

Modelo de aprendizaje adaptativo basado en plataforma de código abierto

Adaptive Learning Model Based on an Open-Code Platform

Raúl Quintanar Casillas
Universidad Autónoma de Querétaro
rquintanaro2@alumnos.uaq.mx

Ma. Sandra Hernández López
Universidad Autónoma de Querétaro
ma.sandra.hernandez@uaq.mx

Ma. Teresa García Ramírez
Universidad Autónoma de Querétaro
teregar@uaq.mx

RESUMEN

A través de la capacitación, las organizaciones e instituciones buscan potencializar las habilidades y competencias de alumnos y empleados, según el caso, con el fin de aumentar su desempeño e impactar de forma positiva el aprendizaje, la productividad o la efectividad laboral. En ese sentido, diversas organizaciones e instituciones han optado por utilizar *Learning Management Systems* (LMS), los cuales permiten que la capacitación se entregue de forma sincronizada o asíncrona y con recursos de aprendizaje actualizados. En los últimos años, los LMS han evolucionado hacia esquemas de Aprendizaje Adaptativo (AA) que se apoyan en el uso de la inteligencia artificial (IA). Varios autores han explorado el mundo de la IA en el AA desde diferentes enfoques. Estos modelos han tenido cierto grado de éxito, sin embargo, no consideran la implementación de los LMS en plataformas de código abierto que utilicen herramientas nativas para la IA, lo cual permita contar con sistemas de aprendizaje robustos, escalables, con alto desempeño y un tiempo de vida prospectado más prolongado. El presente trabajo propone un modelo de AA que se implemente en una plataforma de código abierto con herramientas IA nativas y que se verifique su efectividad a través de la simulación de diez alumnos de diferentes características.

Palabras clave: Sistemas de gestión de aprendizaje, inteligencia artificial, adaptativo

ABSTRACT

In order to increase their performance and positively influence either learning, productivity or work effectiveness both institutions and organizations, through training, are looking into enhancing skills and competencies of students and employees. Thus, various organizations and institutions have implemented *Learning Management Systems* (LMS), which allow synchronous or asynchronous training, and with updated learning resources. In recent years, LMS have evolved towards Adaptive Learning (AL) schemes supported by the

use of artificial intelligence (AI). Various authors have explored the world of AI in AL from different perspectives. These models have succeeded in certain degree; however, they do not consider the implementation of LMS in open-source platforms, which use native and full-compatible tools for AI. This allows having robust, scalable learning systems with high performance and prospected long lifetime.

This work proposes an AL model approach implemented in an open-source platform supported by native AI tools; model's performance is verified through the simulation of ten students with a variety of profiles.

Keywords: Learning management systems, artificial intelligence, adaptive

INTRODUCCIÓN

El aprendizaje personalizado identifica las características individuales de los alumnos, pero deja de adaptarse al progreso continuo de la capacidad de un alumno para realizar tareas. Por su parte, el aprendizaje adaptativo se puede implementar basándose en el desempeño del estudiante sin necesidad de identificar la información personal relevante, ya que las características y preferencias individuales pueden variar con el tiempo y al afectar el progreso y el rendimiento del estudiante. Las competencias de un alumno se pueden evaluar con mayor claridad gracias a soluciones de aprendizaje adaptativo (García, 2017).

Un número creciente de investigaciones en torno al aprendizaje adaptativo han aparecido en los últimos años (Xie, Chu, Hwang & Wang, 2019). De acuerdo con la evaluación de Quintanar y Hernández (2020) de diferentes trabajos de investigación que proponen sistemas o modelos de esta naturaleza, se observa que es un común denominador de estas propuestas el contar con componentes tales como el Modelo del Alumno, el Modelo de Objeto de Aprendizaje o Dominio y el Modelo de Adaptación o Adaptabilidad.

Cabe resaltar el trabajo de Balasubramanian y Magret (2018), que propone un modelo innovador de detección del estilo de aprendizaje a través del estudio de las habilidades cognitivas útiles para el procesamiento de información como son la memoria, la concentración, la percepción y el razonamiento lógico. En este mismo se propone que el modelo del aprendizaje tenga como entrada el nivel de habilidades cognitivas del alumno con el fin de conocerlas y desarrollarlas a través de su exposición a los recursos de aprendizaje (Carrillo, Tigre, Tubón y Sánchez, 2019) diseñados en el Modelo de Objeto de Aprendizaje.

A través de una encuesta de opinión, se realizó un mapeo entre las habilidades cognitivas y los objetos de aprendizaje, es decir, se conoce el estilo de aprendizaje en relación al nivel de habilidad de los alumnos. Si bien este mapeo resulta en cierta forma eficaz para la asignación de recursos de aprendizaje, no tiene un alto grado de precisión por su carácter subjetivo, ya que se establecen estilos de aprendizaje por niveles de habilidad y no de forma individual.

Otra limitante es la del Modelo de Adaptación que utiliza Balasubramanian, ya que emplea Matlab como herramienta de Inteligencia Artificial e induce a tener una flexibilidad limitada para el diseño, implementación y, sobre todo, ejecución por el procesamiento limitado en comparación a las plataformas de código abierto que utilizan herramienta para la IA nativas como es el caso de Python. El modelo se torna complejo ya que se agrega un control difuso para asignar objetos de aprendizaje.

De acuerdo con el portal Capterra, los mejores LMS catalogados para el 2021 son los relacionados en la Tabla 1, en la cual se indica en qué plataforma de programación fueron desarrollados. Como se puede observar ninguna de los LMS usa una plataforma de código abierto que emplee herramientas para la IA de forma nativa, lo que hace complejo y en algunos casos imposible la integración de herramientas avanzadas para la IA, o bien integran componentes simples con funcionalidades limitadas.

TABLA 1. LENGUAJES DE PROGRAMACIÓN UTILIZADOS EN PLATAFORMAS LMS MEJOR POSICIONADAS
FUENTE: ELABORACIÓN PROPIA CON BASE EN INFORMACIÓN DEL PORTAL CAPTERRA (2021) Y LOS PORTALES INDIVIDUALES DE LAS PLATAFORMAS LMS MOSTRADAS.

