



Inteligencia Artificial

Enseñando a las máquinas a aprender

En este proyecto los estudiantes van a programar un videojuego o aplicación en Scratch, integrando un módulo de inteligencia artificial desarrollado con Teachable Machine.

La construcción del videojuego supone un desafío de programación, que se suma a la dificultad adicional de tener que integrar un módulo específico de inteligencia artificial al videojuego. Los contenidos curriculares asociados serán definidos por el docente de la asignatura específica, de forma tal que la propuesta pedagógica le sirva para presentar la temática deseada.

La aplicación o videojuego propuesto involucra a un bloque de inteligencia artificial, que debe ser también desarrollado por los estudiantes. Este bloque será producto de la creatividad de los estudiantes y les permitirá un primer acercamiento para comprender a qué nos referimos cuando decimos que las computadoras aprenden o cuentan con inteligencia. En [este link encuentra un video](#) se encuentra una versión de ejemplo del videojuego para consulta de los docentes.

Duración: 8 etapas.

Materiales: [Teachable Machine](#) (Google), [PoseBlocks](#) (Versión de Scratch online para IA), Bitácora para estudiantes

Índice

Ficha Curricular ↓	2
Síntesis del recorrido	4
ETAPA 1 ↓	6
ETAPA 2 ↓	9
ETAPA 3 ↓	12
ETAPA 4 ↓	15
ETAPA 5 ↓	17
ETAPA 6 ↓	20
ETAPA 7 ↓	23
ETAPA 8 ↓	25
Anexo 1 - Entrenar un modelo de IA	28
Anexo 2 - Evaluar un modelo de IA	31
Anexo 3 - ¿Qué problemas podría traer la IA?	33
Anexo 4 - Conceptos básicos de programación	34

Ficha Curricular ↓

Objetivos de aprendizaje

Comunicación y Colaboración

Reflexionar sobre los sesgos (género, nivel socioeconómico, etc.) que existen en la computación.

Reconocer distintos usos de los algoritmos de inteligencia artificial en la computación.

Análisis de datos, desarrollo y uso de Abstracciones

Identificar los datos importantes y seleccionar la información relevante de un problema dado

Utilizar una forma de representación y sistematización de datos a partir del problema a modelar

Creación de algoritmos, programas y dispositivos

Explicar durante el proceso de creación de un programa su ejecución parcial o total

Construir programas que permitan ser analizados, comprendidos y corregidos con facilidad.

Resolución de problemas computacionales

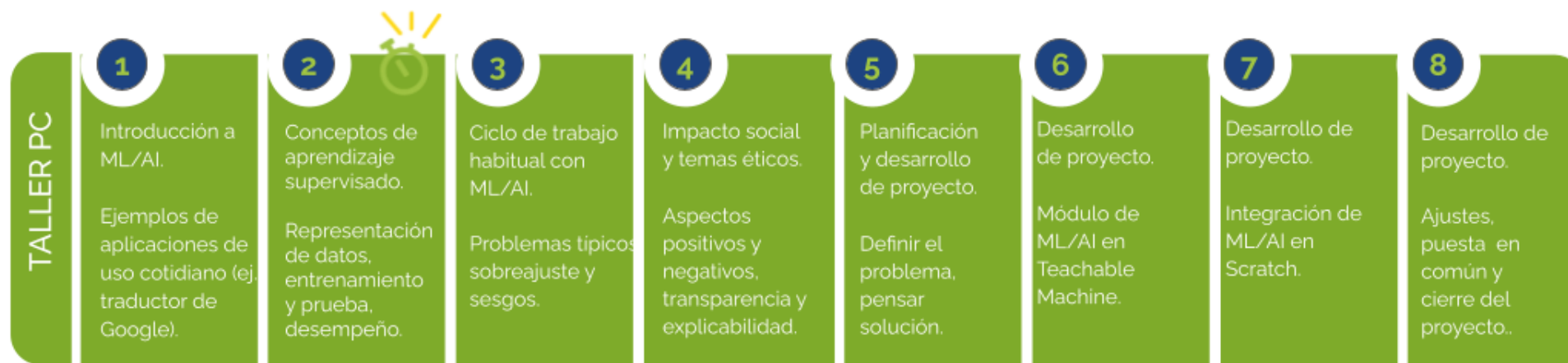
Incorporar la estrategia de división de un problema en subproblemas.

Seleccionar las herramientas de programación adecuadas para el subproblema a resolver.

Recuperar estrategias de solución y herramientas de programación construidas en experiencias anteriores para adaptarlas a nuevos problemas.

<p>Competencias RED Global</p>	<p>Posibles vinculaciones con el Programa de Educación Media <u>A definir por el docente de la asignatura específica</u></p>
<p>Ciudadanía. Dimensión Apalancamiento digital. Creatividad. Dimensión Consideración y búsqueda de nuevas ideas y soluciones.</p>	<p>Área de Conocimiento</p> <hr/> <p>Idioma español</p> <p>Espacio Curricular Abierto</p> <p>Educación Social y Cívica</p> <p>Educación Sonora</p> <p>Educación Visual, Plástica y Dibujo</p> <p>Matemática</p>
<p>Perspectiva de género</p>	
<p>Durante toda la propuesta de PC se busca propiciar una experiencia educativa inclusiva y promotora de equidad de género. Para ello docentes de aula y remotos deberán estar atentos a desnaturalizar en forma constante el sesgo de la computación y la programación como tarea exclusiva de varones.</p> <p>No existe una competencia informática inherente a un género en particular sino una desigualdad en el acceso y las posibilidades de varones y mujeres que queremos superar. Para paliar esta desigualdad, buscamos incentivar especialmente el trabajo de las niñas y brindarles todas las herramientas necesarias (atención, apoyo, retroalimentación positiva, entre otras).</p>	

Síntesis del recorrido



🕒 *En esta etapa, las actividades presenciales están previstas para más de un módulo de 45 minutos.*

Bitácora y trabajo en Scratch

El trabajo de los estudiantes a lo largo de todo el proyecto se organizará en una **bitácora por cada grupo**, donde se registran algunas actividades, y las decisiones de cada grupo.

Se ofrece un [modelo de bitácora](#) que incluye las actividades de registro que se sugieren a lo largo de todo el proyecto. Esta [bitácora](#) está completa simplemente a modo de referencia, para ejemplificar lo que se espera en cada apartado.

- Definir la forma de trabajo en la bitácora:
 - En el caso que se pueda trabajar con conectividad a internet de forma estable, una buena opción es que la bitácora sea una presentación en Google Drive (el docente previamente debería generar una cuenta de gmail para el grupo), que admite la escritura colaborativa y permite compartirse.
 - En los casos en los que se dificulta trabajar en línea se puede generar una versión offline de la presentación para que los estudiantes puedan completarla utilizando el programa Impress, instalado en sus computadoras. Al final de cada etapa deberán subir el archivo a Crea.

Versión de Scratch e idioma

Para integrar el bloque de IA que se va a generar con [Teachable Machine](#), se deberá usar la versión online de Scratch denominada [PoseBlocks](#), del sitio [Dancing with AI del MIT](#). Durante la guía se alude al nombre de los bloques en idioma “Español Latinoamericano”.

Curso en plataforma Crea ↓

Se destinará una carpeta en Crea para este proyecto dentro del Curso de PC, organizando cada carpeta internamente en las etapas previstas.

Este espacio virtual de encuentro será una guía y referencia constante, en la que se compartirán registros, consignas, foros de intercambio y actividades interactivas.

Los estudiantes publicarán allí tanto los enlaces o avances en sus bitácoras como los archivos de Scratch.



ETAPA 1 ↓

Introducción a la Inteligencia Artificial

Se presenta la temática de Inteligencia Artificial y Aprendizaje Automático. Se ilustran diversos ejemplos de aplicaciones de uso cotidiano (ej. [traductor de Google](#)). Se trabaja con los estudiantes con algunas aplicaciones ilustrativas (ej. [QuickDraw](#), [dirección de orquesta](#)). Los estudiantes exploran algunas aplicaciones y, junto al docente, analizan las distintas tareas que debe resolver la computadora para su funcionamiento.

Esta etapa se asocia a la primera de las 5 ideas principales de la guía de AI4K12/CSTA: *percepción*. Las computadoras cuentan con diversos sensores para recibir información de su entorno (ej. micrófono, cámara, sensor de luz, temperatura, acelerómetro).

Objetivos

Se espera que los estudiantes sean capaces de:

- Conocer e identificar distintos ejemplos de aplicaciones que usen aprendizaje automático.
- Conceptualizar a las aplicaciones como programas ejecutándose en una computadora.
- Identificar los datos que usa para resolver la tarea y cuáles son las formas de entrada por las que los recibe (ej. cámara).
- Descomponer una aplicación en las distintas tareas que la computadora debe resolver para su funcionamiento.

Presentación y exploración de aplicaciones de IA

Sugerencias de trabajo integrado ↓

Idioma español

Lengua. Texto escrito y oral.



Educación Visual, Plástica y Dibujo

Educación Sonora

Notas para el Docente ↓

- Se sugiere explorar la selección de aplicaciones de IA que se presenta en esta etapa. Recorrer las preguntas incluidas en el modelo de bitácora permitirá enriquecer esta exploración.
- Repasar la estructura de alguna de estas aplicaciones de ejemplo, para encontrar relaciones con el contenido específico de la asignatura.

Recursos sugeridos ↓

- [Traductor de Google](#)
 - Conversión de voz a texto 
 - Conversión de texto a voz 
- [Quick Draw](#)
- [Sketch RNN](#)
- [Director de orquesta](#)
- [GAN Paint](#)
- [Chat Bot](#)
- [Dibujar con la nariz](#)

ACTIVIDADES ↓

Presentación y exploración de aplicaciones de IA

Desafío

Impulsar la exploración de aplicaciones de IA para que los estudiantes se pongan en contacto con el desafío de crear su propia aplicación de IA.

1. Inicio (10 min)

Inteligencia Artificial

La clase inicia con una primera parte de intercambio con los estudiantes para introducir el tema. Si bien aún no vamos a profundizar en los conceptos, ya podemos ir comentando sobre las diferencias que tienen los componentes de inteligencia artificial respecto a la programación tradicional. Algunas preguntas de referencia para esta parte serían:

- ¿Alguna vez interactuaron con un [chat bot](#) o usaron [faceApp](#)?
- ¿Han llevado envases a las máquinas que están en los supermercados?
- ¿Conocen quiénes son [Alexa](#) o [Siri](#)? ¿Han usado el [asistente de Google](#) o su [traductor](#)?
- ¿Se imaginan cómo funcionan estas aplicaciones?
- ¿Qué saben de la inteligencia artificial?
- ¿Conocen otros dispositivos o aplicaciones que tengan IA?

El docente debería cerrar el intercambio comentando algunos ejemplos concretos de aplicaciones que usen IA. Es importante que sean al menos dos, por ejemplo uno con imágenes/video (ej. [detección de objetos](#)) y otro de texto (ej. [traductor de Google](#)). Esto brinda mayor variedad, lo que amplía las posibilidades de abstracción. Si se desea profundizar un poco más en esta parte, se puede hacer énfasis en la diferencia entre la IA y la programación tradicional, en particular en el hecho de que las computadoras en este caso pueden cometer errores.

- ¿Cómo hace una computadora para detectar que hay una cara en una foto?
 - ¿El programa que lo hace funciona igual que una calculadora?
- Resaltar el hecho de que la computadora se puede equivocar detectando la cara u otro objeto, pero no pasa lo mismo con la calculadora, donde el resultado siempre es correcto.

