



Casa abierta al tiempo

UNIVERSIDAD AUTÓNOMA METROPOLITANA
Unidad Iztapalapa

JDIE. 2025.1.10.3

Iztapalapa, CDMX, a 23 de octubre de 2025.

DR. ROMÁN LINARES ROMERO
PRESIDENTE DEL CONSEJO DIVISIONAL
DIVISIÓN DE CIENCIAS BÁSICAS E INGENIERÍA
P R E S E N T E

Estimado Dr. Linares Romero, le solicito, atentamente, someta al pleno del Consejo que usted acertadamente preside, el **informe de actividades del periodo sabático** de 14 meses del **Profr. JOEL RICARDO JIMÉNEZ CRUZ** para su conocimiento. Las fechas de inicio y terminación de dicho periodo son 8 de julio de 2024 y 7 de septiembre de 2025, respectivamente.

A la presente se anexan:

Formato del informe.

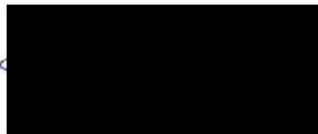
Informe elaborado por el profesor.

Documentos probatorios.

Agradezco la atención otorgada a la presente y quedo a sus órdenes.

Atentamente

"Casa abierta al tiempo"



M. en C. Omar Lucio Cabrera Jiménez
Jefe del Departamento de Ingeniería Eléctrica



Departamento de Ingeniería Eléctrica

Departamento de Ingeniería Eléctrica
División de Ciencias Básicas e Ingeniería
Av. Ferrocarril San Rafael Atlixco 186,
Col. Leyes de Reforma 1ª Sección, Iztapalapa,
Cd. de México. C.P. 09310, Oficina T-163



55 5804 4628



ie@xanum.uam.mx

DATOS GENERALES

Nombre del profesor: Joel Ricardo Jiménez Cruz N° empleado: 18815
 Departamento: Ingeniería Eléctrica Área: Ingeniería Biomédica
 Teléfono particular: [REDACTED] Extensión UAM-I: [REDACTED] E-mail: [REDACTED]@xarum.uam.mx

DATOS DEL PERÍODO SABÁTICO SOLICITADO

N° meses solicitados: 14 Fecha de inicio: 08-07-24 Fecha de terminación: 07-09-25
 Institución donde se realizará: UAM- IZTAPALAPA
 Depto., Laboratorio, etc.: Laboratorio de Cibernética
 Domicilio de la institución: Av. San Rafael Atlixco 186, Leves de Reforma, 09340
 Teléfono: [REDACTED] Fax: [REDACTED] E-mail: [REDACTED]@xarum.uam.mx

OBJETIVOS DEL PERÍODO SABÁTICO

Implementación y notas para los cursos de
- Introducción a la programación
- Introducción a la robótica

METAS ALCANZADAS EN EL PERÍODO SABÁTICO

☒ Memorias in extenso
en libro de resúmenes*

☐ Artículos de investigación en
revista indexada*

☒ Presentaciones en congresos

☒ Libros o capítulos de libros

☐ Grado

☐ % Avance de estudios de
posgrado

☐ Otros (especifique): _____

* Indicar en anexo si se trata de trabajo publicado, aceptado o sometido

TIPO DE ACTIVIDADES ACADÉMICAS DESARROLLADAS

(Indique aquellas relacionadas con las actividades desarrolladas)



Investigación



Docencia



Difusión



Formación académica



Formación profesional



Entrenamiento técnico



Otros (especifique):

RESUMEN DEL PLAN DE ACTIVIDADES ACADÉMICAS DESARROLLADAS

(El llenado de esta sección no sustituye el informe detallado de actividades)

- Dos publicaciones en memorias de congresos
- Elaboración del libro:
Modelos Computacionales de Agentes Robóticos
Biomiméticos

PARA USO DEL JEFE DE DEPARTAMENTO

Después de haber evaluado el informe detallado de actividades del período sabático del interesado según los lineamientos establecidos para tal efecto; informo al Consejo Divisional que:



Los objetivos SE cumplieron satisfactoriamente

Los objetivos SE cumplieron parcialmente

Los objetivos NO se cumplieron

NO se cumplió el propósito del sabático


Firma del Jefe de Departamento

23 oct 2025

Fecha

PARA USO DEL CONSEJO DIVISIONAL

El Consejo Divisional, en su Sesión No. _____ del _____ sobre el Período sabático del interesado acordó que:



Los objetivos SE cumplieron satisfactoriamente

Los objetivos SE cumplieron parcialmente

Los objetivos NO se cumplieron

NO se cumplió el propósito del sabático

Secretario del Consejo Divisional

*Además de este formato-resumen, el interesado deberá entregar su Informe detallado de actividades junto con la documentación probatoria correspondiente.

