ¿Puedes pensar en otras situaciones en las que la IA reacciona a las acciones humanas (por ejemplo, educación o atención médica)?
- **Pregunta de reflexión 4:** Si pudieras mejorar este juego, ¿qué tipo de comportamiento de IA agregarías a continuación?
- **Pregunta de reflexión 5:** ¿Qué hace que el enemigo parezca “inteligente”?
- **Pregunta de reflexión 6:** ¿Cómo ayuda el cálculo de la distancia (math.hypot) a encontrar al jugador?
- **Pregunta de reflexión 7:** ¿Puedes pensar en juegos reales que utilicen un tipo de lógica similar?

CUESTIONARIO CON PREGUNTAS DE OPCIÓN MÚLTIPLE (una respuesta correcta por pregunta)

1. ¿Qué hace el bucle de juego en Pygame?
a) Instala nuevas bibliotecas durante el juego.
b) Repite las acciones del juego muchas veces por segundo.
c) Cierra la ventana del juego.
d) Carga la imagen de fondo solo una vez

2. ¿Cuál es el propósito de pygame.display.flip()?
a) Comprueba la entrada del teclado
b) Dibuja un rectángulo
c) Actualiza la ventana del juego con nuevas imágenes.
d) Pausa el bucle del juego.

3. ¿Qué función verifica qué teclas del teclado se están presionando?
a) pygame.quit()
b) pygame.key.get_pressed()
c) pygame.event.get()
d) pygame.display.set_mode()

4. En el ejemplo de IA, ¿qué hace que el enemigo reaccione ante el jugador?
a) Un evento de temporizador
b) Movimiento aleatorio
c) La distancia entre el jugador y el enemigo
d) El color del jugador

5. ¿Qué sucede si aumenta el valor de react_distance en el código?
a) El enemigo reacciona desde más lejos
b) El enemigo se mueve más rápido
c) El jugador disminuye la velocidad
d) El tamaño de la ventana cambia

CUESTIONARIO CON PREGUNTAS DE OPCIÓN MÚLTIPLE (una respuesta correcta por pregunta)

6. ¿Cuál es el propósito de la función math.hypot() en el código de IA?
a) Calcular la distancia entre dos puntos
b) Dibujar al enemigo en la pantalla.
c) Para crear una nueva posición aleatoria
d) Para detectar colisiones con paredes

7. ¿Qué hace esta línea de código?
a) Dibuja un texto en la pantalla.
b) Crea un rectángulo para representar al jugador.
c) Comienza con el algoritmo de IA
d) Cambia el color de fondo.

8. ¿Qué sucede si eliminas pygame.display.flip() del bucle?
a) El programa se ejecuta más rápido
b) La ventana no se actualiza y se ve congelada
c) Los colores cambian automáticamente
d) El jugador se mueve el doble de rápido

9. ¿Cuál de estos es un ejemplo de comportamiento de la IA en un juego?
a) Dibujar un cuadrado en la pantalla
b) Hacer que un personaje se mueva solo cuando el jugador presiona una tecla
c) Hacer que un enemigo siga o evite al jugador
d) Cargando música de fondo

10.¿Cómo puede la IA hacer que los juegos sean más interesantes?
a) Corrigiendo errores automáticamente
b) Creando desafíos dinámicos y personajes receptivos
c) Cambiando los gráficos automáticamente
d) Controlando el teclado

EJECUTAR EL CÓDIGO DEL MÓDULO 4

Programación de juegos en Python (pygame): Guía de configuración para la versión correcta de Python

El Módulo 4 utiliza la biblioteca pygame. Por el momento, pygame ofrece paquetes de instalación listos solo hasta Python 3.13, todavía no para Python 3.14. Si el ordenador ejecuta Python 3.14 (por ejemplo 3.14.6), la instalación de pygame falla y los ejemplos del Módulo 4 no se inician. Por eso el módulo está escrito para Python 3.13.3.

IImportante – no sobrescribir el Python existente

No desinstalar ni «rebajar» el Python que ya está en el ordenador. En macOS y Linux el sistema operativo depende de él y otros programas pueden dejar de funcionar. La solución segura es instalar Python 3.13 en paralelo y usarlo solo para este módulo. Cada una de las opciones siguientes hace exactamente eso.

Primero, comprobar la versión de Python

Abrir un terminal (macOS/Linux) o PowerShell (Windows) y escribir uno de estos comandos:

```
python --version
py --version
```

- **Python 3.13.x** → todo listo; el Módulo 4 puede ejecutarse directamente.
- **Python 3.14.x** → •seguir una de las opciones siguientes.

Opción 1 – macOS / Linux (Terminal)

El script auxiliar lo hace todo: instala una pequeña herramienta llamada uv, proporciona Python 3.13 junto al Python existente y ejecuta el archivo con pygame. No se cambia nada de forma permanente en el sistema.

Pasos:

1. Guardar el script de la página siguiente como modul4-starten.sh en la misma carpeta que el archivo Python del Módulo 4.
2. Hacerlo ejecutable y ejecutarlo (sustituir modul4.py por el nombre del propio archivo):

```
chmod +x modul4-start.sh
./modul4-starr.sh modul4.py
```

3. La primera ejecución descarga uv y Python 3.13 automáticamente. Cada ejecución posterior se inicia de inmediato.

El script

Copiar todo el contenido del recuadro de abajo en un archivo llamado modul4-starten.sh. (Junto a esta guía se incluye un archivo listo para usar.)

```
#!/usr/bin/env bash
#
==============================================================================
# modul4-start.sh
#
# Runs the Module 4 script (pygame) on Python 3.13 - WITHOUT touching or
# overwriting the Python that is already installed on your computer.
#
# Usage:
# chmod +x modul4-start.sh # make it executable (once)
# ./modul4-start.sh # runs ./modul4.py
# ./modul4-start.sh path/to/script.py
#
# If your module also needs the Arcade library, add --with arcade
# to the last command (see step 4).
#
==============================================================================
```

```
set -euo pipefail

SCRIPT="${1:-modul4.py}"

# --- 1. Install uv if it is not present yet ------------------------------
---
# uv is a small tool that manages Python versions and packages.
if ! command -v uv >/dev/null 2>&1; then
 echo "-> Installing 'uv' (one-time setup) ..."
 curl -LsSf https://astral.sh/uv/install.sh | sh
 # uv is placed in ~/.local/bin - make it available in this session:
 export PATH="$HOME/.local/bin:$PATH"
fi

# --- 2. Provide Python 3.13 ----------------------------------------------
---
# It is installed IN PARALLEL. Your existing Python (3.14.x) stays
untouched.
echo "-> Providing Python 3.13 (if not already installed) ..."
uv python install 3.13

# --- 3. Does the script exist? -------------------------------------------
---
if [[ ! -f "$SCRIPT" ]]; then
 echo "x File not found: $SCRIPT"
 echo " Example: ./modul4-start.sh path/to/script.py"
 exit 1
fi

# --- 4. Run the script on Python 3.13 with pygame ------------------------
---
# uv creates a short-lived, isolated environment, installs pygame into it,
# and runs the script. Nothing is changed globally.
echo "-> Starting '$SCRIPT' on Python 3.13 with pygame ..."
uv run --python 3.13 --with pygame "$SCRIPT"
```