PLATAFORMA LMS	LENGUAJE BASE DE PROGRAMACIÓN
Academy	PHP
Adobe Captive	Action Script
Chamillo	PHP
Cornerstone LMS	Javascript
Desire to Learn (D2L)	HTML y Javascript sobre .NET
Docebo	JSON
Learnupon	Ruby, JS y Perl
Lessonly	Ruby y Shell
Moodle	PHP
Teachable	Liquid
Thinkific	Liquid Javascript
Trainual	Ruby y Shell

Para solventar estas deficiencias, el presente trabajo se enfoca en usar una plataforma de código abierto que maneja de forma nativa componentes para la inteligencia artificial y que se basa en el lenguaje de programación Python y las librerías para la IA de Keras-RL y Tensorflow. Estas últimas permitirán introducir al modelo propuesto las herramientas de aprendizaje por refuerzo y redes neuronales respectivamente.

El aprendizaje por refuerzo se basa en el Proceso de Decisión de Markov, en el cual existe un agente inmerso en un ambiente dentro del cual dicho agente tiene que ejecutar acciones con el fin de lograr un objetivo (Rothman, 2018). Cada que realiza una acción, el agente pasa de un estado i a un estado $i+1$, y el ambiente le proporcionará una recompensa positiva o negativa de acuerdo con el nuevo estado en que se encuentre.

Cuando el agente logre el objetivo será porque habrá alcanzado la recompensa máxima que es producto de la suma de las recompensas obtenidas de los estados anteriores por el que el agente transitó (Alzantot, 2017). En este punto, culmina un episodio. Cabe señalar que el agente no conoce su recompensa hasta después de tomar la acción, por lo tanto debe aprender qué acciones son las que lo llevan la máxima recompensa al probar diferentes o iguales acciones en varios episodios, hasta que transite por el menor número de estados, o los que en suma le gratifiquen la máxima recompensa (Sutton y Barto, 2017).

En otro punto, una red neuronal artificial (o simplemente red neuronal) simula el comportamiento del cerebro humano, que consiste en un conjunto de unidades conectadas entre sí denominadas neuronas (Mota, Juárez y Olguín, 2018). Las neuronas se agrupan por capas y en un modelo artificial simple no se interconectan pertenecientemente a la misma capa, sino que esta interconexión se da con las neuronas de las capas adyacentes. Se da una secuencia de interconexión de manera jerárquica donde la entrada de la red se sitúa en la parte izquierda, de izquierda a derecha se apilan las capas y la salida queda en la parte derecha de la red. Cada enlace neuronal tiene asociado un peso P_i y la unidad neuronal un sesgo S_i . Los pesos se multiplican a la función matemática asociada a la unidad neuronal determinada por la entrada y se suma al sesgo. En la Figura 1 se esquematiza una red neuronal.

Los pesos y los sesgos son calculados por la red neuronal de forma automática con el fin de obtener la salida Y . Dentro del aprendizaje automático, una red neuronal tiene de inicio datos de entrenamiento, que son una muestra de entradas y sus salidas correspondientes, para determinar los pesos y los sesgos. A estas segundas se les conoce como etiquetas. Una vez entrenado el modelo, las etiquetas son omitidas y con los pesos y sesgos que la red determinó se predicen las salidas.

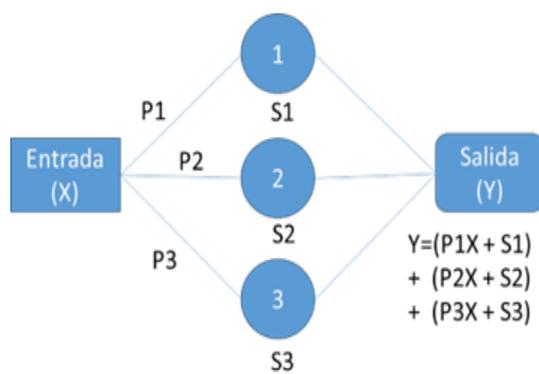


FIGURA 1. RED NEURONAL SIMPLE.

FUENTE: ELABORACIÓN PROPIA CON BASE EN MERCADO, PEDRAZA Y MARTÍNEZ (2015).

Con el uso de la plataforma de código abierto se hizo frente a la problemática planteada, la cual indica que “No hay un sistema LMS que esté basado en alguna plataforma de código abierto que permita el manejo transparente de librerías para la inteligencia artificial” y “No hay un modelo que realice un estudio granular de las habilidades cognitivas y los estilos de aprendizaje del alumno con el fin de brindarle un desarrollo cognitivo integral”.

El objetivo del trabajo es proponer un modelo de aprendizaje adaptativo que sea implementado en una plataforma de código abierto y que permita el uso transparente de herramientas para la inteligencia artificial a través del estudio de las habilidades cognitivas y estilos de aprendizaje del alumno. La hipótesis planteada afirma que, con un

modelo de aprendizaje adaptativo que se integró en una plataforma de código abierto basada en Python, se pueden implementar elementos de inteligencia artificial de forma más dinámica y eficiente, dotándole de una alta escalabilidad y personalización.

METODOLOGÍA

El modelo de aprendizaje adaptativo que se tomará como guía para definir la metodología es el de Balasubramanian y Magret (2018), el cual determina el estilo de aprendizaje del alumno con base en el nivel de habilidades cognitivas que son útiles para el procesamiento de información. Estas habilidades cognitivas son la memoria, la concentración, la percepción y el razonamiento lógico, cada una tiene tres niveles: bajo, medio y alto. Estos niveles corresponden a la escala taxonómica de Bloom, un nivel bajo corresponde a “recordar” y “comprender”, un nivel medio corresponde a “aplicar” y “analizar”, mientras que un nivel alto a “evaluar” y “crear”.

Balasubramanian y Magret (2018) emplean siete tipos de objetos de aprendizaje que corresponden al estilo de aprendizaje, los cuales son texto, audio, video, diagrama, gráfico, caso de estudio y simulación. Estos objetos de aprendizaje se mapean con los niveles de habilidad cognitiva mediante una encuesta de opinión y su análisis ANOVA de los resultados con el fin de conocer los objetos de aprendizaje más representativos para cada nivel de habilidad, lo que se realiza en el módulo del aprendiz. El módulo del objeto de aprendizaje contiene los recursos diseñados de enseñanza que se entregan a los alumnos. Finalmente, el módulo de adaptación integra los recursos de inteligencia artificial que le dan el carácter de adaptativo al modelo.