2. Desarrollo (25 min)

Explorar aplicaciones


En el segundo bloque, la idea es que los estudiantes trabajen directamente con alguna aplicación para explorar su funcionamiento.

Sugerencias

Conformar grupos de 2 a 4 integrantes para trabajar.

Gestionar la copia de la bitácora digital del proyecto entre los estudiantes (en Drive o en cada computadora).

Aquí se lista una selección de posibles aplicaciones de IA para explorar:

- [Traductor de Google](#) (y conversión de voz a texto )
- [Quick Draw](#)
- [Sketch RNN](#)
- [Director de orquesta](#)
- [Chat Bot](#)
- [Dibujar con la nariz](#)
- [Detección de objetos](#)

El objetivo de la exploración es identificar los elementos y comportamientos que podemos encontrar en las aplicaciones de IA. Para eso, los estudiantes cuentan con las preguntas de la bitácora como guía. En la [bitácora completa](#) se encuentran respuestas a algunas de las aplicaciones de referencia para el docente.

Una posible dinámica sería analizar una aplicación en conjunto, toda la clase con el docente, respondiendo a las distintas preguntas planteadas en la bitácora. Luego, los estudiantes trabajando en grupos, deberían hacer lo mismo con una o más aplicaciones de la lista sugerida. Para cerrar la actividad, se pasaría a una instancia de puesta en común de los hallazgos y

respuestas encontradas en cada una de las aplicaciones exploradas.

Puesta en común

El tiempo debería ser suficiente para que cada grupo de estudiantes analice al menos una aplicación. El docente podría elegir que todos los grupos trabajen con la misma aplicación, o elegir varias para que no todos los grupos trabajen con la misma. En ese caso podrían ser distintas aplicaciones que usen distintos tipos de datos (ej. una de texto y otra de imágenes/video).

Los estudiantes identifican las características de cada aplicación en base a las respuestas de su bitácora. Algunas preguntas de referencia para la puesta en común serían:

- ¿Qué aplicación probaron? ¿Cuál era el objetivo?
- ¿Qué era lo que podía hacer el usuario?
- ¿Qué datos de entrada utiliza la computadora? ¿Cómo los recibe?
- ¿Qué parte de la aplicación es la que tiene IA?
- ¿Qué otras tareas realiza que no son IA?

Es importante que los estudiantes sean capaces de identificar **los diferentes tipos de datos que maneja la computadora (texto, imagen/video, audio) y los dispositivos/sensores que usa para recibirlos (teclado, cámara, micrófono). Si bien existen aplicaciones donde estos datos se combinan, lo habitual es que cada aplicación de IA trabaje con un único tipo de datos.**

3. Cierre (10 min) Conceptualización

Para cerrar esta primera etapa es importante tomar unos minutos al final de la clase para repasar con los estudiantes los conceptos más relevantes de la primera clase. A modo de resumen se destaca:

- La **inteligencia artificial** en general se refiere a **computadoras que resuelven tareas** de forma **similar a los humanos**.
- A diferencia de la programación tradicional, **los métodos basados en IA pueden cometer errores** (al igual que los cometemos los humanos).
- Parte importante de su desarrollo es encontrar **datos de ejemplo adecuados** para generar los programas que luego resuelven la tarea. A este proceso habitualmente se le denomina **entrenamiento** y se dice que es cuando las computadoras *aprenden*.

Próximas etapas - Nuestra aplicación de IA

Luego de esta primera etapa, donde se hace una breve introducción al tema, se sugiere cerrar la actividad comentando la idea general de todo el proyecto sobre Inteligencia Artificial y las próximas etapas previstas. Los estudiantes deben saber desde el comienzo que luego serán ellos quienes deben desarrollar una aplicación que integre algún componente de IA.

La exploración de aplicaciones realizada permite que los estudiantes vayan pensando desde ahora cómo y para qué se podría utilizar IA en la aplicación a desarrollar en Scratch (ej. para manejar algún elemento o personaje en la aplicación, o dar respuesta a preguntas múltiple opción).

Bitácora y Foro

El docente publica en Crea la Etapa 1:

- la lista de aplicaciones de IA de ejemplo,
- la bitácora para el análisis de aplicaciones,
- la síntesis de las reflexiones principales.



Un representante de cada grupo comparte el enlace a la bitácora con la Etapa 1 completa.

La Yapa: Actividades para seguir en casa

¿Te quedaste con ganas más?
¡Podrías explorar más aplicaciones de IA en casa!
Esto puede ser útil para que se te ocurran ideas para la aplicación que van a crear más adelante. Además, pueden consultar con familiares y amigos qué otras aplicaciones conocen que usen IA, y analizar, como se hizo en clase, las distintas tareas que la computadora debe resolver para cumplir el objetivo.



ETAPA 2 ↓

Aprendizaje automático supervisado

Se introducen algunos conceptos básicos para comprender cómo las computadoras representan los datos y de qué forma es posible que aprendan de la información que obtienen de los sensores. Mediante ejemplos, se describe la principal diferencia entre la programación tradicional, basada en reglas, y el aprendizaje automático, basado en datos. Se ilustra mediante una actividad cómo es posible enseñar a las computadoras a resolver un problema en base a datos de ejemplo.

Esta etapa se asocia a la segunda de las 5 ideas principales de la guía de AI4K12/CSTA: *representación y razonamiento*. Las computadoras representan la información que reciben de los sensores y la almacenan, usando distintos tipos de datos, para luego analizarla y procesarla.

Objetivos

Se espera que los estudiantes sean capaces de:

- Reconocer información de entrada en situaciones cotidianas de uso de computadoras y cómo las máquinas las representan y almacenan en forma numérica.
- Diferenciar la programación tradicional del aprendizaje de máquinas, reconociendo según el caso a cuál corresponde.
- Identificar cómo las computadoras pueden aprender a resolver un problema de clasificación en base a datos de ejemplo.

Programación tradicional vs Aprendizaje Automático

Sugerencias de trabajo integrado ↓

Matemáticas

Los números. Representaciones. Funciones, Conjuntos.

Idioma español

Educación Visual, Plástica y Dibujo

Notas para el Docente ↓

Se puede trabajar la noción de píxel, como unidad mínima en la descomposición de la imagen. Según la asignatura y el nivel de los estudiantes, se podría profundizar en el tema, comentando sobre las diferencias entre valores continuos y valores discretos

Recursos sugeridos ↓

[Computadoras en el supermercado](#)

[Jugando con razas de perro](#)

ACTIVIDADES ↓

Programación tradicional vs Aprendizaje Automático

🕒 Actividades previstas pueden llevar más de 45 minutos

💡 Desafío

Distinguir la diferencia entre la IA y la programación tradicional. Visualizar cómo se puede “enseñar” a las máquinas mediante aprendizaje supervisado.

1. Inicio

Representación de datos (10 min)

Comenzar con un breve repaso de lo que se comentó en la primera clase sobre cómo las computadoras almacenan los datos que reciben de sus distintas entradas posibles:

- Cámara: imágenes y video,
- Teclado: texto,
- Micrófono: audio.

Los estudiantes deben ser capaces de reconocer distintos tipos de datos: números, letras, audio, imágenes, video, datos categóricos. Además, se puede aprovechar esta introducción para incorporar nociones básicas sobre cómo las computadoras almacenan la información (siempre mediante números, en particular usando el sistema binario) y qué hacen para procesarla (cálculos con dichos números). Para ilustrar los distintos tipos de datos, se puede trabajar con ejemplos concretos, como puede ser la representación numérica de una imagen. En ese caso se puede manejar la noción de píxel, como unidad mínima en la descomposición de la imagen.

2. Desarrollo - Primera parte

Programación tradicional vs IA (10 min)

La etapa sigue adelante trabajando en la diferencia entre la programación tradicional y el aprendizaje automático (inteligencia artificial), en base al siguiente esquema:

- Programación tradicional: entrada + reglas → salida
- Aprendizaje automático: entradas + salidas → reglas

La idea es ilustrar ambos casos con ejemplos concretos, para que los

estudiantes puedan conceptualizar las diferencias principales. Para ello, se propone la actividad [Computadoras en el supermercado](#).

📌 Sugerencias

Conformar grupos de 2 a 4 integrantes para trabajar.

Gestionar la copia de la bitácora digital del proyecto entre los estudiantes (en Drive o en cada computadora).

Administrar tiempos: Al planificar la clase, considerar el tiempo que dedican los estudiantes a la realización de las actividades previstas.

Repasar todo lo necesario: Es preferible extenderse más en esta etapa, repasar conceptos de la etapa previa y dedicar mayor tiempo en esta etapa si es necesario.

Previo al desarrollo de la actividad, se puede comentar algún ejemplo de programación tradicional, como puede ser la implementación de una calculadora. En este caso, el objetivo es que para cada conjunto de entradas (números y operadores) la computadora calcule el resultado. Para ello la computadora sabe exactamente las reglas (operadores suma, resta, multiplicación y división) para obtener la salida deseada en cada caso. Además, es importante notar que en esta aplicación la computadora no comete errores. Como ejemplos de aplicaciones de IA, se puede repasar alguno de los que se comentaron en la primera etapa, como por ejemplo la detección de objetos en una imagen. Un aspecto a tener presente es que al hacer uso de IA las computadoras sí pueden cometer errores, a diferencia del caso previo.

Los estudiantes trabajan la actividad en la bitácora (ver [bitácora completa](#) con respuestas) y al finalizar se puede hacer una puesta en común para compartir

las respuestas con el resto del grupo. En la actividad desarrollada en grupos, donde hay ejemplos de distintas aplicaciones de computadoras en un supermercado, veremos para cada caso si la computadora hace uso de aprendizaje automático o no. Para los casos donde la computadora sí usa IA, también podemos discutir en base a estas preguntas:

- ¿Es más sencillo para la computadora resolver la tarea que para un humano?
- ¿En qué características se podrían basar las reglas que aprende la computadora?

2. Desarrollo - Segunda parte

Aprendizaje supervisado (20 min)

Ahora que ya vimos las diferencias del aprendizaje automático con la programación tradicional, la idea es profundizar en cómo las computadoras pueden “aprender”, mediante una analogía de cómo aprendemos los humanos. Para ello, se propone ilustrar el tema usando razas de perros, como se presenta en el siguiente material: [Jugando con razas de perro](#).

La idea es que los estudiantes puedan hacerse una idea de cómo una computadora es capaz de generar reglas a partir de múltiples entradas de ejemplo cuyas salidas son conocidas (ej. fotos de perros de distintas razas). La actividad Jugando con razas de perro plantea dos desafíos para que los estudiantes incorporen los conceptos básicos que hacen posible que la computadora “aprenda” las reglas necesarias para aplicar a datos nuevos desconocidos.

En primer lugar, el objetivo es trabajar en las distintas características que se podrían considerar para distinguir las razas, y lograr identificar cuáles son útiles para ello y cuáles no. Luego, se propone trabajar en los denominados árboles de decisión, como un ejemplo de las reglas que una computadora podría construir para clasificar perros según la raza.

La [bitácora completa](#) incluye posibles soluciones a la actividad planteada. Los estudiantes deberán completar la actividad en sus respectivas bitácoras.