Opción 2 – Windows (PowerShell)

En Windows no hace falta un script de shell; bastan tres comandos. Abrir PowerShell en la carpeta que contiene el archivo Python y ejecutarlos uno tras otro (sustituir modul4.py por el nombre del propio archivo):

```
powershell -c "irm https://astral.sh/uv/install.ps1 | iex"
uv python install 3.13
uv run --python 3.13 --with pygame .\modul4.py
```

Opción 3 – Seguir con Python 3.14 (sin un segundo Python)

Si se prefiere mantener Python 3.14, se puede usar pygame-ce («edición comunitaria»), un reemplazo casi idéntico que ya incluye paquetes para Python 3.14. El código no cambia; se sigue escribiendo import pygame.

```
pip uninstall pygame
pip install pygame-ce
```

Recomendación

Para los materiales didácticos compartidos, mantener la configuración con Python 3.13 (Opciones 1 y 2). El módulo está documentado y probado con Python 3.13.3 y la biblioteca pygame estándar. Usar pygame-ce solo como prueba personal rápida.

Resolución de problemas

- command not found: uv (macOS/Linux) – cerrar y volver a abrir el terminal, o ejecutar export PATH="$HOME/.local/bin:$PATH".
- uv is not recognized (Windows) – cerrar y volver a abrir PowerShell después de instalar uv.
- «Archivo no encontrado» – asegurarse de indicar el nombre o la ruta correctos, por ejemplo ./modul4-starten.sh games/my_game.py.
- El módulo también usa la biblioteca Arcade – añadir --with arcade al último comando, por ejemplo uv run --python 3.13 --with pygame --with arcade modul4.py.
- Se abre una ventana del juego y se cierra de inmediato – normalmente significa que el propio código Python tiene un error; el mensaje aparece en el terminal.

MÓDULO N° 5|Reconocimiento de objetos con Roboflow: Introducción a la visión artificial

INTRODUCCIÓN

El objetivo del módulo actual es proporcionar una introducción completa a la visión artificial, explicando cómo las máquinas perciben y procesan los datos visuales y guiar a los estudiantes en el desarrollo de sus propios modelos de reconocimiento de objetos utilizando la plataforma Roboflow y algoritmos avanzados como YOLO.

Al finalizar el módulo podrás adquirir diferentes habilidades, tales como:

- **Comprensión de los fundamentos:**Distinguir entre procesamiento de imágenes y visión por computadora, y comprender la lógica de cómo las computadoras interpretan las imágenes como datos (píxeles y código binario).
- **Aplicación del proceso CV:**Aplicar los cinco pasos esenciales del proceso de visión artificial: obtener datos, preparar datos, determinar el modelo de IA, entrenar el modelo e interpretar los resultados.
- **Usando Roboflow:** Navegue por la plataforma Roboflow para crear proyectos, cargar conjuntos de datos y administrar datos de imágenes de manera eficaz.
- **Anotación de datos:**Realice una anotación de datos precisa (etiquetado) para preparar conjuntos de datos para el entrenamiento del modelo de IA.
- **Entrenamiento de modelos:**Seleccione y entrene modelos avanzados de detección de objetos, como YOLOv11, para identificar objetos específicos
- **Implementación práctica:**Ejecute un ciclo de proyecto completo recopilando datos del mundo real (fotografías), entrenando un modelo y probando su precisión en un entorno práctico.

Duración del módulo	2 horas (1 hora de aprendizaje + 1 hora de ejercicios prácticos)

MATERIALES EDUCATIVOS

UNIDAD 5.1 ¿QUÉ ES EL PROCESAMIENTO DE VISIÓN POR COMPUTADORA?

El procesamiento de imágenes es una técnica que se utiliza para detectar, extraer y evaluar información importante contenida en una imagen digital. Este método proporciona datos significativos sobre los objetos o entornos de una imagen, similar a la percepción visual humana. El objetivo principal del procesamiento de imágenes es extraer significado de las imágenes y presentar esta información para su uso en diversos campos.
Si bien la tecnología de procesamiento de información visual existe desde hace mucho tiempo, en el pasado este proceso dependía en gran medida de la intervención humana. Esto requería mucho tiempo y era propenso a errores. Por ejemplo, en los sistemas de reconocimiento facial anteriores, los desarrolladores tenían que etiquetar manualmente miles de fotos; características específicas, como el ancho de la nariz o la distancia entre los ojos, se identificaban una por una. Esto se debía a que los datos de las imágenes solían estar desorganizados y eran difíciles de comprender para las computadoras. Por lo tanto, automatizar el proceso requería procesadores muy potentes y tecnologías informáticas avanzadas.

Hoy en día, con la ayuda de la evolución del poder de procesamiento, las aplicaciones de visión por computadora utilizan inteligencia artificial y aprendizaje automático (IA/ML) para procesar con precisión estos datos para la identificación de objetos y el reconocimiento facial, así como la clasificación, recomendación, seguimiento y percepción.
La visión artificial puede describirse simplemente como "permitir que las máquinas realicen esta tarea", inspirada en el sistema visual humano y su relación con el cerebro. Es el proceso mediante el cual las máquinas, como los ordenadores, perciben imágenes y vídeos, los analizan y les dan significado mediante el aprendizaje automático y la inteligencia artificial. Originaria de la década de 1950, la visión artificial comenzó a comercializarse en la década de 1970 con la aplicación de la detección de la diferencia entre la escritura a máquina y la escritura a mano. Hoy en día, se utiliza en diversas áreas, como las tecnologías de detección y reconocimiento de personas, los sistemas de control de calidad de productos, los vehículos autónomos, las aplicaciones agrícolas, la sanidad, la educación y la defensa.
La visión artificial y el procesamiento de imágenes son campos relacionados, pero distintos.