El presente trabajo de investigación se centra principalmente en el módulo de adaptación, empleando la técnica de aprendizaje profundo por refuerzo al aplicarse al contexto laboral, en específico a los empleados del Ayuntamiento de Huixquilucan. Para el módulo del aprendiz, se realizó una encuesta de opinión dirigida a 200 empleados, los cuales representan aproximadamente el 10 % de los servidores públicos que laboran en este lugar. En esta encuesta se le pregunta al empleado “¿Cuáles son las actividades que más le gusta realizar?”, es decir, con qué objeto de aprendizaje se identifica, y “¿en cuál de las cuatro habilidades cognitivas estudiadas se considera mejor?”.

A partir de los resultados de la aplicación de la encuesta, se clasifican a los empleados con base en la edad y la experiencia en cuatro grupos diferentes, se realiza el mapeo y análisis ANOVA antes mencionados sobre estos grupos y se establecen los objetos de aprendizaje más representativos para cada nivel de habilidad. En segundo lugar, con la información que arroja el ANOVA, se realiza el diseño del módulo de adaptabilidad que contiene tres secciones: el aprendizaje por refuerzo, la red neuronal y el agente de aprendizaje profundo.

Una vez con el diseño, se realiza la configuración del entorno del sistema que simula a los alumnos (Shawky y Badawi, 2018), la programación o desarrollo y finalmente su implementación al realizar los ajustes necesarios a la red neuronal como son la tasa de aprendizaje y la cantidad de neuronas requeridas para obtener el mejor desempeño posible. El modelo de Balasubramanian y Magret (2018) se puede observar en la Figura 2.

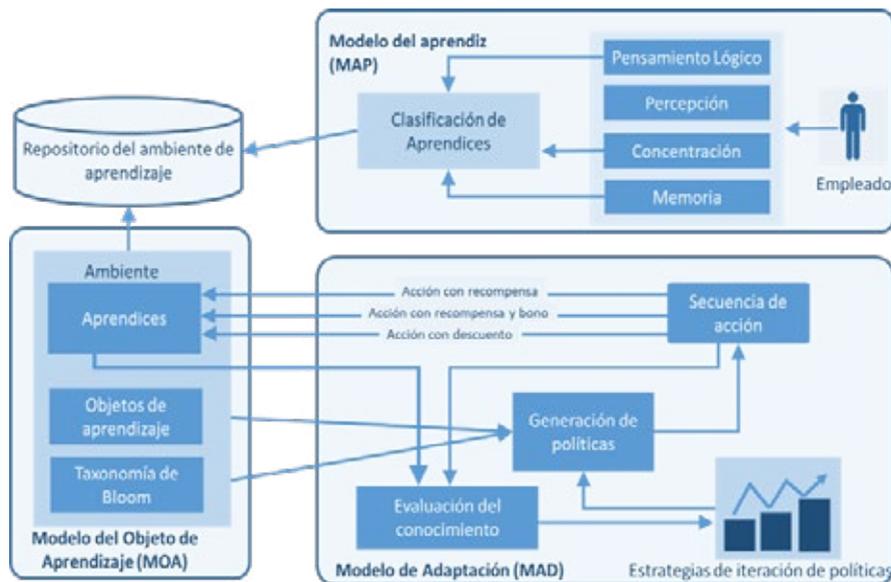


FIGURA 2. MODELO PARA LA DETECCIÓN DE HABILIDADES COGNITIVAS Y ESTILOS DE APRENDIZAJE APOYADO EN EL APRENDIZAJE ADAPTATIVO.

FUENTE: ELABORACIÓN PROPIA CON BASE EN BALASUMBRAMANIAN Y MARGRET (2018).

RESULTADOS

Módulo del Aprendiz

Al seguir la metodología propuesta por Balasubramanian y Magret, se aplicó la encuesta de opinión a 200 sujetos de estudio, en la que se conocen los objetos de aprendizaje, con los que mejor se desempeñan, así como la percepción que tienen con respecto a sus habilidades cognitivas. Con las respuestas proporcionadas, se realizó el análisis ANOVA para obtener el peso de los objetos de aprendizaje por cada habilidad cognitiva, es decir, el porcentaje de influencia que tienen para cada habilidad cognitiva. Los resultados se muestran en las tablas 1a – 4b.

TABLA 1A. RESULTADOS DEL ANÁLISIS ANOVA PARA LA MEMORIA.

FUENTE: ELABORACIÓN PROPIA.

GRUPOS	CUENTA	SUMA	PROMEDIO	VARIANZA	% DE INFLUENCIA
Texto	4	18	4.5	5.66666667	21.73 %
Caso de Estudio	4	28	7	6.66666667	25.56 %
Grafico	4	13	3.25	0.25	0.96 %
Video	4	10	2.5	1	3.83 %
Audio	4	17	4.25	2.91666667	11.18 %
Simulación	4	31	7.75	9.58333333	36.74 %
Diagrama	4	4	1	0	0.00 %

TABLA 1B. RESUMEN DEL ANÁLISIS ANOVA PARA LA MEMORIA.

FUENTE: ELABORACIÓN PROPIA.

ORIGEN DE LAS VARIACIONES	SUMA DE CUADRADOS	GRADOS DE LIBERTAD	PROMEDIO DE LOS CUADRADOS	F	PROBABILIDAD	VALOR CRÍTICO PARA F
Entre grupos	137.85	6	22.98	6.17	0.0007521	2.57271
Dentro de los grupos	78.25	21	3.73			
Total	216.11	27				

TABLA 2A. RESULTADOS DEL ANÁLISIS ANOVA PARA LA CONCENTRACIÓN.

FUENTE: ELABORACIÓN PROPIA.