3. Cierre (5 minutos)

Para cerrar esta etapa es importante remarcar la diferencia conceptual entre los tipos de datos (imágenes/video, texto, audio) y la definición de clases o

categorías del modelo de aprendizaje automático. Los datos, y sus respectivos tipos, están dados por la información que recibe la computadora a partir de alguno de sus “sensores” o periféricos (por ejemplo, en el ejemplo de los perros, los datos son imágenes, y se obtienen a partir de una cámara). Por otro lado, las categorías son algo que define el humano al programar el método de inteligencia artificial. En el ejemplo de los perros las categorías se corresponden con las razas, pero se podría pensar en otras opciones (ej. usar su tamaño para agruparlos por edad).

El problema de clasificación o de reconocimiento de patrones, es posible trabajarlo desde una perspectiva matemática, mediante conceptos tales como conjuntos y funciones. Esto puede ser de utilidad para profundizar en el concepto de clases o categorías, las cuales indican las posibles “etiquetas”. Estas etiquetas deben ser conocidas para los datos de entrenamiento, que son necesarios para definir un modelo de aprendizaje automático supervisado. Como veremos en la siguiente etapa, este será un paso relevante para la construcción del modelo de aprendizaje automático, ya que es algo que tendremos que indicarle a la computadora para que sepa qué tiene que “aprender”. Esto ilustra además que el aprendizaje automático también requiere de intervención humana para su implementación.

Bitácora

Los estudiantes registran en la bitácora las distintas actividades de esta etapa, guiados por las preguntas.

[Ver bitácora de ejemplo](#).



La Yapa: Propuestas para seguir en casa

En base al trabajo realizado con las imágenes de perros, pueden agregar más razas y pensar qué otras características podrían usar para diferenciarlas y generar posibles árboles de decisión para la clasificación.



ETAPA 3 ↓

Ciclo de trabajo del aprendizaje automático

Se describe mediante un ejemplo con [Teachable Machine](#) cómo es el desarrollo de un módulo basado en aprendizaje supervisado. Se presenta los distintos conceptos asociados, como datos etiquetados, conjunto de entrenamiento y conjunto de prueba, medidas de desempeño, recorriendo cada una de las etapas del ciclo habitual de trabajo. Esto involucra la recolección de datos, la generación del modelo y la evaluación del desempeño. Se trabaja además sobre los distintos problemas típicos que aparecen, como la falta de generalización (sobreajuste y mal desempeño frente a ejemplos desconocidos) o los sesgos.

Esta etapa se asocia a la tercera de las 5 ideas principales de la guía de AI4K12/CSTA: aprendizaje. Las computadoras procesan la información, almacenada con distintas estructuras de datos, y son capaces de “aprender” identificando patrones estadísticos que se repiten o diferencian.

Objetivos

Se espera que los estudiantes sean capaces de:

- Describir los conceptos básicos del aprendizaje supervisado: datos etiquetados, categorías o clases, entrenamiento (generación del modelo) y evaluación de desempeño.
- Recolectar datos y definir etiquetas para generar un modelo de IA.
- Evaluar el desempeño que tiene un modelo generado, en base a pruebas de ensayo y error.
- Comprender cómo los algoritmos de aprendizaje automático dependen de los datos de ejemplo utilizados para generarlos.

Entrenar y evaluar un modelo de IA

Sugerencias de trabajo integrado ↓

Espacio Curricular Abierto

Notas para el Docente ↓

Se sugiere trabajar con el reconocimiento de patrones desde una perspectiva matemática ya que funcionará como precedente para introducir el concepto de repetición en programación, pero el DA decidirá si además aprovecha la oportunidad para abordar el reconocimiento de patrones desde otras áreas como Lengua (ortografía), Arte, Desafíos Bebras y los relaciona con esta actividad.

Recursos sugeridos ↓

[Teachable Machine](#)

[Anexo 1](#)

[Anexo 2](#)

ACTIVIDADES ↓

Entrenar y evaluar un modelo de IA

Desafío

Trabajar en el ciclo de creación de IA, desde la recolección de datos hasta la evaluación de desempeño del modelo definido y entrenado.

1. Inicio

Presentación de Teachable Machine (10 min)

En esta etapa vamos a trabajar en la generación de modelos de aprendizaje automático supervisado usando [Teachable Machine](#). Esta herramienta nos va a permitir con ejemplos concretos, mostrar cómo es el ciclo habitual para trabajar con aprendizaje supervisado:

- Definición de clases/categorías (etiquetas)
- Recolección de datos de entrenamiento
- Preparación del modelo de IA
- Evaluación de desempeño

Además, los estudiantes mediante la experimentación con los modelos entrenados, podrán incorporar nociones básicas sobre conceptos como sobreajuste y sesgos. Estos son problemas típicos que los datos de entrenamiento generan a la hora de usar modelos de aprendizaje automático.

2. Desarrollo

Definir y entrenar un modelo de IA (30 min)

La actividad propuesta tiene dos partes. En la primera ([ver Anexo 1](#)) se presenta la generación del modelo. Para ello, es necesario en primer lugar definir las clases del problema de clasificación a resolver (en el ejemplo: mano izquierda, mano derecha y ninguna mano). Luego, se deberá recolectar los datos correspondientes a cada clase o categoría para poder entrenar el modelo.

Actividad 1 - Modelo de IA en Teachable Machine

El objetivo de esta actividad es generar un modelo de IA en Teachable Machine, definiendo tres clases o categorías:

- Mano Izquierda: corresponde a una imagen de una persona en primer plano donde la mano izquierda está levantada con la palma abierta.
- Mano Derecha: corresponde a una imagen de una persona en primer plano donde la mano izquierda está levantada con la palma abierta.
- Ninguna Mano: corresponde a una imagen de una persona en primer plano donde ninguna de sus manos está presente en la imagen.

Vale aclarar que una imagen de una persona en primer plano, corresponde a una foto de frente donde es posible ver su cuerpo desde el pecho hacia arriba, incluyendo su rostro en forma completa.

Una vez definidas las tres clases, el siguiente paso es recolectar datos para cada una de las clases. Para ello se puede utilizar la funcionalidad que tiene Teachable Machine para hacer esto en forma automática con una frecuencia predefinida (ej. 20 fotos por segundo).

Por último, para generar el modelo de IA solamente resta indicar al sistema que los datos están prontos y presionar el botón correspondiente para que se realice el entrenamiento.

Actividad 2 - Evaluación de desempeño

Una vez que tenemos el modelo funcionando, podemos verificar su funcionamiento también dentro de Teachable Machine. Para ello, seguiremos el cuestionario ([ver Anexo 2](#)) y dejaremos registro de las respuestas en la bitácora.

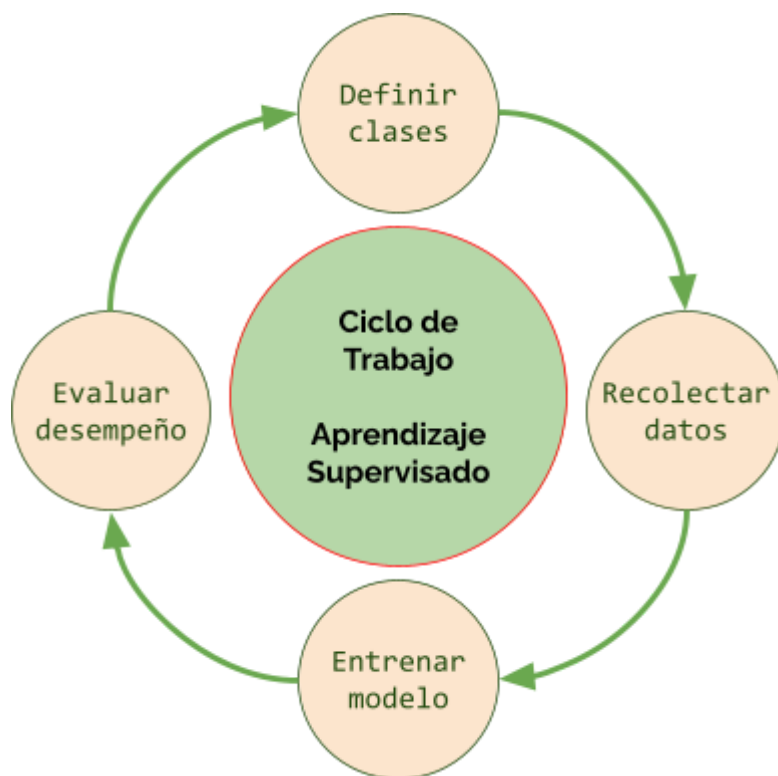
Atención

Administrar tiempos: Al planificar la clase, considerar el tiempo que dedican los estudiantes a la realización de las actividades previstas.

3. Cierre

Conceptualización (5 minutos)

Para cerrar la actividad, es importante volver a hacer énfasis en las distintas etapas presentadas del ciclo habitual de trabajo con aprendizaje supervisado, representadas en esta figura:



Bitácora

Los estudiantes registran en la bitácora las actividades realizadas en esta etapa, guiados por las preguntas.

[Ver bitácora de ejemplo.](#)



La Yapa: Propuestas para seguir en casa

¿Qué les gustaría que la computadora sea capaz de reconocer? Pueden seguir explorando el uso de Teachable Machine para ir pensando ideas para su proyecto.



ETAPA 4 ↓

Impacto social y temas éticos asociados a la IA

En esta etapa se pretende generar conciencia y discutir sobre el impacto que tiene o podría tener, el uso masivo de IA en distintas aplicaciones, yendo desde la medicina hasta el desarrollo de armas de guerra. Además, el propio hecho de delegar en computadoras la toma de ciertas decisiones, implica también consideraciones éticas que deben tenerse en cuenta a la hora de desarrollar este tipo de sistemas.

Esta etapa se corresponde con la quinta de las 5 ideas principales de la guía de AI4K12/CSTA: *impacto social*. La implementación en aumento de sistemas que usan ML/AI puede tener impactos positivos pero también negativos en la sociedad. Por eso es importante discutir el impacto y encontrar criterios para un manejo ético adecuado.

Objetivos

Se espera que los estudiantes sean capaces de:

- Tomar conciencia de las implicancias que tiene y/o tendría delegar ciertas tareas a las máquinas, haciendo uso de aprendizaje automático.
- Identificar cuestiones éticas que puede ser necesario resolver a priori, al momento de generar los modelos y programar.
- Problematizar la sustitución masiva de personas por máquinas para distintas tareas, y su impacto a nivel laboral y social.

¿Qué problemas podría traer la IA?

Sugerencias de trabajo integrado ↓

Espacio Curricular Abierto

Educación Social y Cívica

Notas para el Docente ↓

El docente puede buscar ejemplos asociados a su asignatura o al tema de interés particular que le gustaría abordar, ligado al impacto que podría tener la irrupción de algoritmos de inteligencia artificial en esa aplicación específica.

Recursos sugeridos ↓

[Máquina moral](#)

ACTIVIDADES ↓

¿Qué problemas podría traer la IA?

Desafío

Problematizar y discutir acerca del uso indiscriminado de IA en diversas aplicaciones.

1. Inicio (5 min)

Se presenta la temática, ilustrando con ejemplos los problemas o dilemas éticos que el uso de IA podría generar. Algunas de las preguntas planteadas ([ver Anexo 3](#)) podrían servir como disparador:

¿Quién se hace responsable en un accidente de tránsito causado por un vehículo autónomo?

¿Qué actividades realizan las computadoras mejor que los humanos?