El procesamiento de imágenes implica procesar, restaurar y ajustar el color y el brillo de los datos de imágenes capturadas, incluyendo fotos y vídeos. La visión artificial, por otro lado, consiste en la aplicación de técnicas de procesamiento de imágenes a las imágenes y la combinación de este proceso con la inteligencia artificial. Por ejemplo, mientras que el procesamiento de imágenes se utiliza para encontrar los bordes de un objeto dentro de una imagen, el uso de técnicas de aprendizaje automático para identificar y clasificar dicho objeto se enmarca en el campo de la visión artificial.

UNIDAD 5.2 ¿CÓMO VE UNA COMPUTADORA?

Los componentes básicos de casi todas las computadoras del mundo, sus "células", están formadas por los números 1 y 0. Esto significa que las computadoras perciben los datos que los usuarios quieren transmitirles en su propio lenguaje: 1 y 0.

Al fin y al cabo, las imágenes y los vídeos, que constan de muchas imágenes, también son datos, por lo que las computadoras los perciben como ceros y unos, o como el sistema binario. Para ilustrarlo mejor, podemos usar la imagen a continuación.

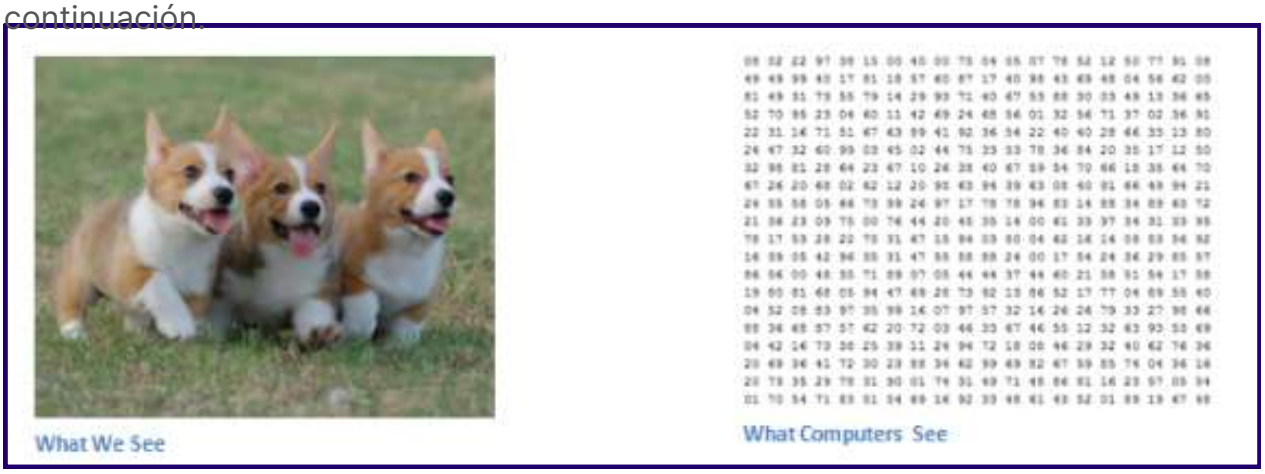


Figura 7 Forma de vista: Uso de redes neuronales para videovigilancia (2025) de https://www.videonet9.com/using-neural-networks-for-video-surveillance.html; Fecha de publicación: 09.08.2025.

Pero si te preguntas si esta imagen muestra valores de dos dígitos, pero no hay ceros ni unos, entonces sí, es cierto. Nuestros procesadores interpretan estos valores convirtiéndolos en ceros y unos. Si estos valores también se mostraran como ceros y unos, la imagen sería muy larga y antiestética. De hecho, cada uno de estos valores de dos dígitos en la imagen de arriba se puede llamar un píxel.

Entonces, ¿qué es un píxel? Es la unidad más pequeña que permite generar y controlar la imagen en las pantallas digitales.

UNIDAD 5.3 AHORA HAGAMOS UN EJERCICIO

- Tome un cuaderno cuadriculado.
- Numera cada cuadrado de la cuadrícula del cuaderno con números entre 0 y 1.
- Cuanto más cerca esté el número escrito en cualquier cuadrado del 1, más blanco será el cuadrado.
- Cuanto más cerca esté el número de 0, más negro será el cuadrado.
- Si el número es 0,5, colorea el cuadrado de gris.

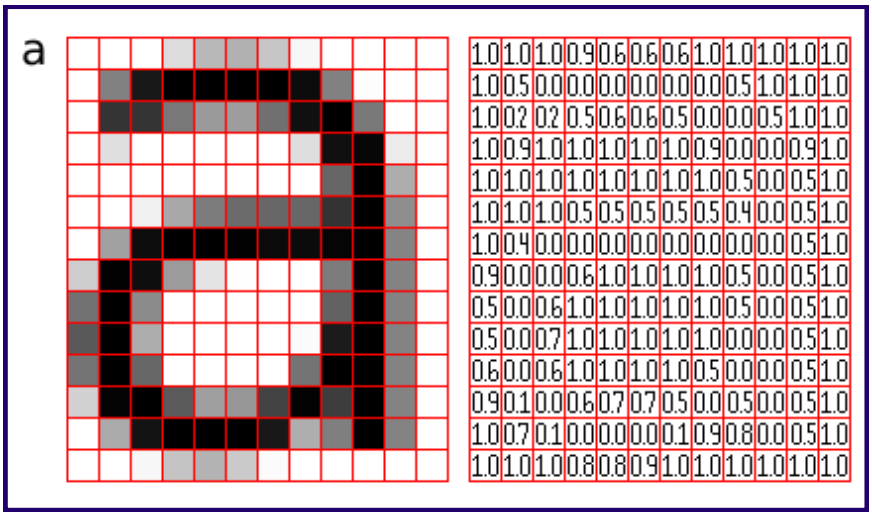


Figura 8. Cómo las computadoras ven los píxeles. (2025) de https://medium.com/@meriyananjalika99/from-pixels-to-predictions-getting-started-with-cnns-convolutional-neural-networks-e3f84781cc1e; Fecha de publicación: 09.08.2025.