GRUPOS	CUENTA	SUMA	PROMEDIO	VARIANZA	% DE INFLUENCIA
Texto	4	33	8.25	0.25	0.88 %
Caso de Estudio	4	21	5.25	4.25	15.04 %
Grafico	4	25	6.25	14.91666667	52.80 %
Video	4	12	3	5.33333333	18.88 %
Audio	4	17	4.25	0.91666667	3.24 %
Simulación	4	35	8.75	0.91666667	3.24 %
Diagrama	4	10	2.5	1.66666667	5.90 %

TABLA 2B. RESUMEN DEL ANÁLISIS ANOVA PARA LA CONCENTRACIÓN.

FUENTE: ELABORACIÓN PROPIA.

ORIGEN DE LAS VARIACIONES	SUMA DE CUADRADOS	GRADOS DE LIBERTAD	PROMEDIO DE LOS CUADRADOS	F	PROBABILIDAD	VALOR CRÍTICO PARA F
Entre grupos	142.21	6	23.70	5.87	0.001007464	2.572712
Dentro de los grupos	84.75	21	4.04			
Total	226.96	27				

TABLA 3A. RESULTADOS DEL ANÁLISIS ANOVA PARA LA PERCEPCIÓN.

FUENTE: ELABORACIÓN PROPIA.

GRUPOS	CUENTA	SUMA	PROMEDIO	VARIANZA	% DE INFLUENCIA
Texto	4	19	4.75	5.58333333	22.26 %
Caso de Estudio	4	34	8.5	1.66666667	6.64 %
Grafico	4	17	4.25	1.58333333	6.31 %
Video	4	29	7.25	8.25	32.89 %
Audio	4	22	5.5	1.66666667	6.64 %
Simulación	4	36	9	6	23.92 %
Diagrama	4	6	1.5	0.33333333	1.33 %

TABLA 3B. RESUMEN DEL ANÁLISIS ANOVA PARA LA PERCEPCIÓN.

FUENTE: ELABORACIÓN PROPIA.

ORIGEN DE LAS VARIACIONES	SUMA DE CUADRADOS	GRADOS DE LIBERTAD	PROMEDIO DE LOS CUADRADOS	F	PROBABILIDAD	VALOR CRÍTICO PARA F
Entre grupos	166.86	6	27.81	7.76	0.00017315	2.57271164
Dentro de los grupos	75.25	21	3.58			
Total	242.107143	27				

TABLA 4A. RESULTADOS DEL ANÁLISIS ANOVA PARA EL RAZONAMIENTO LÓGICO.**FUENTE: ELABORACIÓN PROPIA.**

GRUPOS	CUENTA	SUMA	PROMEDIO	VARIANZA	% DE INFLUENCIA
Texto	4	5	1.25	1.58333333	20.21 %
Caso Estudio	4	9	2.25	0.91666667	11.70 %
Grafico	4	9	2.25	1.58333333	20.21 %
Video	4	5	1.25	2.25	28.72 %
Audio	4	5	1.25	0.25	3.19 %
Simulación	4	10	2.5	1	12.77 %
Diagrama	4	5	1.25	0.25	3.19 %

TABLA 4B. RESUMEN DEL ANÁLISIS ANOVA PARA EL RAZONAMIENTO LÓGICO.**FUENTE: ELABORACIÓN PROPIA.**

ORIGEN DE LAS VARIACIONES	SUMA DE CUADRADOS	GRADOS DE LIBERTAD	PROMEDIO DE LOS CUADRADOS	F	PROBABILIDAD	VALOR CRÍTICO PARA F
Entre grupos	8.21	6	1.37	1.22	0.3337726	2.5727116
Dentro de los grupos	23.5	21	1.12			
Total	31.71	27				

MÓDULO DEL OBJETO DE APRENDIZAJE

Con los resultados del análisis ANOVA se obtienen los objetos de aprendizaje más representativos para cada habilidad cognitiva de acuerdo con su porcentaje de influencia, que puede observarse en las Tablas 1a, 2a, 3a y 4a. En la Tabla 5 se tiene el resumen del mapeo de los objetos de aprendizaje y las habilidades cognitivas.

TABLA 5. OBJETOS DE APRENDIZAJE INFLUYENTES POR CADA HABILIDAD COGNITIVA.
FUENTE: ELABORACIÓN PROPIA.

HABILIDAD COGNITIVA	OAI	OA2	OA3
Memoria	Simulación	Caso de Estudio	Texto
Concentración	Grafico	Video	Caso de Estudio
Percepción	Video	Simulación	Texto
Razonamiento	Video	Texto	Grafico

Se pueden establecer reglas para entregar los objetos de aprendizaje adecuados para desarrollar las habilidades cognitivas a la vez que se adquiere conocimiento en un área determinada. Al seguir la metodología de Balasubramanian y Magret, las reglas quedarían establecidas conforme a la Tabla 6. Las herramientas para la inteligencia artificial consumen muchos recursos de procesamiento y memoria de los equipos de cómputo utilizados, por lo cual se utilizó una estrategia de agrupación de objetos de aprendizaje en paquetes de aprendizaje (PA).

TABLA 6. REGLAS PARA LA ASIGNACIÓN DE OBJETOS DE APRENDIZAJE.
FUENTE: ELABORACIÓN PROPIA.

SI EL USUARIO TIENE:	LOS OBJETOS DE APRENDIZAJE Y SU NIVEL A ENTREGAR SON:		
Memoria baja	Simulación baja	Caso de Estudio medio	Texto alto
Memoria media	Simulación media	Caso de Estudio alto	Texto medio
Memoria alta	Simulación alta	Caso de Estudio medio	Texto bajo
Concentración baja	Gráfico bajo	Video medio	Caso de Estudio alto
Concentración media	Gráfico medio	Video alto	Caso de Estudio medio
Concentración alta	Gráfico alto	Video medio	Caso de Estudio bajo
Percepción baja	Video bajo	Simulación medio	Texto alto
Percepción media	Video medio	Simulación alto	Texto medio
Percepción alta	Video alto	Simulación medio	Texto bajo
Razonamiento bajo	Video bajo	Texto medio	Gráfico alto
Razonamiento medio	Video medio	Texto alto	Gráfico medio
Razonamiento alto	Video alto	Texto medio	Gráfico bajo

En la Figura 2 se observan los paquetes de aprendizaje, los objetos de aprendizaje contenidos y el grado de contenido (alto, medio, bajo) que se entregan a los alumnos simulados. Una vez establecidos los PA, se utilizan en el Módulo de Adaptabilidad para que el sistema con base en las reglas establecidas realice la entrega de contenidos a los estudiantes que, para efectos de esta investigación, fueron simulados.