¿Podrá la inteligencia artificial sustituir a los humanos en todo tipo de trabajos?

¿Las computadoras pueden usar datos personales para aprender?

2. Desarrollo

Debate sobre impacto y temas éticos (20 min)

Para problematizar el tema con los estudiantes, se propone como actividad realizar el Debate ([ver Anexo 3](#)).

Sugerencias

Conformar grupos de 2 a 4 integrantes para trabajar, y que cada grupo se concentre en una sola pregunta. Luego al finalizar, cada grupo expone la pregunta que le tocó discutir y a qué respuesta llegaron en consenso. También pueden participar allí los estudiantes de otros grupos que no hayan trabajado con esa pregunta.

Administrar tiempos: Al planificar la clase, considerar el tiempo que dedican los estudiantes a la realización de las actividades previstas.

Finalizado el debate, se propone como actividad trabajar con la [máquina moral](#). Esta actividad nos enfrenta a dilemas éticos de difícil solución, donde no es sencillo decidir cuál es la opción correcta, y ni siquiera son claros los factores que habría que tener en cuenta para optar por una opción u otra. Todo esto sirve para ilustrar las implicancias que tiene el uso de IA, y también cuestionarse cómo se dirimen estos dilemas actualmente, cuando aún las tareas son realizadas por seres humanos.

3. Cierre (5 minutos)

Para dar cierre a la etapa, se sugiere que el docente sintetice todo lo discutido en las distintas actividades. Es importante fomentar el pensamiento crítico de los estudiantes, desmitificando la temática y planteando con claridad los temas a resolver para el uso cotidiano de IA en distintos contextos.

Bitácora

Los estudiantes registran en la bitácora las actividades realizadas en esta etapa.

[Ver bitácora de ejemplo.](#)



La Yapa: Propuestas para seguir en casa

Se propone seguir conversando el tema en casa, junto a las familias, discutiendo posibles mundos futuros imaginarios con mayor presencia de inteligencia artificial, y las implicancias que esto podría tener.



Jugando con la máquina moral (15 min)

ETAPA 5 ↓ Planificación del proyecto

En esta etapa, los estudiantes deberán diseñar el proyecto que van a llevar a cabo en las próximas etapas. Una vez definido, tendrán que descomponer el trabajo en los distintos bloques y pensar posibles soluciones para cada uno.

En la guía se trabaja en base a un videojuego de ejemplo, que servirá de apoyo y se podrá tomar como base para el desarrollo a realizar por parte de los estudiantes.

Objetivos

Se espera que los estudiantes sean capaces de:

- Reconocer a la computadora como una máquina que procesa información de entrada y produce resultados en consecuencia.
- Diseñar la aplicación en un papel, que luego se llevará a la implementación mediante la programación.
- Incorporar la resolución de problemas mediante el uso de IA con el módulo elegido para el proyecto.
- Planificar las etapas necesarias para el desarrollo del proyecto.

Diseñar la aplicación que integra IA

Sugerencias de trabajo integrado ↓

En la materialización del proyecto es posible que entren en juego todas las asignaturas, de acuerdo a los intereses de estudiantes y docentes.

Notas para el Docente ↓

En esta etapa comienza la fase de diseño e implementación del proyecto. Es decir que es importante definir el alcance y temática del proyecto para que el resultado final sea factible en el tiempo disponible.

Recursos sugeridos ↓

[Anexo 4](#)

[Tarjetas de programación en Scratch](#)

[Tarjetas de módulos de IA](#)

ACTIVIDADES ↓

Diseñar la aplicación que integra IA



Desafío

Diseñar el proyecto que cuente con algún módulo de IA.

1. Inicio

Presentación del proyecto (10 min)

En esta etapa vamos a trabajar en base a un ejemplo que ayude a los estudiantes a diseñar su propia aplicación en Scratch, integrando un módulo con IA. Para ello, comenzaremos ilustrando un videojuego, que hace uso del módulo de IA presentado en la etapa 3. En este caso, mediante el bloque que detecta si mostramos la mano izquierda, la mano derecha o ninguna, generamos una aplicación en Scratch que nos permita manejar un personaje usando las manos para controlarlo. Luego la idea es construir un juego en base a dicha integración. La consigna es identificar si un número es par o impar, para lo cual se sortea un número aleatorio entre 0 y 100, y el jugador deberá elegir moviendo el personaje hacia la derecha o hacia a la izquierda, para lo cual levantará la mano correspondiente.

Sugerencia

Adaptar la propuesta a la experiencia previa de los estudiantes con Scratch: El juego de ejemplo se puede ilustrar con [distintos niveles de complejidad en la programación](#), debiendo seleccionar el que más se ajuste a los conocimientos y experiencia previa de los estudiantes con Scratch. En la medida que ellos luego avancen en el desarrollo de su propio juego, se podrá mostrar nuevas funcionalidades de mayor complejidad.

La idea en el comienzo es dar unos minutos para que los estudiantes experimenten con el videojuego de ejemplo. De esa forma podrán hacerse una idea de lo que ellos van a poder realizar en base a sus propias ideas. Además esto sirve para luego facilitar la comprensión de los distintos bloques de la programación.

2. Desarrollo

Bloques relevantes para el juego (15 min)

El siguiente paso es trabajar en los bloques fundamentales del juego. El tiempo dedicado a esta parte dependerá mucho de la experiencia previa que los estudiantes tengan trabajando con Scratch. Algunos aspectos fundamentales a destacar del juego son:

Generador de números aleatorios y uso de variables: Hay múltiples aplicaciones donde generar números al azar es sumamente útil, por lo que es importante aprender la forma de hacerlo al programar. Además, es necesario guardar el número generado, para luego compararlo con la respuesta del usuario, lo que introduce el concepto de variables, otro concepto muy relevante a la hora de programar.

Integración de módulo de IA: Resulta relevante comprender el funcionamiento del bloque basado en Teachable Machine, el cual indica como salida la clase detectada (en este caso qué mano levanta el usuario o si no levanta ninguna).

Uso de condicionales: Como en la mayoría de los programas, la salida del juego dependerá de la respuesta del usuario, por lo que es fundamental conocer el uso de condicionales, para que la computadora genere la salida correcta según la respuesta del usuario.

Para trabajar en la comprensión del funcionamiento de estos bloques, podemos plantear actividades en base a [tarjetas de Scratch](#). En base a estas tarjetas podemos trabajar algunos conceptos claves de programación como variables, condicionales y bucles. [Ver el Anexo 4](#) para más información al respecto.

Preguntas guía para el diseño (15 min)

En la segunda parte de esta etapa, se trabaja ya en el diseño del juego. Para ello, se plantean algunas preguntas, que en primera instancia se deberán responder para el juego de ejemplo (ver [bitácora con respuestas](#)). Luego, estas mismas preguntas sirven para que los estudiantes en base a sus propias ideas diseñen el juego que luego van a programar. Las preguntas de referencia en este caso son:

¿Cuál es la consigna del juego?

¿De qué forma el usuario indica la respuesta?

¿Qué ocurre cuando el usuario responde?

¿Cuándo termina el juego?

Para simplificar el diseño en el caso de estudiantes con escasa o nula experiencia previa en programación, las variantes sugeridas sobre el juego de ejemplo son:

- Modificar consigna del juego, manteniendo el sorteo aleatorio de un número entre 0 y 100. Por ejemplo, en vez de identificar si el número es par o impar, la consigna puede ser responder si es divisible entre 3 o no, si es divisible entre 5 o no, si es primo o no, etc. En cada caso es importante pensar cómo será posible chequear la condición en base al número sorteado, de manera de verificar si la respuesta del usuario es correcta o no.
- Cambiar los gestos que usa el usuario para dar la respuesta. En vez de levantar una u otra mano, se podría considerar mover la cabeza para un lado o para el otro, o hacer distintos gestos con las manos. Es importante que al pensar los gestos, también se tenga en cuenta la dificultad que podría tener generar el modelo de IA que cumpla con el objetivo. En base a la experiencia previa trabajando con Teachable Machine, los estudiantes ya deberían ser conscientes de las dificultades que enfrenta la computadora para identificar los gestos en base a las imágenes capturadas por la cámara.

3. Cierre

Planificación de los próximos pasos (5 min)

Para finalizar, adelantamos a los estudiantes que en la próxima etapa nos concentraremos únicamente en el módulo de IA. Por lo tanto, deberán pensar

en base a los gestos seleccionados para que el usuario de sus respuestas, cómo debería ser el modelo necesario para implementarlo.

Bitácora

Los estudiantes suben el diseño del proyecto a sus respectivas bitácoras. [Ver bitácora de ejemplo](#).



La Yapa: Propuestas para seguir en casa

¿Y si hacemos que se escuchen distintos sonidos cuando acertamos o erramos? Explorar los bloques de la categoría *Sonido* para elegir sonidos que nos gusten para usar cuando el jugador responde bien y cuando se equivoca. También es posible explorar más opciones sobre los personajes y textos que podríamos mostrar (ver [tarjetas de Scratch](#)). ¿Se te ocurren otras maneras de informar que acertamos al objetivo?



ETAPA 6 ↓

Implementación del bloque de IA

En esta etapa los estudiantes deberán implementar mediante Teachable Machine el módulo de IA que van a integrar a su proyecto. Se debe trabajar en las distintas etapas que vieron previamente, como ser la definición de las distintas clases a considerar y la correspondiente recolección de datos, el entrenamiento del modelo, y la evaluación de desempeño.

Objetivos

Se espera que los estudiantes sean capaces de:

- Generar de forma autónoma un modelo a IA con la herramienta [Teachable Machine](#).
- Comprender los conceptos de datos de entrenamiento y datos de prueba.
- Incorporar una metodología adecuada de evaluación del desempeño de un algoritmo de IA.

Desarrollo de módulo de IA para el juego

Notas para el Docente ↓

Tomar como base y hacer mención a lo trabajado durante la etapa 3, de modo que los estudiantes puedan poner en juego lo aprendido hasta el momento

Recursos sugeridos ↓

[Anexo 4](#)

[Tarjetas de programación en Scratch](#)

[Tarjetas de módulos de IA](#)

ACTIVIDADES ↓

Desarrollo de módulo de IA para el juego

Desafío

Definir los gestos que el jugador va a usar para indicar su respuesta y crear modelo de IA que los identifique.

1. Inicio

Repaso de aprendizaje supervisado (10 min)

En esta etapa se retoma lo trabajado en la Etapa 3, con el objetivo de que cada grupo defina e implemente de forma autónoma, su propio modelo de IA. Este será luego integrado en el juego a desarrollar con Scratch, usado para identificar las respuestas que da el jugador al hacer cierto gesto (ej. mover sus manos o la cabeza).

Es importante repasar los conceptos de datos de entrenamiento, clases o categorías del modelo, y las nociones básicas de evaluación de desempeño.

La idea es que cada grupo logre cumplir con las siguientes tareas a lo largo de la etapa: definición de clases del modelo, recolección de datos de entrenamiento, generación del modelo de IA, evaluación de desempeño, ajustes y re-entrenamiento con nuevos datos (de ser necesario).

2. Desarrollo

Generación del modelo de IA (25 min)

Repasando el [modelo generado en la etapa 3](#), los estudiantes deberán ahora repetir los pasos para el desarrollo de su propio modelo. Para ello, se sugiere contestar algunas preguntas, que deberán registrar en su [bitácora](#):

¿Qué respuestas posibles tiene cada instancia del juego? ¿Cuántas son?