Si dibujas con cuidado en un cuaderno como este, desde lejos te parecerá un dibujo normal. Prueba esto con la imagen de arriba. Aléjate unos metros de la pantalla. La imagen se verá más normal. De hecho, las pantallas de ordenador y televisión muestran imágenes y vídeos usando millones de cuadritos diminutos de esta manera. Para guardar las imágenes, almacenan los números en la memoria del cuaderno cuadriculado. En las imágenes de baja calidad que llamamos pixeladas, se pueden ver los cuadritos individuales en este cuaderno cuadriculado.

UNIDAD 5.4 PROCESOS DE VISIÓN POR COMPUTADORA

Los procesos de visión por computadora constan esencialmente de cinco pasos:

1. Obtención de la imagen/datos
2. Preparación de la imagen/datos
3. Determinación del modelo de IA a utilizar
4. Entrenamiento del modelo de IA a utilizar
5. Interpretación de los resultados

A continuación se muestran las imágenes obtenidas para una aplicación sencilla de reconocimiento de dígitos escritos a mano. Estas imágenes se convirtieron a una escala de grises compatible con computadoras y se utilizaron para entrenar la IA. Los resultados de las pruebas indican una precisión del 99 % tras el entrenamiento.

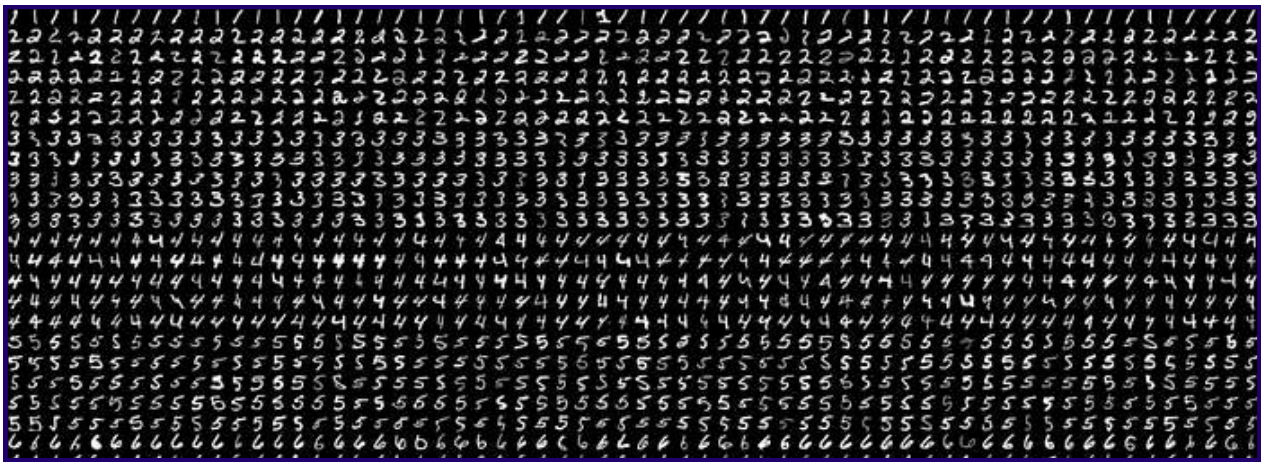


Figura 9 Conjunto de datos Kaggle (2025) de www.roboflow.com; fecha de adición: 09.08.2025.

Roboflow es una plataforma integral de inteligencia artificial diseñada para el desarrollo de proyectos de visión artificial. Simplifica el proceso de creación, entrenamiento e implementación de modelos de visión artificial mediante datos de imagen y vídeo. Ofrece una interfaz intuitiva tanto para principiantes como para desarrolladores experimentados.

UNIDAD 5.5 ¿CÓMO REGISTRARSE EN ROBOFLOW?

• Vaya a https://roboflow.com/.

• Haga clic en el botón Iniciar sesión en la esquina superior derecha.

- En la ventana que se abre, ingrese su información de membresía, si la tiene, e inicie sesión en el sistema. Si inicia sesión por primera vez, use la opción "Continuar con Google", ya que es más práctico continuar con una cuenta de Google.

• Después de seleccionar la cuenta de Google que desea utilizar en el sitio de Roboflow y otorgar los permisos necesarios, iniciará sesión.

• Haga clic en el botón NUEVO PROYECTO en la página que se abre, ingrese la información del proyecto y cree el proyecto.

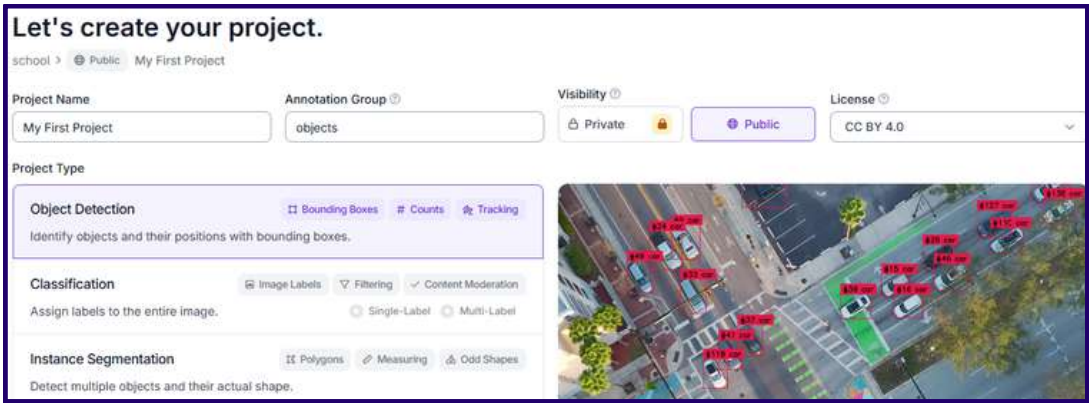


Figura 10 Conjunto de datos de Kaggle (2025) de www.roboflow.com; fecha de adición: 09.08.2025.

● En la nueva página que se abre, agregamos los datos de entrenamiento, validación y prueba de nuestra IA. A partir de ahora, nos referiremos a estos datos como "conjunto de datos". Si no tenemos un conjunto de datos, podemos usar conjuntos de datos predefinidos dentro del sistema RoboFlow Universe (https://universe.roboflow.com).

De igual forma, Kaggle es un sitio web que ofrece conjuntos de datos predefinidos (https://www.kaggle.com/).