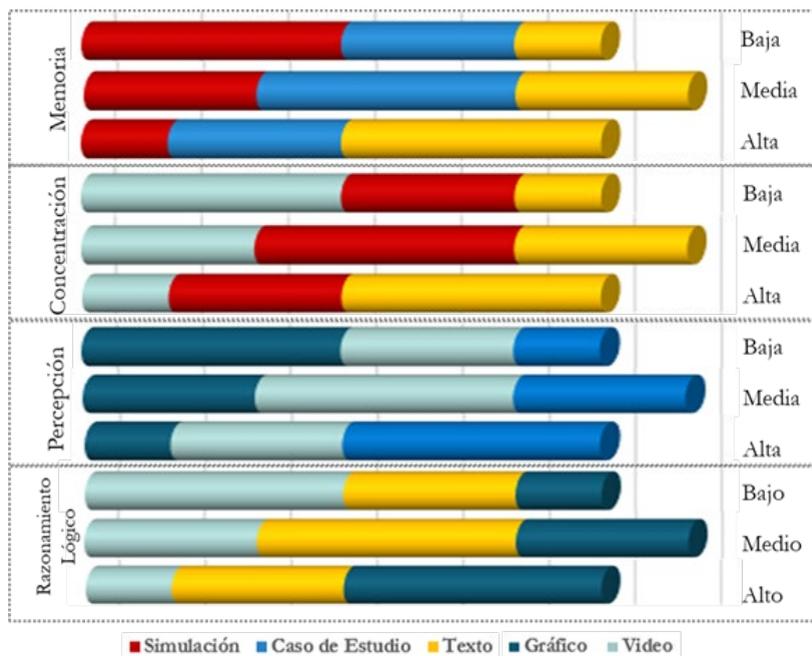


FIGURA 2. PAQUETES DE APRENDIZAJE A ENTREGAR A LOS ALUMNOS SIMULADOS.
FUENTE: ELABORACIÓN PROPIA.

Módulo de Adaptabilidad

Para lograr la adaptabilidad, se utiliza la técnica de aprendizaje por refuerzo, con la cual se establece al aula virtual como ambiente. En esta aula virtual conviven el agente, los objetos de aprendizaje y el alumno. El agente es el encargado de entregar los objetos de aprendizaje adecuados para el desarrollo de habilidades cognitivas y del conocimiento al alumno. Dentro del programa que se implementa en la plataforma de código abierto (*Python*), el aula virtual es una clase que contiene funciones tales como la inicialización, el paso que corresponde a la acción que se toma, y el restablecimiento al estado inicial. Al conjunto de estados en los que el agente puede estar se le llama espacio de observación (*observation space*). En la inicialización del ambiente, se definen 12 acciones que corresponden a las reglas de asignación de la Tabla 6. En las Figuras 3, 4 y 5 se observa el pseudocódigo utilizado en la configuración del ambiente que se corresponden a la inicialización, el paso o acción, y el restablecimiento cuando se inicia un nuevo episodio.

```

class AulaVirtual(Env):
    def __init__(self): # Función de inicialización
        self.action_space = Discrete(12) # Número de acciones = número de reglas
        # Espacio de Observación [Memoria, Concentración, Percepción, Razonamiento]
        # Niveles Bajo=0, Medio=1, Alto=2
        self.observation_space = [0, 0, 0, 0] -> [2, 2, 2, 2]

        estado_inicial = [0, 0, 0, 0]
        self.state = estado_inicial
        self.modulos_curso = 5 # Cinco módulos del curso

```

FIGURA 3. PSEUDOCÓDIGO DE LA DEFINICIÓN DEL AMBIENTE Y EL ESTADO INICIAL DEL AGENTE.
FUENTE: ELABORACIÓN PROPIA.

```

def step(self, action): # Función paso o acción

    memoria_ant, concentracion_ant, percepcion_ant, razonamientoLogico_ant = self.state[0],[1],[2],[3]

    # Se simula la mejora de las HC de acuerdo a la entrega de OAs con base a las reglas
    if memoria_ant == 0 and action == 0: memoria_act = 1
    elif memoria_ant == 1 and action == 1: memoria_act = 2
    elif memoria_ant == 2 and action == 2: memoria_act = 2
    else: memoria_act = memoria_ant
    if concentracion_ant == 0 and action == 3: concentracion_act = 1
    elif concentracion_ant == 1 and action == 4: concentracion_act = 2
    elif concentracion_ant == 2 and action == 5: concentracion_act = 2
    else: concentracion_act = concentracion_ant
    if percepcion_ant == 0 and action == 6: percepcion_act = 1
    elif percepcion_ant == 1 and action == 7: percepcion_act = 2
    elif percepcion_ant == 2 and action == 8: percepcion_act = 2
    else: percepcion_act = percepcion_ant
    if razonamientoLogico_ant == 0 and action == 9: razonamientoLogico_act = 1
    elif razonamientoLogico_ant == 1 and action == 10: razonamientoLogico_act = 2
    elif razonamientoLogico_ant == 2 and action == 11: razonamientoLogico_act = 2
    else: razonamientoLogico_act = razonamientoLogico_ant

    # ----- Se establece el nuevo estado y se da por tomado un módulo-----
    self.state = [memoria_act, concentracion_act, percepcion_act, razonamientoLogico_act]
    self.modulos_curso -= 1

    # ----- Se establece la política de aprendizaje por refuerzo para cada HC
    if HC_act > HC_ant: reward += 1
    elif HC_act == HC_ant: reward += 0
    else: reward -= 1

    # Se verifica si el alumno tiene niveles altos
    if HC == 2: done = True
    else: done = False
    return self.state, reward, done, info

```

FIGURA 4. PSEUDOCÓDIGO DE LA DEFINICIÓN DEL AMBIENTE Y EL ESTADO INICIAL DEL AGENTE.
FUENTE: ELABORACIÓN PROPIA.

```
def reset(self):
    # Resetear el estado - arreglo
    Para cada HC = 0 + random.randint(0, 1)

    # Resetear numero de módulos del curso
    self.modulos_curso = 5
    return self.state

episodios = 5 #En cada episodio el agente tiene un alumno diferente
```

FIGURA 5. PSEUDOCÓDIGO DE LA DEFINICIÓN DEL AMBIENTE Y EL ESTADO INICIAL DEL AGENTE.
FUENTE: ELABORACIÓN PROPIA.