¿Cuáles son los gestos que debe usar el usuario para indicar cada respuesta?

¿Cuántas clases deberá tener el modelo de IA a generar?

En el juego de ejemplo las posibles respuestas son solamente dos, en ese caso decir si el número sorteado es par o impar. De esa forma, cada respuesta está asociada a levantar una de las manos, por ejemplo la derecha para par y la izquierda para impar. Para hacerlo simple, se sugiere mantener un juego con solamente dos respuestas posibles. Eso facilita la tarea a la hora de desarrollar el modelo de IA. Además, se sugiere siempre agregar una categoría adicional

para el caso donde no se está indicando ninguna respuesta (es decir, el usuario no está haciendo ningún gesto que corresponda a las opciones válidas). Con esto tendremos todo lo que necesitamos para pasar a trabajar con [Teachable Machine](#). Allí tendremos que definir las clases, recolectar las imágenes para cada una y luego generar el modelo de IA. Finalmente, debemos probar si el modelo generado funciona como esperamos.

3. Ajustes finales y cierre (10 minutos)

Si el desempeño no es el esperado, hay que ver qué problemas pueden estar ocurriendo e intentar solucionarlos. Recordar lo que vimos en etapas previas, como incluir imágenes variadas para cada clase, contemplando todos los casos que pueden ocurrir luego en el funcionamiento real. También pensar si las clases definidas son sencillas de identificar o si son muy parecidas entre sí, lo cual puede llevar a que sea más complejo diferenciarlas por el algoritmo de IA.

4. Opcional

De haber tiempo adicional, se puede avanzar en el trabajo de integración del módulo de IA en Scratch, usando las [tarjetas de IA de PoseBlocks](#).

Bitácora

Los estudiantes registran en la [bitácora](#) las decisiones que tomaron en esta etapa, guiados por preguntas. Suben a Crea el archivo de Teachable Machine con el avance de esta etapa.



La Yapa: Propuestas para seguir en casa

Si bien aún no está pronto el videojuego completo, el módulo de IA sí está listo. Por lo tanto, se puede seguir probando en casa y compartirlo con familiares y amigos.



ETAPA 7 ↓

Implementación del juego en Scratch

En esta etapa el trabajo principal se concentra en el desarrollo necesario en Scratch para completar el proyecto. Se debe integrar el módulo de IA desarrollado previamente, el cual será un bloque relevante para completar el proyecto.

Objetivos

Se espera que los estudiantes sean capaces de:

- Adquirir un manejo de programación en bloques que les permita implementar un juego sencillo.
- Integrar el módulo de IA desarrollado al juego implementado con [PoseBlocks](#) (Scratch de Dancing with AI).
- Incorporar conceptos básicos de programación, como son variables, estructuras condicionales y repetitivas.

Integración de módulo de IA al juego

Notas para el Docente ↓

Es posible que esta etapa presente dificultades vinculadas a la programación que hasta el momento no se habían considerado en profundidad. Es importante considerar las experiencias previas en el uso de Scratch y estar atento a los nuevos desafíos para ayudar y hacer preguntas que guíen la resolución de los mismos.

Recursos sugeridos ↓

[Anexo 4](#)

[Tarjetas de programación en Scratch](#)

[Tarjetas de módulos de IA](#)

ACTIVIDADES ↓

Integración de módulo de IA al juego

🕒 Actividades previstas pueden llevar más de 45 minutos

💡 Desafío

Implementar el videojuego diseñado mediante programación en bloques, incorporar el módulo de IA previamente desarrollado.

1. Inicio

Consigna del juego (5 min)

Los estudiantes deben retomar la consigna del juego que crearon, para poder realizar la implementación usando programación en bloques. Para eso, se debe definir el conjunto de respuestas posibles y la condición que determina si el jugador acierta o falla en cada caso. Las distintas opciones posibles se registran en la [bitácora](#).

Opciones posibles para el juego de ejemplo de identificar si un número sorteado al azar entre 0 y 100, es par o impar:

- Si se sortea un número **par** y la respuesta es **par**, es un **acierto**.
- Si se sortea un número **impar** y la respuesta es **impar**, es un **acierto**.
- Si se sortea un número **par** y la respuesta es **impar**, es un **fallo**.
- Si se sortea un número **impar** y la respuesta es **par**, es un **fallo**.
- Si se sortea un número **par o impar** y **no hay respuesta** (se termina el tiempo máximo para responder), es un **fallo**.

Los estudiantes deberán hacer el mismo análisis para el juego que hayan diseñado, identificando los distintos resultados posibles.

2. Desarrollo (30 min)

El resto de la etapa estará dedicado a guiar a los estudiantes en la implementación del juego que diseñaron previamente en papel, ahora utilizando programación en bloques en Scratch (usando [PoseBlocks](#) que permite integrar el módulo de IA hecho antes). Para ello se sugiere presentar

algunas herramientas y conceptos básicos de programación, que seguramente serán de gran ayuda ([ver material en el Anexo 4](#)). Se puede usar las tarjetas de Scratch presentadas previamente ([tarjetas de bloques relevantes](#), [tarjetas de módulos de IA](#)). Además, se presentan los principales bloques con el juego de ejemplo de identificar números pares e impares.

3. Cierre (10 minutos)

Se puede aprovechar la ocasión para generar un intercambio que sirva para sacar conclusiones en relación al contenido matemático y su programación en el proyecto. Por ejemplo, sobre el bloque **módulo**, y resaltar la práctica de generalización, tanto de las reglas como del programa construido.

También es una oportunidad para trabajar la noción de generalización, tanto desde la perspectiva matemática (una afirmación válida sin importar el número sobre el que predique) como desde la computacional (un único mecanismo para resolver una pregunta -¿el número es par?- que funciona para todos los números posibles).

Bitácora

Los estudiantes registran en la [bitácora](#) las decisiones que tomaron en esta etapa, guiados por preguntas. Además, pueden completar el glosario de bloques.

Suben a Crea el archivo de Scratch con el avance de esta etapa.



ETAPA 8 ↓ Cierre del proyecto

Ajustes finales y puesta en común del trabajo

En esta última etapa, se realizan ajustes finales a los distintos juegos creados. Además, se hace la puesta en común de los proyectos, donde cada grupo puede compartir su trabajo y ver lo que hicieron los otros. Por último, se planifica la difusión interna del proyecto y con la comunidad.

Como adicional opcional, también es posible desarrollar una actividad de evaluación final del proyecto.

Objetivos

Se espera que los estudiantes sean capaces de:

- Realizar una autoevaluación de su trabajo a lo largo de la propuesta.
- Reconocer y reflexionar sobre su propio proceso de aprendizaje durante todo el proyecto.
- Compartir sus producciones con sus familias y la comunidad educativa.

Notas para el Docente ↓

El cierre es el momento para repasar todo el recorrido, haciendo énfasis tanto en el producto final como en el proceso. Es importante que los estudiantes tengan la oportunidad de hacer consciente y explícito los momentos de aprendizaje y las estrategias que utilizaron en el proyecto.

ACTIVIDADES ↓

Ajustes finales y puesta en común del trabajo

1. Ajustes finales y publicación de los proyectos (25 min)

Los grupos resuelven los pendientes y ajustan los detalles finales de sus proyectos, para luego poder compartirlos con el resto de la clase. A medida que van finalizando, suben en Crea sus proyectos. Con los proyectos compartidos, cada estudiante puede jugar o utilizar el videojuego/aplicación de otro grupo.

2. Metacognición del proyecto (20 min)

A partir de la dinámica de cierre que el docente prefiera, el propósito de la metacognición es que los estudiantes reflexionen sobre los siguientes aspectos:

Los programas de computadora como producción humana.

¿Son conscientes de que crearon un programa de computadora?

¿Cómo se crearán los otros programas que usamos cotidianamente?

El videojuego como un programa construido por los estudiantes para jugar y generalización de la tarea de programación: todos los programas que usan en sus computadoras (para escribir, dibujar, navegar por internet, etc) fueron construidos por personas, como un conjunto de instrucciones para la computadora. Esta observación es una oportunidad para que los estudiantes tomen conciencia de lo poderoso de la tarea de aprender a programar.

Estrategia de división en subproblemas y módulo de IA.

¿Recuerdan la descomposición en subtareas que hicimos para distintas aplicaciones?

¿Por qué creen que lo trabajamos de esa manera?

¿Cómo distinguimos aquella tarea que hacía uso de IA de las otras?

La descomposición de un problema grande en varios problemas más

pequeños, suele ser una estrategia fructífera para distintas aplicaciones (no solamente en el ámbito de la programación). El famoso “divide y vencerás” suele ser útil para facilitar la resolución del problema grande, que a priori puede abrumarnos por su alcance y dificultad, en base a trabajar paso a paso en etapas más simples y acotadas, que finalmente todas juntas nos darán el resultado esperado. En este contexto además se introduce la IA, como un enfoque distinto a la programación tradicional. En esta última, lo que hacemos es escribir reglas para que la computadora siga, y de esa forma calcule la salida para cada entrada. En cambio, la IA está basada en datos, y lo que se busca es que la computadora infiera las reglas a partir de un conjunto de datos para los que la salida esperada es conocida.

Aprendizaje supervisado para enseñarle a la computadora.

¿A qué nos referimos con aprendizaje supervisado?

¿Qué quiere decir que la computadora “aprende”?

¿Qué problemas típicos enfrentamos al implementar modelos de IA?

Resulta muy importante en el cierre, repasar los conceptos principales de IA que se vieron en las distintas etapas. De esta forma los estudiantes reflexionan nuevamente sobre ellos, una vez que ya pasaron por todo el proceso de desarrollo de su propio proyecto. También es una oportunidad para volver a insistir en que no hay “magia” detrás de la IA, y que todos los resultados que podamos obtener van a depender fuertemente de los datos que tengamos disponibles a priori para la construcción del modelo.

Herramientas de programación aprendidas.

Algunos bloques fueron particularmente relevantes en este proyecto (el uso de variables o las alternativas condicionales, por ejemplo). Podemos asociarlas tanto a su uso en el proyecto, como en otras situaciones cotidianas discutidas. El glosario de bloques de la [bitácora](#) puede ser de utilidad como

punto de partida.

Proceso de aprendizaje

¿Qué cosas de las que hicimos no conocían? (qué aprendimos)

¿Cómo relatarían a otra persona el camino recorrido durante este proyecto? (cómo lo aprendimos)

¿Qué lograron construir con sus nuevos conocimientos? (para qué les sirvió)

¿En qué otras situaciones creen que pueden utilizar lo aprendido? (generalización)

Aspectos emocionales

¿Recuerdan qué fue lo que los ayudó a destrabar un problema o encontrar una solución? ¿Distinguen alguna actitud propia o de los compañeros que los ayudó a avanzar con el proyecto? ¿Se escucharon todas las voces y se buscó que se reflejen en el producto final? ¿Se tuvieron en cuenta las ideas y fortalezas de cada miembro del equipo? ¿Cómo enfrentaron el error? ¿Lo tomaron como una oportunidad de aprendizaje o los frustró? ¿Por qué?

¿Recuerdan cómo se sintieron frente al desafío al iniciar este proyecto? ¿Y cómo se sienten ahora con lo logrado? ¿Cómo se sintieron a lo largo de las distintas etapas

Crea

Los estudiantes suben el archivo del videojuego creado en Scratch usando IA. También pueden dejar el enlace al módulo de Teachable Machine.