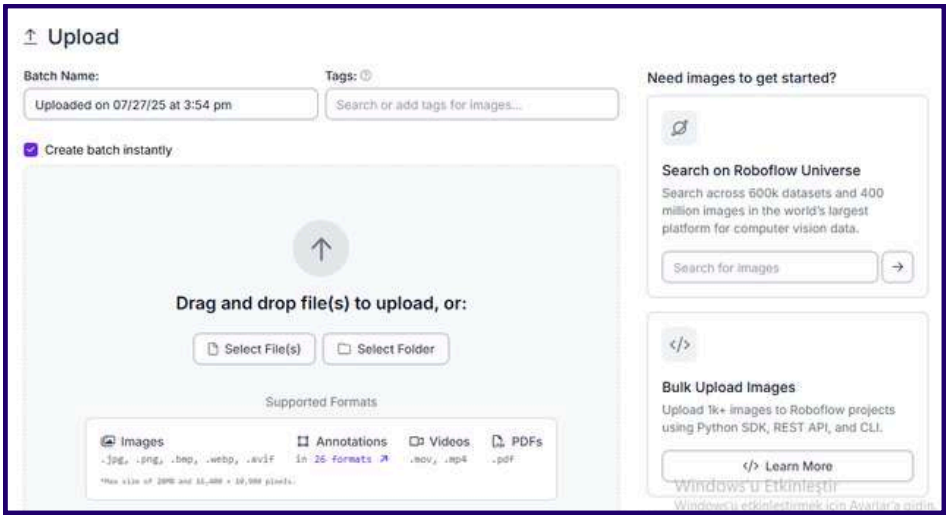


Figura 11 Roboflow (2025) de www.roboflow.com; Fecha de adición: 09.08.2025.

En este sistema, continuaremos los pasos del proceso utilizando un conjunto de datos prediseñado. Por lo tanto, usaremos el conjunto de datos disponible en https://universe.roboflow.com/roboflow-58fyf/rock-paper-scissors-sxsw a través de RoboFlow. Tras ingresar esta dirección, haga clic en el botón "Descargar conjunto de datos". Sin embargo, antes de hacer clic en este botón, debemos decidir qué modelo de IA usar. En esta aplicación, usaremos el modelo de IA YOLOv11, que es más eficiente en la detección de objetos. Si visita el perfil de GitHub del algoritmo correspondiente (https://github.com/ultralytics/ultralytics), encontrará diferentes pesos prediseñados e información detallada. (Prueba en línea de Yolov3: https://v-iashin.github.io/detector)

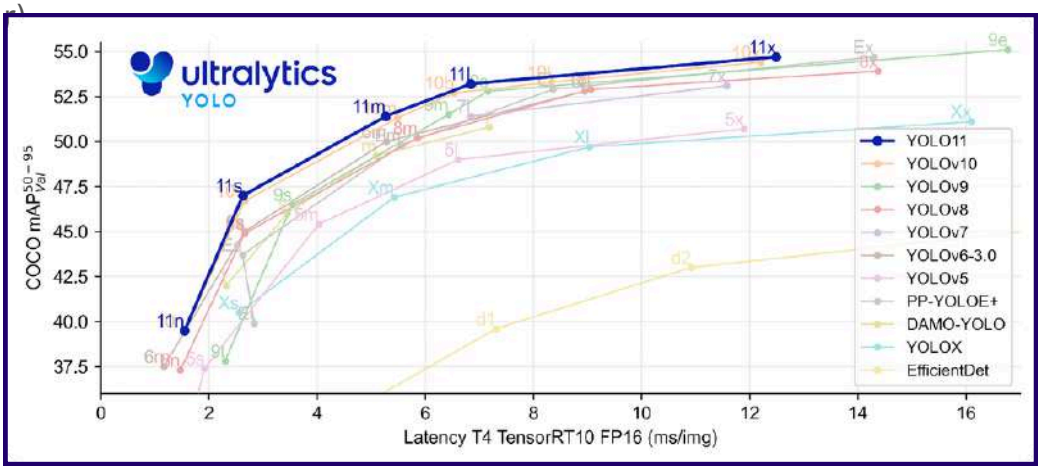


Figura 12 Ultralytics (2025) de https://github.com/ultralytics/ultralytics; Fecha de adición: 09.08.2025.

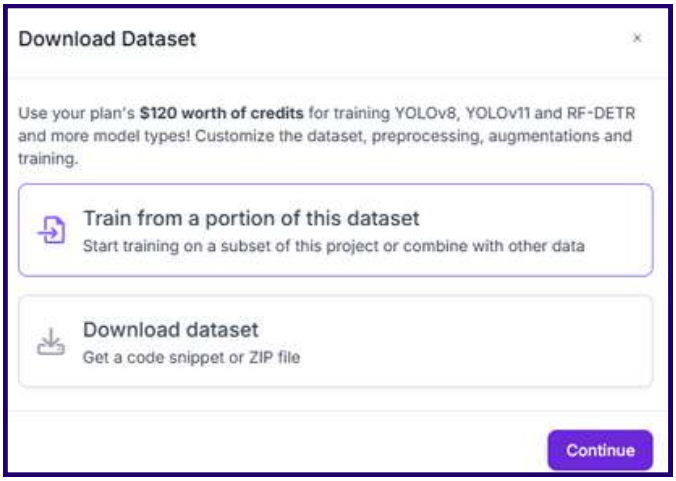


Figura 13 Conjunto de datos (2025) de www.roboflow.com; Fecha de adición: 09.08.2025.

Después de etiquetar y verificar los datos en el conjunto de datos, haga clic en el botón de descarga del conjunto de datos, seleccione el modelo y el método en línea, y obtenga el código Python proporcionado por el sitio.

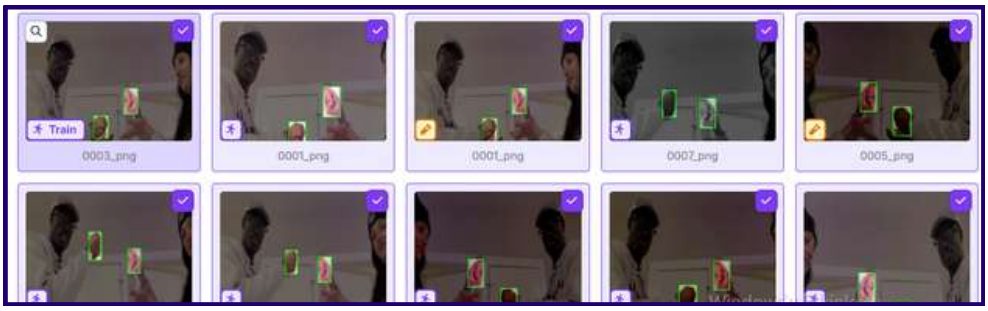


Figura 14 Corte (2025) de www.roboflow.com; Fecha de adición: 09.08.2025.