La segunda sección del modelo se conforma por la red neuronal, la cual le dará al sistema el carácter de aprendizaje profundo permitiendo así mejorar su desempeño. La red para tener como entrada la clase del Aula Virtual, es decir, el ambiente, se esquematiza en la Figura 6. Esta red tiene como entradas las habilidades cognitivas y como salidas las acciones a tomar, que en este caso son las entregas de los paquetes de aprendizaje según las reglas de asignación. Se integra por dos o tres capas internas, dos densas con 8 y 24 unidades neuronales y una plana (*flatten*). En Python, con la función `model = build_model(states, actions)` se crea la red neuronal. El número de capas puede ajustarse para obtener un mejor desempeño del modelo de la red.

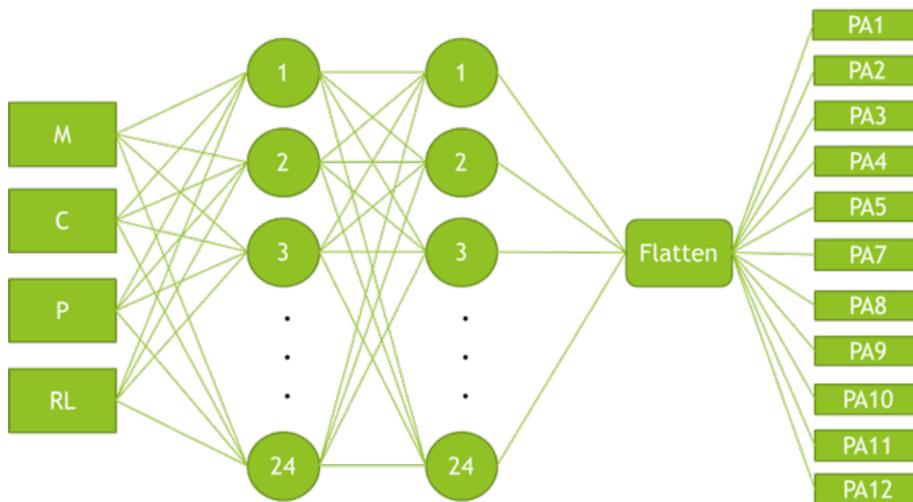


FIGURA 6. MODELO DE LA RED NEURONAL UTILIZADA EN EL MÓDULO DE ADAPTACIÓN.
FUENTE: ELABORACIÓN PROPIA.

En la tercer y última sección se realiza la implementación del agente con las librerías Keras RL (véase Figura 7). En esta sección del código se define la política que rige el aprendizaje del agente, la forma en que se utilizan los recursos de procesamiento del equipo de cómputo, y los parámetros de aprendizaje que utilizará el agente. Un parámetro importante es la tasa de aprendizaje (*learning rate*), la cual debe ajustarse para conseguir el mejor desempeño del agente.

```
def build_agent(model, actions):
    policy = BoltzmannQPolicy() #Se define la política que rige el aprendizaje del agente
    # Se define la forma en que se utilizan los recursos de procesamiento:
    memory = SequentialMemory(limit=10000, window_length=1)
    # Se define el agente, la red neuronal que usará, el conjunto de acciones y otros
    # parámetros de configuración ajustables
    dqn = DQNAgent(model=model, memory=memory, policy=policy,
                   nb_actions=actions, nb_steps_warmup=100, target_model_update=1000)
    return dqn
# %%
dqn = build_agent(model, actions) #Se crea el agente
# Se compila el agente con el optimizador Adam a una tasa de aprendizaje de 0.01:
dqn.compile(Adam(learning_rate=1e-2, amsgrad='true'), metrics=['mse'])
# Se realiza el entrenamiento del agente
dqn.fit(env, nb_steps=50000, visualize=False, verbose=1)
# %%
# Se prueba el desempeño del modelo a 10 episodios (alumnos)
scores = dqn.test(env, nb_episodes=10, visualize=False)
print(np.mean(scores.history['episode_reward']))
```

FIGURA 7. PSEUDOCÓDIGO DE LA DEFINICIÓN Y CREACIÓN DEL AGENTE.
FUENTE: ELABORACIÓN PROPIA.

En la Figura 8, se puede apreciar la salida que arroja como resultado el sistema adaptativo al simular la participación de 10 estudiantes en un curso de 5 módulos. Como antes se mencionó, cada estudiante representa un episodio distinto. En el ambiente se simula que el alumno puede obtener una mejora a una habilidad cognitiva por cada módulo del curso que toma, es decir, en un curso un máximo total de 5 mejoras que corresponde a una recompensa igual a 5. Se puede observar que las recompensas finales oscilan entre 4 y 5, lo cual indica que el sistema adaptativo está entregando en la mayoría de los casos, los paquetes de aprendizaje adecuados. La salida mostrada indica una eficiencia del 90% del sistema adaptativo.