Se sugiere que escriban una breve descripción del juego, que también pueden publicar en el blog disponible en su perfil de usuario.



Evaluación final del proyecto

Los estudiantes completan la actividad de evaluación disponible en la carpeta de esta etapa.

La Yapa: Propuestas para seguir en casa

¡Excelente! ¡Ya tienes un videojuego completo funcionando y que usa Inteligencia Artificial!

Puedes seguir trabajando para agregarle más cosas, continuar con el dibujo de los disfraces y agregar nuevos personajes u objetos que se muevan para animar el fondo. También puedes agregarle una presentación con imágenes o sonido ¡que hasta puedes grabar tú! o un mensaje de cierre. ¡Ahora te toca a ti!

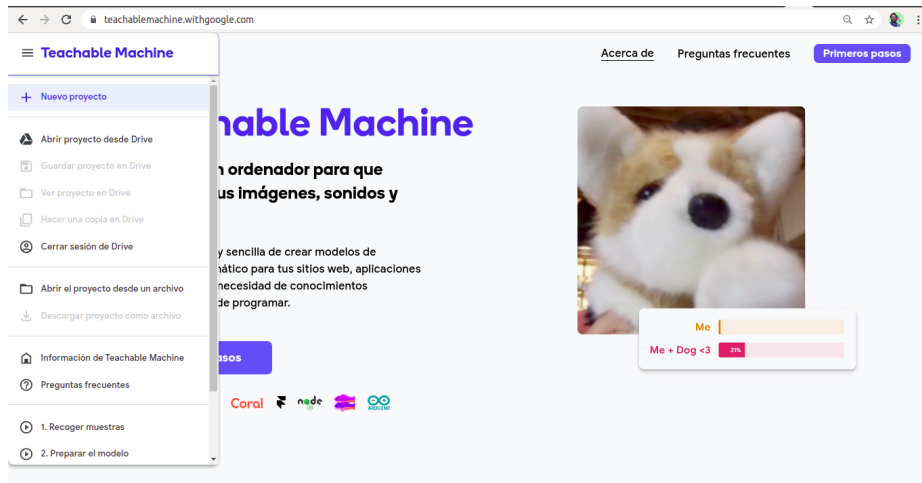


Anexo 1 - Entrenar un modelo de IA

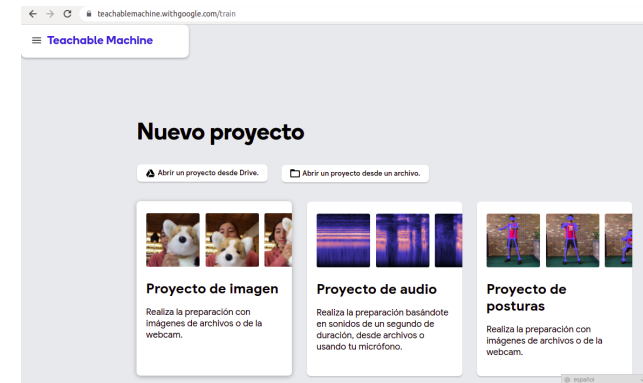
Ciclo de trabajo del aprendizaje automático

Modelo de IA en Teachable Machine

El objetivo de esta actividad es generar un modelo de IA en [Teachable Machine](https://teachablemachine.withgoogle.com). Para ello el primer paso es acceder a la web y seleccionar *Nuevo proyecto*:



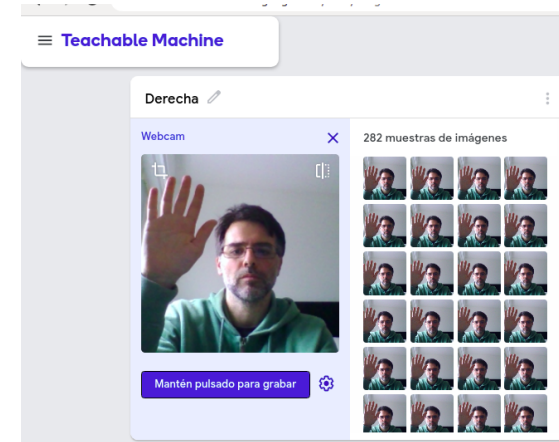
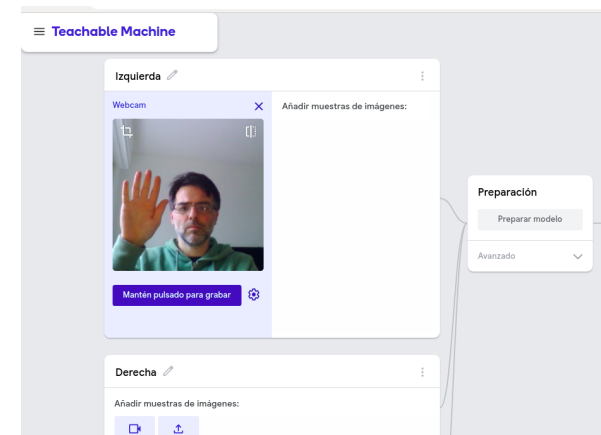
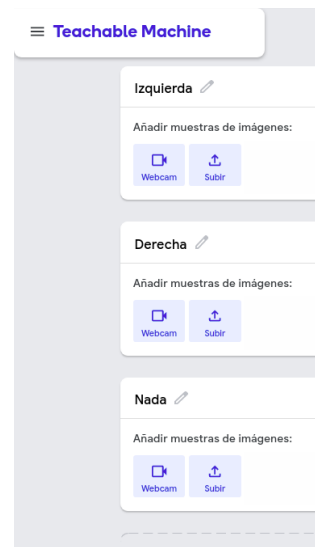
En este caso vamos a trabajar con un *Proyecto de imagen*, por lo que seleccionamos esa opción, y luego *Modelo de imagen estándar*:



El modelo a definir tendrá tres clases, que llamaremos:

- **Izquierda:** corresponde a una imagen de una persona en primer plano donde la mano izquierda está levantada con la palma abierta.
- **Derecha:** corresponde a una imagen de una persona en primer plano donde la mano izquierda está levantada con la palma abierta.
- **Nada:** corresponde a una imagen de una persona en primer plano donde ninguna de sus manos está presente en la imagen.

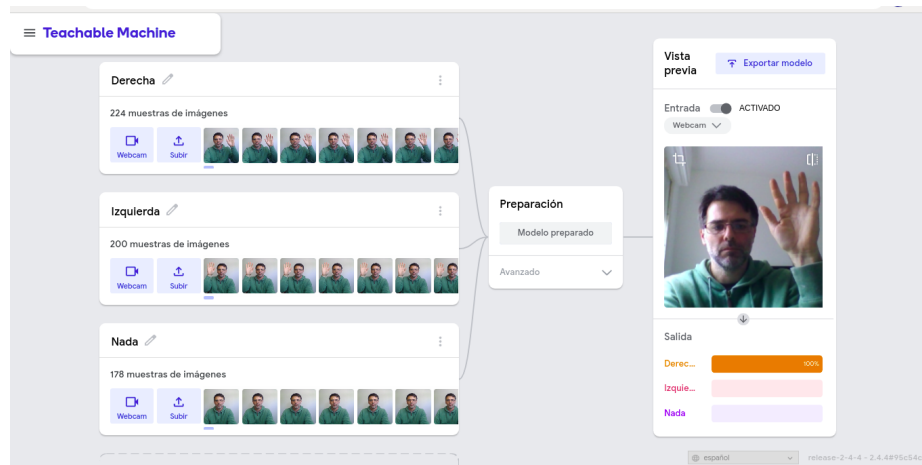
Para ello debemos editar los nombres de las clases que aparecen en el proyecto vacío (*Class 1* y *Class 2*). Además, como nuestro modelo tiene tres clases, debemos añadir una más con la opción *Añadir una clase*.



Una vez definidas las tres clases, el siguiente paso es recolectar datos para cada una de las clases. Para ello se puede utilizar tanto la webcam (con la opción *Webcam*) como cargando imágenes desde la computadora (con la opción *Subir*). Con la opción *Webcam* es posible sacar múltiples fotos de forma automática con el botón *Mantén pulsado para grabar* (lo hace a una frecuencia predefinida de 24 fotos por segundo que puede ser modificada).

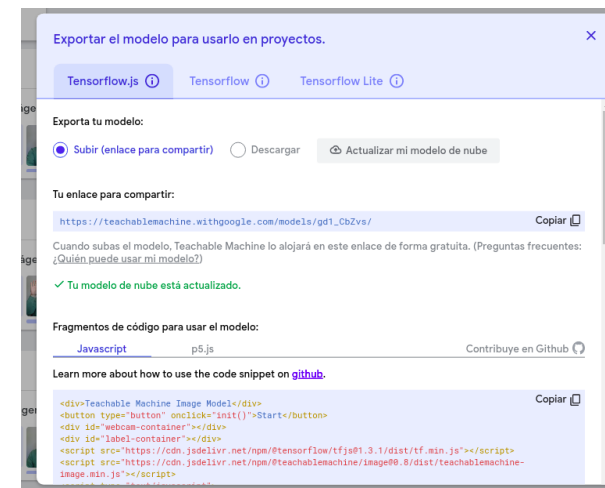
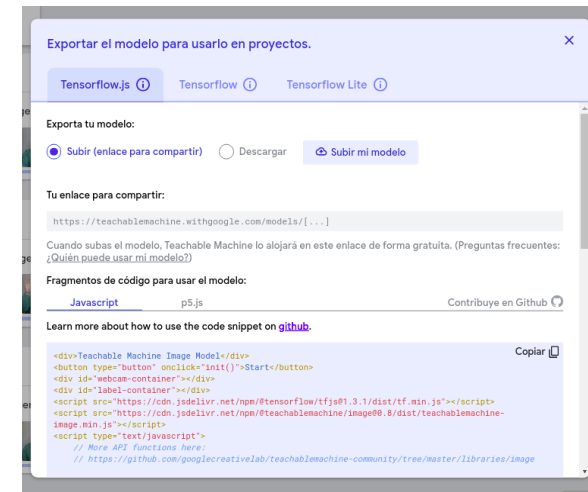
Debajo podemos ver un ejemplo cargando fotos para una de las clases. Vale aclarar que una imagen de una persona en primer plano, corresponde a una foto de frente donde es posible ver su cuerpo desde el pecho hacia arriba, incluyendo su rostro en forma completa. Por último, para generar el modelo de IA se debe presionar el botón *Preparar modelo* y esperar unos segundos a que el entrenamiento culmine.

Una vez que el modelo queda pronto, podemos previsualizar su funcionamiento a la derecha de la pantalla.



Exportar modelo para usar en Poseblocks

Además, el botón *Exportar modelo*, nos permite generar un enlace, que luego podremos compartir para que otros prueben nuestro modelo. Para ello debemos seleccionar *Subir mi modelo a la nube*. Una vez terminada la carga, se genera un enlace debajo que es el que podemos compartir. Este mismo enlace, será además el que nos permita integrar el modelo de IA generado en un programa de Scratch utilizando [PoseBlocks](#).



Anexo 2 - Evaluar un modelo de IA

Evaluación de desempeño

Una vez que tenemos el modelo funcionando, podemos verificar su funcionamiento también dentro de Teachable Machine. Para ello, seguiremos el cuestionario indicado a continuación y dejaremos registro de las respuestas en la bitácora.