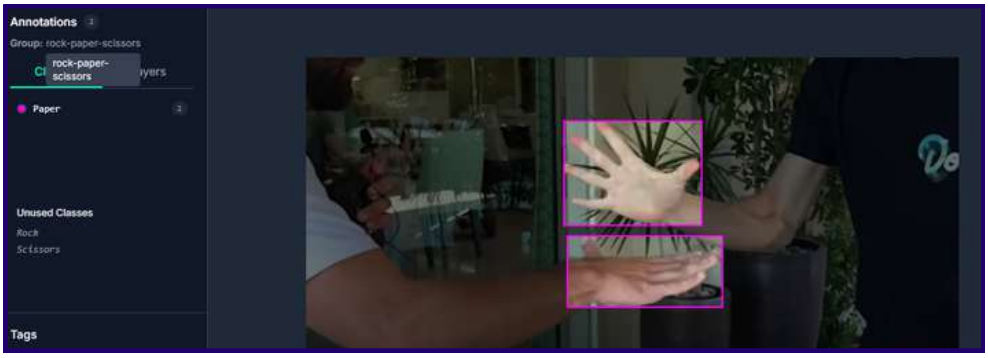


Figure 15 Cut-off (2025) from www.roboflow.com; DateAdd: 09.08.2025.

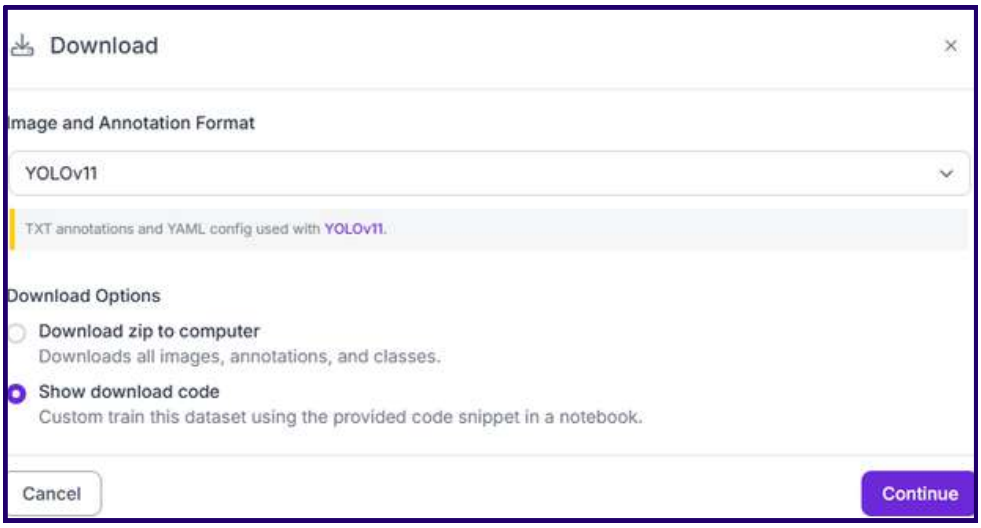
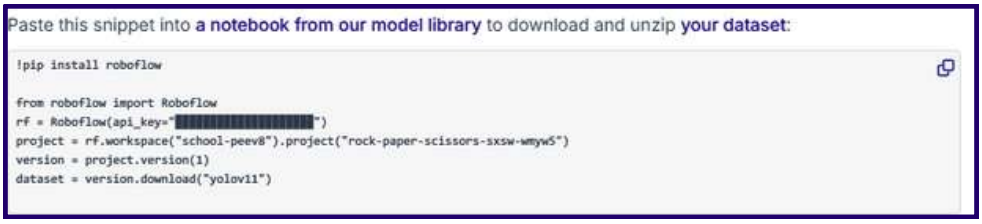


Figura 16 YOLOv11 (2025) de www.roboflow.com; Fecha de adición: 09.08.2025.



Fragmento de la figura 17 (2025) de www.roboflow.com; fecha de adición: 09.08.2025

En el siguiente paso, abriremos el código del algoritmo YOLOv11, que se ejecuta en Google Colab, preparado para nosotros por Roboflow. (https://blog.roboflow.com/yolov11-how-to-train-custom-data/)

Visite https://colab.research.google.com/github/roboflow-ai/notebooks/blob/main/notebooks/train-yolo11-object-detection-on-custom-dataset.ipynb?ref=blog.roboflow.com y siga las instrucciones allí.

Otra opción es crear un modelo y usarlo en Roboflow. Esto puede ser más rápido y sencillo, pero YOLOv11 ofrece resultados mejores y más precisos.
En los últimos años, dos arquitecturas de detección de objetos han revolucionado el mundo de la visión artificial: las redes neuronales convolucionales (CNN) y las de solo una mirada (YOLO). Tanto las CNN como las YOLO han contribuido a dar forma a la industria moderna de la visión artificial. Los transformadores también desempeñan un papel cada vez más importante en la visión artificial y la detección de objetos.

Las CNN utilizan convoluciones, una técnica de procesamiento de imágenes, para aprender las características de una imagen. Este proceso implica aplicar una ventana deslizante a cada píxel de la imagen para aprender las características. La información obtenida de las convoluciones es procesada por una red neuronal. Existen numerosas implementaciones de CNN, como R-CNN, Mask R-CNN y Fast R-CNN.
La familia de modelos YOLO también ha sido influyente en el mundo de la visión artificial. Introducida por Joseph Redmon en 2014, YOLO se ha convertido en un área activa de investigación y desarrollo, ha sido adoptada por una amplia comunidad y ha sido implementada por numerosos desarrolladores e investigadores. YOLOv5 y YOLOv8, desarrollados y perfeccionados por el equipo de Ultralytics, impulsan modelos de detección de objetos de producción a nivel mundial.