Con algunos ajustes en el código, se puede imprimir el resultado detallado para cada alumno simulado en el que se observe su progreso conforme toma los diferentes módulos del curso. En la Figura 9 se observa el detalle del episodio 2, es decir, el alumno 2. La entrega del paquete de aprendizaje adecuado permitirá al alumno incrementar su nivel cognitivo y conocimiento. En el ejemplo se aprecia el desarrollo de un estado (Memoria baja, Concentración baja, Percepción media, Razonamiento bajo) a un estado (Memoria Media, Concentración alta, Percepción alta, Razonamiento medio).

```
Interval 5 (40000 steps performed)
10000/10000 [=====]
done, took 272.181 seconds
Testing for 10 episodes ...
Episode 1: reward: 5.000, steps: 5
Episode 2: reward: 5.000, steps: 5
Episode 3: reward: 4.000, steps: 5
Episode 4: reward: 5.000, steps: 5
Episode 5: reward: 4.000, steps: 5
Episode 6: reward: 4.000, steps: 5
Episode 7: reward: 4.000, steps: 5
Episode 8: reward: 5.000, steps: 5
Episode 9: reward: 4.000, steps: 5
Episode 10: reward: 5.000, steps: 5
```

FIGURA 8. RESULTADOS DE LA SIMULACIÓN DE 10 ESTUDIANTES.

FUENTE: ELABORACIÓN PROPIA.

	Nivel anterior	Objetos de aprendizaje entregados	Nivel actual
Memoria	bajo	Grafico bajo, Video med y Caso de Estudio alto	bajo
Concentración	bajo	Grafico bajo, Video med y Caso de Estudio alto	medio
Percepción	medio	Grafico bajo, Video med y Caso de Estudio alto	medio
Razonamiento Lógico	bajo	Grafico bajo, Video med y Caso de Estudio alto	bajo
	Nivel anterior	Objetos de aprendizaje entregados	Nivel actual
Memoria	bajo	Video bajo, Texto med y Grafico alto	bajo
Concentración	medio	Video bajo, Texto med y Grafico alto	medio
Percepción	medio	Video bajo, Texto med y Grafico alto	medio
Razonamiento Lógico	bajo	Video bajo, Texto med y Grafico alto	medio
	Nivel anterior	Objetos de aprendizaje entregados	Nivel actual
Memoria	bajo	Simulación baja, Caso de Estudio med y Texto alto	medio
Concentración	medio	Simulación baja, Caso de Estudio med y Texto alto	medio
Percepción	medio	Simulación baja, Caso de Estudio med y Texto alto	medio
Razonamiento Lógico	medio	Simulación baja, Caso de Estudio med y Texto alto	medio
	Nivel anterior	Objetos de aprendizaje entregados	Nivel actual
Memoria	medio	Grafico med, Video alto y Caso de Estudio med	medio
Concentración	medio	Grafico med, Video alto y Caso de Estudio med	alto
Percepción	medio	Grafico med, Video alto y Caso de Estudio med	medio
Razonamiento Lógico	medio	Grafico med, Video alto y Caso de Estudio med	medio
	Nivel anterior	Objetos de aprendizaje entregados	Nivel actual
Memoria	medio	Video med, Simulación alta y Texto med	medio
Concentración	alto	Video med, Simulación alta y Texto med	alto
Percepción	medio	Video med, Simulación alta y Texto med	alto
Razonamiento Lógico	medio	Video med, Simulación alta y Texto med	medio
Episode 2: reward: 5.000, steps: 5			

FIGURA 9. RESULTADOS DE LA SIMULACIÓN.

FUENTE: ELABORACIÓN PROPIA.

CONCLUSIONES

Con la implementación del modelo de aprendizaje adaptativo propuesto en un ambiente simulado, que emplea la asignación dinámica de objetos de aprendizaje contenidos en paquetes de aprendizaje para el desarrollo de las habilidades cognitivas útiles para el procesamiento de información, se puede verificar la eficiencia del modelo antes de llevarlo a un ambiente productivo.

La conformación de los paquetes de aprendizaje simplifica el número de acciones que el agente puede tomar (12), ya que en el modelo de Balasubramanian y Magret propone una acción por cada objeto de aprendizaje (7) y por cada habilidad cognitiva (4), lo que representa para el agente decidir entre 2401 acciones. Esto reduce considerablemente la ocupación de los recursos de cómputo destinados al sistema. Otra ventaja que tiene este modelo es el desarrollo de las capacidades del alumno dentro de un proceso cognitivo en un dominio determinado del conocimiento.

No obstante, la simplificación del modelo también induce una desventaja, y esta es que se deben diseñar de una mayor cantidad de recursos educativos digitales adaptativos REDA (Velandia, 2015), uno por cada regla propuesta y por cada módulo de curso. Para un curso de cinco módulos con las doce reglas establecidas se requeriría un total de 60 REDA. Por otra parte, el hecho de utilizar una plataforma de código abierto como Python tanto para el desarrollo del LMS como de las herramientas de inteligencia artificial asegura la compatibilidad entre sus módulos, la integración transparente de sus elementos y la escalabilidad hacia versiones más eficientes y de mayor alcance.

Una alternativa metodológica que también puede ser empleada es que en lugar de realizar la encuesta de opinión y el análisis ANOVA en el Módulo del Aprendizaje apliquen pruebas tipo test para cada habilidad cognitiva, donde estén embebidos en cada prueba los objetos de aprendizaje, con el fin de ponderar de forma evaluativa el nivel de habilidad detectado por cada objeto de aprendizaje definido. Esta alternativa metodológica no se emplea en la presente investigación, sin embargo, en la Figura 10 se puede observar una captura de pantalla del sistema LMS que el presente autor desarrolló para el Ayuntamiento de Huixquilucan, denominado “Huixquieduca” con los valores evaluativos de un curso para cada OA por cada HC.

Tarea/Nombre del test	Calificación Total	Puntaje Texto	Puntaje Audio	Puntaje Video	Puntaje Infografía
HC3 - Percepción	75.0	58.33	100.0	100.0	52.38
HC1 - Memoria	93.75	80.0	100.0	100.0	100.0
HC2 - Concentración	100.0	100.0	100.0	100.0	100.0
HC4 - Razonamiento Lógico	100.0	100.0	100.0	100.0	100.0

FIGURA 10. CAPTURA DE PANTALLA DEL SISTEMA HUIXQUIEDUCA.
FUENTE: ELABORACIÓN PROPIA.

Los posible usos y aplicaciones del modelo propuesto son en los procesos de inducción del personal de recién ingreso en diversos tipos de organizaciones, en procesos de capacitación en organizaciones en general, para coadyuvar en el desarrollo humano y en el desarrollo organizacional, y en instituciones educativas como herramienta de apoyo en el proceso de enseñanza aprendizaje.