Análisis del desempeño del modelo

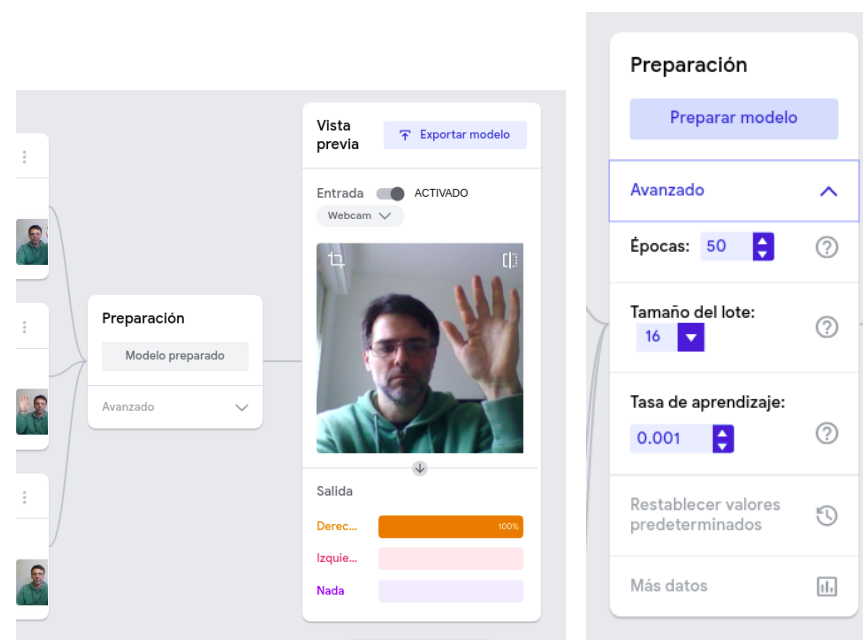
Ya sea usando la previsualización, a la derecha del proyecto cuando entrenamos el modelo, o usando el enlace generado, podemos verificar cualitativamente si nuestro modelo funciona como esperamos o no. En particular deberíamos verificar si:

- Al levantar la mano derecha, ¿el modelo detecta en la mayoría de los casos que la mano levantada es la derecha?
- Al levantar la mano izquierda, ¿el modelo detecta en la mayoría de los casos que la mano levantada es la izquierda?
- Cuando no levantamos ninguna mano, ¿el modelo detecta en la mayoría de los casos que no hay ninguna mano levantada?
- En caso de haber errores, ¿por qué creen que ocurren? ¿qué factores llevan a que la computadora se equivoque en la detección?
- ¿Qué pasa si lo que le mostramos a la computadora no se parece a ninguna de las fotos de entrenamiento, que tomamos previamente para generar el modelo?

Avanzado - Análisis de la precisión del modelo y su matriz de confusión

Dentro del propio Teachable Machine, es posible ver un poco más acerca del desempeño del modelo generado. Para ello, debemos seleccionar la opción

Debajo del botón que usamos para generar el modelo, y dentro de ella acceder a *Más datos*.



En esta sección podemos ver más detalles acerca del proceso de generación del modelo, lo que comúnmente se denomina **el entrenamiento del modelo**. Además, es posible conocer más acerca del desempeño del modelo generado, que es en lo que nos vamos a concentrar aquí. En particular nos vamos a detener en las opciones *Precisión por clase* y *Matriz de confusiones*.

Más datos



Aquí tienes algunos gráficos que pueden ayudarte a analizar el nivel de rendimiento de tu modelo.

No te preocupes si al principio no le encuentras mucho sentido. No necesitas nada de esto para usar Teachable Machine (y la mayoría de la gente no utiliza estos recursos).

Vocabulario

Precisión por clase

Calcular la precisión por clase

Matriz de confusiones

Calcular la matriz de confusiones

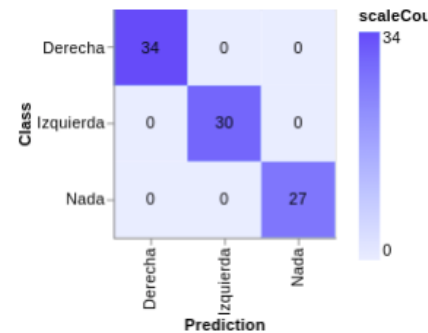
Precisión por época



Precisión por clase

CLASS	ACCURACY	# SAMPLES
Derecha	1.00	34
Izquierda	1.00	30
Nada	1.00	27

Matriz de confusiones



Para hacer la evaluación de desempeño, lo que se hace habitualmente es lo siguiente:

- Del conjunto de muestras de entrenamiento, se guardan algunas para probar el modelo.
- Una vez generado el modelo, se usan esas muestras que se separaron (denominadas conjunto de test) para evaluar el funcionamiento del modelo.
- En base al resultado que se obtiene para cada muestra, para las cuales se sabe cuál debería ser la salida correcta, se calcula la precisión y la matriz de confusión.

En la sección *Vocabulario* podemos ver el detalle de cómo se calculan estas cosas en Teachable Machine. La precisión, no es otra cosa que la proporción de muestras que se clasifican de manera correcta. Para un modelo nos puede interesar tanto la precisión para cada clase (como lo muestra Teachable Machine), como un promedio de todas ellas si queremos reducir la expresión del desempeño del modelo a un único parámetro.

Por otra parte, la matriz de confusión contiene todos los casos posibles si consideramos las clases a las que corresponden los datos, y las que asigna el modelo entrenado. De esta forma, resulta una forma útil de visualizar cómo funciona el modelo, y en caso de cometer errores (cosa que no ocurre en el ejemplo anterior) dónde se concentran la mayoría de ellos (por ejemplo, puede darse una confusión habitual entre dos clases del modelo, y habrá que ver cómo mejorar el entrenamiento para solucionarlo).

Anexo 3 - ¿Qué problemas podría traer la IA?

Etapas 4 - Impacto social y temas éticos asociados a la IA

Debate sobre el impacto social y ético de usar IA

La idea de esta actividad es problematizar con los estudiantes las implicancias que tiene el uso de IA en diversos ámbitos. A partir de esta instancia de debate, donde se plantean distintos temas para analizar y discutir en grupos, se busca concientizar acerca los aspectos no técnicos, pero igualmente relevantes, a la hora de implementar sistemas de IA.

1. Cada grupo de estudiantes toma una ficha y discute sobre la temática planteada, buscando una respuesta de consenso entre los distintos integrantes del grupo.
2. Después que todos los grupos discutieron por separado, se vuelve a juntar toda la clase para compartir las preguntas y respuestas, de manera que todos los estudiantes expongan sus opiniones sobre el tema, ya sea concordando o no con las respuestas de sus compañeros.
3. Es importante aclarar que no existen respuestas “correctas” o “equivocadas”, dado que todos los temas planteados resultan sumamente subjetivos, generando así posiciones encontradas aún entre especialistas en la materia.

Lista de preguntas para el debate:

- ¿Quién se responsabiliza en un accidente de tránsito causado por un vehículo autónomo?
- La inteligencia artificial, ¿se convertirá en algo más inteligente que los propios humanos?

- ¿Qué actividades realizan las computadoras actualmente mejor que los humanos?
- ¿Las computadoras pueden usar datos personales de los humanos para aprender?
- ¿Podrá la inteligencia artificial sustituir a los humanos en todo tipo de trabajos?
- ¿Puede la inteligencia artificial juzgar los casos de la misma forma que lo hace un juez?
- ¿Qué reglas definirían para un robot?
- ¿Confiarían en un robot para tomar decisiones por ustedes?
- ¿Cualquier persona puede programar una computadora para que tome decisiones?
- ¿Un médico, podría ser sustituido por una máquina?
- ¿Ustedes creen que las computadoras podrían definir sus propias reglas, ignorando las de los humanos?
- Un político, ¿podría ser sustituido por una máquina?

Jugando con la máquina moral

En este caso la idea es que trabajen, ya sea en grupos o individualmente, en las situaciones planteadas en la [máquina moral](#). Como verán, allí se plantean diferentes situaciones donde un vehículo autónomo debe resolver entre dos posibles opciones, ambas con consecuencias dramáticas en términos de vidas humanas y/o de otros animales. El objetivo es problematizar este tipo de situaciones, y ver cómo hay aspectos asociados a la incorporación de IA en la vida cotidiana que no son de fácil solución, y no necesariamente implican temas técnicos, sino más bien de índole moral o ética.

Anexo 4 - Conceptos básicos de programación

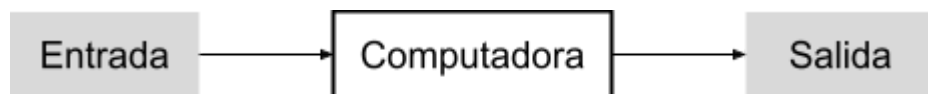
En este anexo se presentan algunos conceptos básicos de programación, que dependiendo de la experiencia previa en la temática que tengan los estudiantes, podrán ser necesarios o no para avanzar en las distintas etapas de la propuesta. Estos conceptos serán aplicados en el videojuego a desarrollar, los distintos bloques que lo componen y alguna condición que determine si una salida es correcta.

La computadora como máquina que procesa información de entrada y produce información de salida. En el caso del videojuego los mensajes de acierto o fallo son ejemplos de **salida de información**.

¿Cuándo entra información a la computadora al jugar el videojuego?

¿Qué sucede después? ¿Se les ocurren otras situaciones en las que suceda esta secuencia al usar una computadora?

Se recupera la noción de **evento como entrada de información** para establecer analogías e identificar un flujo de información en ambos sentidos entre usuario y máquina. También se reconstruye la secuencia en que el usuario levanta una u otra mano, la computadora lo detecta, desplaza en la pantalla el personaje según corresponda, y finalmente muestra el acierto o el fallo. A partir de estas interacciones, se construye entre todos una caracterización de **computadora como una máquina capaz de recibir información, procesarla y generar nueva información de salida**. Se puede resumir esta idea en el **esquema clásico** (ver figura debajo) de computadora con entrada y salida, preguntando cuáles son los dispositivos que permiten el ingreso de información en el juego (el ratón, el teclado, la cámara) e incorporando el cartel de acierto o error como salida. **Los estudiantes proponen otras situaciones en las que identifiquen ingreso, procesamiento y salida de información.**



El programa como responsable de la relación entre la información de entrada y la de salida.

¿La salida es siempre la misma? ¿De qué depende?

Los estudiantes analizan las situaciones que propusieron e identifican que la salida depende tanto de información de entrada (qué mano levanta el jugador) como de las reglas escritas en el programa (para indicar cuándo fue un acierto o un fallo). A partir de esta observación se completa el esquema agregando el **programa** en el interior de la computadora como un **componente** que, si bien **no es tangible es fundamental para determinar qué hace y para qué podemos usar una computadora**.

Los estudiantes agregan situaciones en las que identifiquen ingreso, egreso y procesamiento de información con un programa. Por ejemplo el desbloqueo de un teléfono con una contraseña, en el que, además, se puede identificar una alternativa condicional:

- Información de entrada: la contraseña que escribe el usuario.
- Procesamiento: decidir si la contraseña que ingresó el usuario es la correcta; Si es la correcta, entonces desbloquear el teléfono; Si es incorrecta, entonces, pedirle que reintente.
- Información de salida: pantalla principal del teléfono en el caso de que la contraseña sea correcta, mensaje de error en caso de que sea incorrecta.

La alternativa condicional como regla.

¿Qué tipos de reglas nos permite expresar la alternativa condicional?