Puede encontrar más información sobre Yolo en:
https://blog.roboflow.com/guide-to-yolo-models

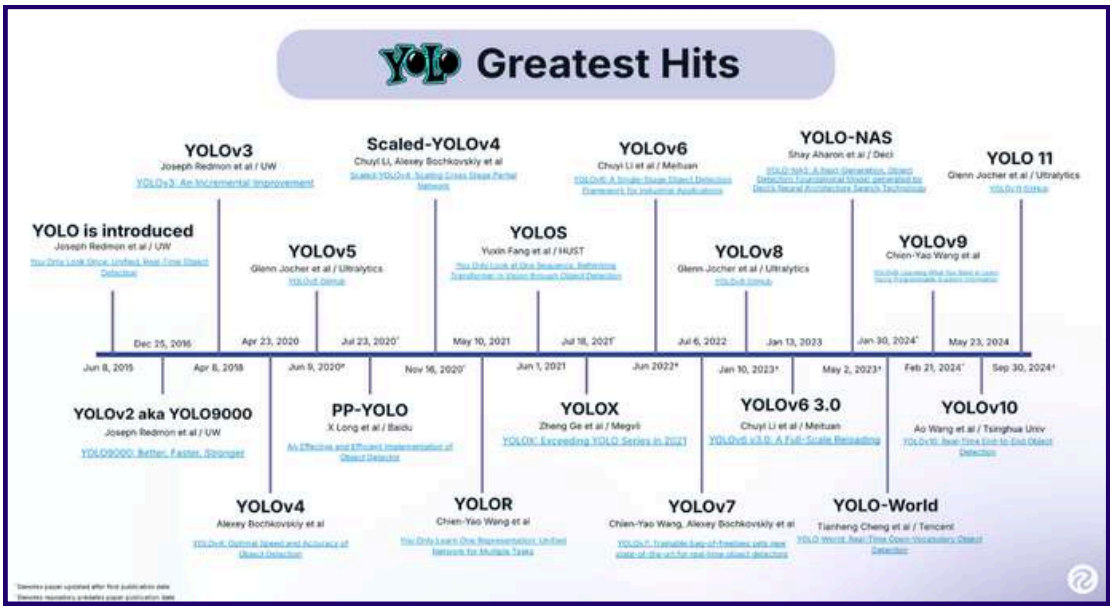


Figura 18 YOLO-Historia (2025) de www.roboflow.com; Fecha de adición: 09.08.2025.

Para analizar la clasificación detallada de los algoritmos utilizados activamente en el campo de la visión artificial en todo el mundo, puede visitar la dirección correspondiente: https://leaderboard.roboflow.com/?ref=blog.roboflow.com

REFERENCIAS

Libros y Documentación

- Banerjee, Chayan y Fookes, Clinton y Karniadakis, George. (2023). Visión artificial basada en la física: una revisión y perspectivas. 10.48550/arXiv.2305.18035.

- LeCun, Y. (2025). MNIST: una vista remota del conjunto de datos dehttp://yann.lecun.com/exdb/mnist/. Fecha de adición: 09.08.2025.

- Mohite, Amruta y Kulkarni, Atharva y Chitnis, Rutwik y Mane, Swapnil y Asabe, Shubham. (2021). Inspección con IA: Visión artificial para la inspección visual. Revista Internacional de Investigación Avanzada en Ciencias de la Computación y Gestión. 29 de julio.

- Reconocimiento de objetos (2025). De https://viso.ai/product/computer-vision-parking-lot-occupancy-tutorial/. Fecha de publicación: 10/08/2025.

- Uso de redes neuronales para videovigilancia (2025) de https://www.videonet9.com/using-neural-networks-for-video-surveillance.html. Fecha de publicación: 09/08/2025.

ENLACES A PLATAFORMAS DE APRENDIZAJE

Para analizar la clasificación detallada de los algoritmos utilizados activamente en el campo de la visión artificial en todo el mundo, puede visitar la dirección correspondiente: https://leaderboard.roboflow.com/?ref=blog.roboflow.com

Roboflow es una plataforma integral de inteligencia artificial diseñada para desarrollar proyectos de visión artificial.https://roboflow.com/.

EJERCICIO PRÁCTICO

CAZADORES DE OBJETOS – GUÍA DEL PROFESOR

Propósito de la actividad

• Comprender la lógica de la visión artificial.
• Creación de un modelo de reconocimiento de objetos utilizando Roboflow
• Experimentar los procesos de recopilación de datos, etiquetado, entrenamiento de modelos y pruebas.

Lista de preparación

• Computadoras o tabletas conectadas a Internet
• Grupos de 4-5 estudiantes
• Teléfonos inteligentes (para tomar fotografías)
• Cuenta gratuita de Roboflow
• Objetos simples (libro, lápiz, botella de agua, etc.)

Duración y horario

Parte 1: Introducción a la visión artificial – 15 minutos
Parte 2: Colección de fotografías – 20 minutos
Parte 3: Etiquetado – 25 minutos
Parte 4: Entrenamiento de modelos – 20 minutos
Parte 5: Prueba y competencia – 20 minutos

Pasos de implementación

1. Introduzca a los estudiantes el concepto de visión por computadora con una breve presentación.
2. Los grupos seleccionan dos objetos diferentes y toman entre 15 y 20 fotografías desde diferentes ángulos.
3. Inicie sesión en Roboflow y cree un nuevo proyecto.
4. Se cargan las fotos y se etiquetan los objetos mediante marcos de dibujo.
5. El modelo está entrenado y probado
6. Los grupos prueban los modelos de los demás y se otorgan puntuaciones.

CAZADORES DE OBJETOS – GUÍA DEL PROFESOR

Ideas de gamificación

- Carrera contra el tiempo: ¿Quién puede recopilar y etiquetar datos más rápido?
- Prueba mixta: el modelo se prueba con objetos que nunca ha visto.
- Búsqueda de errores: se encuentran y se discuten casos en los que el modelo es erróneo

CAZADORES DE OBJETOS – HOJA DE ACTIVIDADES PARA ESTUDIANTES

¿Qué es la visión por computadora?

La visión por computadora es el campo de la inteligencia artificial que permite a las computadoras detectar y comprender objetos utilizando cámaras o imágenes

Tus tareas

1. Como equipo, elijan dos objetos.
2. Tome entre 15 y 20 fotografías de cada objeto desde diferentes ángulos.
3. Abra un proyecto en Roboflow y cargue las fotos.
4. Etiqueta los objetos.
5. Entrena tu modelo.
6. Intenta reconocer los objetos de los otros equipos.

Consejos para fotografiar

- Dispara desde diferentes ángulos.
- Pruebe diferentes condiciones de iluminación.
- Fotografíe el sujeto tanto de cerca como de lejos.