En lo que concierne a futuras investigaciones, se puede ahondar en lo referente a los estilos de aprendizaje mixtos adaptables. Si bien ya se han desarrollado estudios anteriores en los que se habla de un estilo mixto de aprendizaje basados en ya sea en el modelo de Felder y Silverman (Velandia, 2015, Qodad *et al.*, 2016, Duque, Ovalle y Carrillo, 2020, Tabares, Duque y Fabregat, 2020) o en el modelo VARK (Duque, Ovalle y Carrillo, 2020, Tabares, Duque y Fabregat, 2020), estos no abordan un enfoque de estilos de aprendizaje adaptables, de acuerdo con los objetos de aprendizaje disponibles en una organización, tal y como lo sugiere el trabajo de Balasubramanian y Magret (2018).

De aquí puede desprenderse una investigación que dé un enfoque cuantitativo de los estilos de aprendizaje mixtos con el apoyo de la lógica difusa al clasificar a los individuos mediante las funciones de membresía. Finalmente, otra investigación posible es verificar si el estilo de aprendizaje o la composición de estilos de aprendizaje varía con el tiempo, es decir, si un individuo cambia su comportamiento frente a los objetos de aprendizaje en momentos diferentes.

REFERENCIAS

- Alzantot, M. (2017, 8 de julio). Deep Reinforcement Learning Demystified (Episode 2) — Policy Iteration, Value Iteration and Q-learning. *Medium*. Recuperado de: <https://medium.com/@m.alzantot/deep-reinforcement-learning-demystified-episode-2-policy-iteration-value-iteration-and-q-978f9e89ddaa#:~:text=The%20value%20function%20represent%20how,agent%20picks%20actions%20to%20perform.> [24 de mayo de 2021]
- Balasubramanian, V. y Margret, S. (2018). Learning style detection based on cognitive skills to support adaptive learning environment – A reinforcement approach. *Ain Shams Engineering Journal*, 9, pp. 895-907. <https://doi.org/10.1016/j.asej.2016.04.012>
- Capterra (2021) The 2021 Capterra Shortlist. *LMS Software*. <https://www.capterra.com/learning-management-system-software/#shortlist>
- Carrillo, S., Tigre, F., Túbón, E. y Sánchez, D. (2019, enero 31). Objetos Virtuales de Aprendizaje como estrategia didáctica de enseñanza aprendizaje en la educación superior tecnológica. *Revista Científica Mundo de la Investigación y el Conocimiento*, 3(1), pp. 287-304. Recuperado de: <https://dialnet.unirioja.es/servlet/articulo?codigo=7065194>
- Duque, N., Ovalle, D. y Carrillo A. (2020, enero-abril). Sistema basado en reglas para la generación personalizada de curso virtual. *TecnoLógicas*, 23(47). <https://doi.org/10.22430/22565337.1494>
- García, L. (2017). Educación a distancia y virtual: calidad, disrupción, aprendizajes adaptativo y móvil. *RIED, Revista Iberoamericana de Educación a Distancia*, 20(2), pp. 9-25 Recuperado de: <http://www.redalyc.org/articulo.oa?id=331453132001>

- Mercado, D., Pedraza, L. y Martínez, E. (2015, julio-diciembre). Comparación de Redes Neuronales aplicadas a la predicción de Series de Tiempo. *PROSPECTIVA*, 13(2), 88-95. <https://www.redalyc.org/articulo.oa?id=496250642011>
- Mota, C., Juárez, C. y Olguín, J. (2018) Clasificación de manzanas utilizando visión artificial y redes neuronales artificiales. *Ingeniería y Región*. 20 (2018). <https://doi.org/10.25054/22161325.1917>
- Qodad, A., Seghroucheni, Y., Al Achhab, M., El Yadari, M., El Kenz, A. y Benyoussef, A. (2016). An Adaptive Learning System based on a Job Model, the Differentiated Instruction and Felder and Silverman's Learning Styles Model. *2016 4th IEEE International Colloquium on Information Science and Technology (CiSt)*. <https://doi.org/10.1109/CiSt.2016.7805100>
- Rothman, D. (2018). Artificial Intelligence by example. *Birmingham: Packt Publishing*.
- Shawky, D. y Badawi, A. (2018). A Reinforcement Learning-Based Adaptive Learning System, en A. E. Hassanien *et al.* (Eds.). International Conference on Advanced Machine Learning Technologies and Applications, pp. 221-231. https://doi.org/10.1007/978-3-319-74690-6_22
- Sokol, V., Bronin, S., Karnaukh, V. y Bilov, M. (2020, enero 17). Developing Adaptive Learning Management Application for Project Team in IT-Industry. *Bulletin of National Technical University "KHP"*. Series: System Analysis, Control and Information Technologies. 1(3), 97-105. <https://doi.org/10.20998/2079-0023.2020.01.17>
- Sutton, R. y Barto, A. (2017). Reinforcement Learning: An Introduction. *The MIT Press*. Boston: MA
- Tabares, V., Duque, N. y Fabregat, R. (2020, enero-abril). Raim: framework para la inclusión adaptativa en entornos educativos para todos. *TecnoLógicas*, 23(47), pp. 179-196. <https://doi.org/10.22430/22565337.1495>
- Vallejo, M. (2002). El diseño de investigación: una breve revisión metodológica. *Medigraphic*, 72(1), pp. 8-12. Recuperado de: <https://www.medigraphic.com/pdfs/archi/ac-2002/aco21b.pdf>
- Velandia, J. (2015). Mejoras en habilidades cognitivas con el apoyo de un recurso educativo digital adaptativo. *Tesis de Maestría, Universidad de la Sabana: Colombia*. Recuperado de: <https://intellectum.unisabana.edu.co/handle/10818/22992>
- Xie, H., Chu, H., Hwang, G. y Wang, Ch. (2019). Trends and development in technology-enhanced adaptive/ personalized learning: A systematic review of journal publications from 2007 to 2017. *Computers & Education*, 140. <https://doi.org/10.1016/j.compedu.2019.103599>