¿Qué situaciones de nuestra vida cotidiana podríamos expresar con esta estructura?

Los estudiantes identifican que fue posible incorporar reglas del tipo “**Si** se cumple una condición **entonces** realizar una acción”. Se propone buscar ejemplos de estas situaciones en nuestra vida cotidiana y en otros juegos.

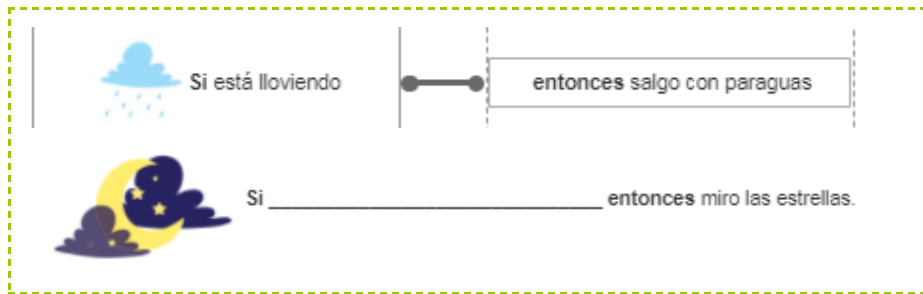


Recursos: para profundizar sobre alternativa condicional

-El video [PCtubers 6 - Condicionales El desafío del huevo](#).

-La actividad interactiva que remite a ejemplos de la vida cotidiana expresados como alternativas condicionales.¹

¹ Adaptación de la actividad del Manual para docentes “[Ciencias de la Computación para el Aula, 1er Ciclo de Primaria](#)” (pág 167) disponible en <http://program.ar/manual-primer-ciclo-primaria>.



Variables: las cajitas personales de la computadora

Existen diversos casos donde es necesario para la computadora guardar información (ej. un número) que luego se utilizará para otra tarea más adelante. En el juego de ejemplo, el número sorteado debe guardarse, para saber luego cuando el jugador responda si el número es par o impar, si la respuesta fue correcta o no. Algunas preguntas para discutir esto en clase con el juego de ejemplo serían:

¿En qué momento se sortea el número? ¿Cuándo necesitamos conocer el número elegido? ¿Cómo podemos hacer para saber qué número se eligió al momento de decidir si el jugador acierta o falla?

Cuando comienza el juego, la computadora realiza el sorteo del número y lo muestra al jugador en pantalla. Dicho número debe conservarse de alguna manera para usarse en el futuro, para decidir si el jugador al dar su respuesta acierta o falla. La necesidad de la computadora de **recordar** la podemos asociar a la **noción de variable** como herramienta de programación. Una **variable**, se puede pensar como una cajita con un nombre que tiene la computadora, donde se guarda un valor para no "olvidarlo" o perderlo. En nuestro juego de ejemplo, podemos pensar en una cajita donde ponemos el número sorteado, que usaremos luego cuando el jugador responda, para chequear si la respuesta es correcta o no.

Idea para explicar qué es una variable

Es posible realizar una actividad para facilitar la incorporación del concepto de variable como un espacio de almacenamiento. Utilizar una cajita etiquetada y un papel dentro, donde vamos a anotar un número, para luego recuperarlo o modificarlo cuando sea necesario.

Le pedimos a un estudiante que comience anotando en el papel un número cualquiera del 1 al 10. Luego, otro estudiante debe abrir la caja y fijarse si el número anotado es par o impar. Podemos repetirlo varias veces, de manera de mostrar que en cada pasada el número puede ir cambiando (según lo que elija cada estudiante) pero la caja siempre será la misma.

Sortear un número aleatorio y guardarlo en una variable

Para cerrar la presentación del concepto de variable, se puede aprovechar a presentar la función necesaria para sortear números aleatorios. Mediante el bloque **número aleatorio entre () y ()** podemos mostrar cómo inicializar una variable (es decir, una cajita a la que le pondremos un nombre) sorteando un número del 0 al 100 (el cual pondremos dentro de la cajita). Luego podemos simplemente ver cuál es el número (o mejor dicho, el contenido de la cajita), con algún bloque que permita mostrar la variable en la pantalla. Haciendo esto varias veces, veremos que si bien la cajita tiene siempre el mismo nombre (ya que la variable siempre es la misma), el número irá cambiando en función del resultado de cada sorteo.

Decidir si se produjo un acierto o un fallo (repaso condicionales)

Los estudiantes deberán comprender la relación entre el contenido de la variable (número sorteado) y la consigna del juego (identificar si es par o impar) para hacer que el programa decida si se produjo un acierto o un error. En la introducción ya se listaron todos los casos posibles de aciertos o errores, por lo que ahora veremos cómo se implementa esto mediante el uso de **alternativas condicionales**. Para ello habrá que asociar un acierto o error con las condiciones que escribieron e identificar los bloques a utilizar para implementarlo. En el juego diseñado por los estudiantes, las reglas hayan formulado según la consigna de su propio juego, determinará si su programa deberá hacer uso o no de **una estructura condicional anidada**.

Atención

Anidar estructuras condicionales e interpretarlas correctamente puede resultar transparente para algunos estudiantes pero críptico para otros, dependiendo del nivel de abstracción con el que estén acostumbrados a razonar y de la práctica previa con este tipo de bloques. La prioridad de esta propuesta de trabajo es la comprensión profunda por parte de los estudiantes de los programas que ellos mismos construyen y las decisiones que toman en el proceso. **Por ese motivo, es más valioso que trabajen con estructuras más simples pero sobre las que pueden proceder de manera analítica a que ensayen la prueba y el error con una estructura más compleja.**

Para implementar las condiciones que determinan si el jugador acierta o falla, en general será necesario expresar en términos de las operaciones matemáticas disponibles la regla invariante asociada al contenido matemático. Se sugiere que el docente recuerde ciertas asociaciones entre las condiciones utilizadas para expresar cada resultado posible (ej: “el número es par”) y las operaciones disponibles (“el resto de dividirlo por 2 es 0”). Para la implementación se deberá introducir el significado del bloque **módulo** y su utilidad para distinguir si un número es par o impar. Se sugiere escribir primero la fórmula en el pizarrón o en un papel y luego construirla en Scratch con la expresión de la condición (ej. para que el número sea par `< <_> módulo <2> > = <0>`).

Luego de ilustrar el uso de condicionales con el juego de ejemplo, también se pueden mostrar otras variantes (ej. múltiplo de 3 o no, múltiplo de 7 o no, etc) que podrían ser usadas por los estudiantes en sus propios juegos.

Extender el juego a varias instancias consecutivas (bloques de repetición)

Si bien conceptualmente el juego ya estaría casi pronto, es posible introducir una extensión sencilla que permite introducir otro tipo de bloques muy importante en la programación, como son los bloques de repetición.

Veremos como ejemplo una versión avanzada del juego de ejemplo. En este caso, el juego permite jugar infinitas veces consecutivas. Existe ahora un

contador con el puntaje acumulado que obtiene el jugador, quien ante cada acierto suma 100 puntos. Aquí tendremos por tanto otro caso de uso de variables, ya que en cada nueva ronda se deben sumar los nuevos puntos al resultado anterior.



Variante de complejidad

Puede agregarse que el jugador pierda 20 puntos ante cada fallo. Si los estudiantes no conocen la aritmética de los números negativos, los docentes deben mencionar que una manera de restar un valor a una variable es usar el bloque para sumar y colocar “un menos adelante” del número a sumar (ej. ante cada fallo suma -20 puntos, lo que equivale a perder 20 puntos).

El uso de una nueva variable hace también necesario resaltar la **importancia del nombre** que le dieron a la variable donde se guarda el resultado del sorteo. Es claro que esta nueva “cajita” con el puntaje total acumulado es otra, por lo que su nombre debe ser diferente.

Crear variable para contar los puntos y actualización del puntaje

¿Cuál debería ser el puntaje al iniciar el juego? ¿Hay alguna parte del programa que realice acciones en ese preciso momento? ¿Es siempre ese mismo valor? ¿Cómo podríamos asignárselo?

¿En qué momento del juego tiene que cambiar el puntaje? ¿A continuación de qué otro suceso debe ocurrir? ¿Con qué bloques del programa se corresponde ese momento?

Si no se inicializa la variable al comenzar el programa, los estudiantes notarán que la nueva partida comienza con el puntaje de la partida anterior. Se los orienta a identificar que el mejor momento para darle un valor inicial a una variable es al inicio del programa. Con respecto al valor, observan que debería ser 0 cada vez que se ejecuta el programa, por lo cual es necesario el bloque **dar a [] el valor ()**.

Luego, para actualizar el puntaje, los estudiantes deben reconocer que actualizar con el puntaje acumulado en el bloque de alternativa condicional a continuación de la respuesta del jugador.



Variantes de complejidad

Visibilidad del puntaje ¿Está visible el puntaje? ¿Podemos mostrarlo solo cuando acertamos?

Condiciones sobre el puntaje: A los estudiantes que estén familiarizados con el manejo de variables, la alternativa condicional y la construcción de condiciones numéricas, se les puede proponer que intenten que el puntaje nunca se vuelva negativo o que aparezcan mensajes al alcanzar determinados puntajes.

Más preguntas para trabajar en el diseño del juego

- ¿Qué es lo más importante que debe tener un videojuego?
¿Conocen algún juego sin objetivo? ¿Sería entretenido?
Todos los juegos tienen un objetivo o una meta que alcanzar (conseguir el mayor puntaje, llegar al último nivel, jugar el mayor tiempo posible sin perder).
- ¿Se imaginan un juego en el que el jugador no hace nada?
¿Dirían que es un juego?
En todos los juegos, hay interacción con el jugador: las acciones de quien juega (mover objetos, presionar botones, responder preguntas) determinan el desarrollo del juego.
- ¿Cómo se imaginan el juego más fácil del mundo? ¿Es interesante?
¿Sigue siendo un juego?
Todos los juegos tienen un desafío: algo que dificulta al jugador alcanzar el objetivo.
- ¿Cuánto dura el juego? ¿Cómo está establecido en las reglas?
Las reglas del juego establecen un límite para la duración del juego (se termina cuando se cumple cierto tiempo, cuando se agotan los intentos, cuando se choca contra un obstáculo, etc.).
- ¿Qué usamos para jugar un videojuego?
Aquí retomamos lo que vimos anteriormente sobre cómo la

computadora puede recibir información de entrada (ratón o el teclado, cámara, micrófono) y cómo muestra la salida (pantalla, parlantes).

- ¿Cómo se hace un videojuego? ¿Quién los hace?
En este caso se trabaja en Scratch, como una herramienta para construir videojuegos, que permite agregar personajes, controlar su movimiento, etc. Apoyándose en experiencias previas de los estudiantes, se señala que el funcionamiento del juego está dado por bloques que contienen instrucciones para los elementos que conforman el videojuego. Scratch interpreta y usa la computadora para ejecutarlos. En definitiva, **los videojuegos son programas de computadora**. Los juegos de Scratch, en su mayoría, están contruidos por otros estudiantes.
- ¿Cuándo debe aparecer el cartel de acierto? ¿Y el de error? ¿En qué lugar del programa hay que agregarlos?
Los estudiantes identifican a partir del ejemplo el comportamiento y lo que debe suceder en cada caso. Recordar que la computadora es una máquina que sigue instrucciones al pie de la letra, permite explicar que las frases deben estar pensadas en términos que existan en Scratch.