Guía de etiquetado

Cuadra el objeto de la foto y escribe su nombre.
Ejemplo: Lápiz, Borrador

Mis observaciones

- Situaciones que mi modelo reconoce mejor.
- Situaciones con las que mi modelo tiene dificultades.

Preguntas de reflexión

Tómate un tiempo para reflexionar profundamente sobre lo aprendido en este módulo. Responde las siguientes preguntas con atención:

- **De los píxeles a la percepción:** En este módulo, aprendiste que las computadoras perciben las imágenes como cuadrículas de números y código binario, en lugar de formas y colores. ¿Cómo cambia esta visión "numérica" del mundo tu comprensión de cómo una IA distingue entre dos objetos diferentes, como un lápiz y una goma de borrar?

- **La importancia de la variedad de datos:** La actividad "Cazadores de objetos" se centró en tomar fotos desde diferentes ángulos y bajo diversas condiciones de iluminación. ¿Por qué es crucial esta diversidad en el conjunto de datos de entrenamiento para que la IA reconozca con precisión un objeto en un entorno real?

- **Análisis de precisión y errores:** Durante la fase de pruebas, se le animó a identificar errores del modelo (una "búsqueda de errores"). Según sus observaciones, ¿qué factores ambientales específicos (como las sombras o la distancia) dificultaron el correcto funcionamiento de su modelo?

- **Resolución de problemas reales:** La visión artificial ya se utiliza en campos como la salud, los vehículos autónomos y la agricultura. Ahora que has entrenado tu propio modelo, ¿qué problema específico de tu escuela o comunidad local crees que podría resolverse con esta tecnología?

CUESTIONARIO CON PREGUNTAS DE OPCIÓN MÚLTIPLE(una respuesta correcta por pregunta)

1. ¿Cuál es el objetivo principal de la visión por computadora?
a) Permitir que las computadoras piensen como cerebros humanos.
b) Permitir que las máquinas "vean" e interpreten el mundo como el ojo humano.
c) Almacenar grandes conjuntos de datos en bases de datos.
d) Analizar únicamente datos basados en texto.

2. ¿Cuál de las siguientes NO es una aplicación típica de la visión por computadora?
a) Reconocimiento de señales de tráfico en vehículos autónomos.
b) Sistemas de reconocimiento facial.
c) Analizar las preferencias de los usuarios en los sistemas de recomendación de productos.
d) Detección de tumores en imágenes médicas.

3.¿Cuál es la tarea de visión artificial que tiene como objetivo determinar la ubicación? ¿Y el tipo de objetos específicos en una imagen?
a) Clasificación de imágenes
b) Detección de objetos
c) Segmentación de imágenes
d) Extracción de características

4. ¿Cuál de los siguientes es uno de los desafíos de la visión por computadora? ¿A qué se enfrentan los sistemas en el mundo real?
a) La escasez de datos y la diversidad de los mismos.
b) La baja dimensionalidad y estructura simple de los datos visuales.
c) Consistencia en iluminación, posición y escala.
d) Modelos que obtienen consistentemente resultados perfectamente precisos.

CUESTIONARIO CON PREGUNTAS DE OPCIÓN MÚLTIPLE(una respuesta correcta por pregunta)

5. ¿Cuál de las siguientes afirmaciones indica que un modelo está sobreajustado?

a) Baja precisión tanto en los datos de entrenamiento como de prueba.
b) Precisión muy alta en los datos de entrenamiento, precisión baja en los datos de prueba.
c) Alta precisión tanto en los datos de entrenamiento como de prueba.
d) El modelo etiqueta cada imagen de la misma manera.

Resumen y próximos pasos

Basándose en los conocimientos básicos del primer curso, estos materiales de aprendizaje avanzado están diseñados para sumergir a los estudiantes de secundaria en la aplicación práctica de la inteligencia artificial. El contenido conecta la teoría con la práctica, guiando a los alumnos en la creación de modelos de IA sencillos y básicos con Python y Scratch. Desde la exploración de la lógica matemática tras la toma de decisiones en una "sala de escape de IA" hasta el entrenamiento de modelos de aprendizaje automático para personajes de videojuegos y el desarrollo de aplicaciones de visión artificial con Roboflow, los módulos ofrecen una experiencia práctica integral. Un enfoque narrativo, con personajes cercanos como Aylin y Alara, garantiza que conceptos técnicos complejos, como redes neuronales, agrupamiento K-means y reconocimiento de objetos, sigan siendo accesibles y atractivos.

Al completar este curso, los estudiantes pasarán de ser consumidores pasivos a creadores activos de tecnología. Adquirirán la competencia técnica para codificar sistemas inteligentes básicos, el pensamiento crítico para evaluar la precisión de los modelos y la conciencia ética para implementar la IA de forma responsable. Esta inmersión profunda en la programación y los primeros pasos en la ciencia de datos no solo consolida sus competencias STEM, sino que también los prepara para futuras carreras académicas y profesionales en un mundo impulsado por la tecnología.

¿Qué sigue?

Para apoyar la integración sostenible de estos temas avanzados en el aula, el proyecto presentará próximamente un "Kit de herramientas electrónicas para educadores escolares". Este recurso proporcionará a los docentes planes de clase detallados y orientación metodológica para impartir eficazmente la formación en IA. Además, se anima a los estudiantes a demostrar sus nuevas habilidades participando en los próximos concursos "Exploradores del Futuro de la IA", donde podrán aplicar sus conocimientos para resolver retos del mundo real y conectar con una comunidad más amplia de jóvenes innovadores.

Socios del proyecto

FUTURE-STEM-HUB

UNIVERSITÄT DUISBURG ESSEN
Open-Minded

KÜTAHYA İL MİLLÎ EĞİTİM MÜDÜRLÜĞÜ

tetra solutions

M&M profuture training

ETAP ESCOLA PROFISSIONAL

Coordinador del Proyecto:
University of Duisburg-Essen, Alemania

Correo electrónico
mustafa.bilgin@uni-due.de

Website
www.future-stem-hub.eu

Co-funded by the European Union

Financiado por la Unión Europea. Las opiniones y puntos de vista expresados son, sin embargo, responsabilidad exclusiva del/de los autor(es) y no reflejan necesariamente los de la Unión Europea ni los de la Agencia Ejecutiva Europea de Educación y Cultura (EACEA). Ni la Unión Europea ni la EACEA se responsabilizan de ellas.