Informe de Año Sabático

Dr. Román Linares Romero

Presidente del Consejo Divisional de Ciencias Básicas e Ingeniería

Presente:

Por medio del presente, me permito entregar el **reporte de actividades correspondientes al año sabático**, el cual dio inicio el **8 de julio de 2024** y concluyó el **7 de septiembre de 2025**.

Durante este periodo se realizaron **dos publicaciones académicas** y se avanzó de manera significativa en la **elaboración de un libro**, actualmente en fase final de revisión y edición.

Publicaciones

Artículos

Jiménez, J. (2025). *Explorando el aprendizaje por descubrimiento en la programación del robot mBot en la educación primaria.* *Pistas Educativas*, 46(149), enero-junio 2025. México: Tecnológico Nacional de México en Celaya. Disponible en línea.
<https://pistaseducativas.celaya.tecnm.mx/index.php/pistas/article/view/3786/2677>

Este artículo presenta una experiencia educativa basada en el **aprendizaje por descubrimiento**, aplicada al uso del **robot mBot** y del lenguaje visual **mBlock** en estudiantes de educación primaria. El enfoque promueve que los niños aprendan programación y robótica mediante la experimentación y la resolución de problemas, estimulando la curiosidad, la creatividad y la autonomía. A lo largo del proyecto, los participantes diseñaron y programaron comportamientos del robot —como movimientos y respuestas a estímulos— integrando nociones de matemáticas, física y pensamiento computacional. Los resultados muestran un incremento significativo en la motivación, la comprensión conceptual y el trabajo colaborativo. El estudio concluye que el aprendizaje por descubrimiento, mediado por herramientas tecnológicas, potencia la formación integral y el desarrollo del pensamiento lógico y computacional desde edades tempranas, consolidando una pedagogía activa y significativa.

Jiménez, J. (2024). *La sinergia entre el microaprendizaje, los chatbots y el aprendizaje autodirigido.* En XXXVII Simposio Internacional SOMECE 2024 y Jornadas Internacionales de Sistemas y Ambientes Educativos (JISAE 2024). México. Disponible en línea.

El artículo propone el diseño e implementación de un **chatbot educativo** orientado a fortalecer el **aprendizaje autorregulado y ubicuo** mediante técnicas de inteligencia artificial. El sistema, basado en **procesamiento del lenguaje natural (PLN)**, permite la interacción personalizada

con el estudiante, acompañándolo en las etapas de planificación, monitoreo y autoevaluación. El chatbot facilita el acceso a contenidos y tutoría adaptativa en cualquier momento y lugar, promoviendo la autonomía y la gestión individual del conocimiento. Entre sus beneficios destacan la **retroalimentación inmediata**, la **personalización del ritmo de aprendizaje** y la **detección temprana de dificultades**. El trabajo plantea un modelo de educación apoyado por tecnología que impulsa la autorregulación, la motivación y la flexibilidad del aprendizaje, alineado con las tendencias contemporáneas de **educación mediada por inteligencia artificial generativa** y centrada en el estudiante.

Libro

Modelos computacionales de agentes robóticos bioinspirados: del movimiento aleatorio a la conducta adaptativa

El libro *Agentes Robóticos Bioinspirados: Modelos Computacionales del Comportamiento Animal* presenta una propuesta integral para el estudio y la simulación de la **inteligencia natural** mediante la robótica y la programación en **Python**. Su propósito es mostrar la evolución progresiva de la conducta inteligente, desde las formas más simples de movimiento reactivo hasta los modelos de aprendizaje y decisión adaptativos inspirados en sistemas biológicos.

Cada capítulo combina teoría, simulaciones y ejercicios diseñados para fomentar el **pensamiento computacional**, la comprensión de **sistemas dinámicos** y el **análisis del comportamiento emergente**. La obra está organizada en módulos auto-contenidos que permiten al lector reproducir modelos biológicos —como estrategias de forrajeo, quimiotaxis, navegación de insectos y aprendizaje por refuerzo— y traducirlos en algoritmos aplicables tanto a la **robótica simulada** como a sistemas físicos basados en **Arduino, ESP32 o Raspberry Pi**.

El texto se complementa con tres apéndices:

- **Apéndice A:** Herramientas matemáticas y computacionales para comprender los modelos.
- **Apéndice B:** Extensiones y proyectos avanzados que amplían los experimentos de cada capítulo.
- **Apéndice C:** Código completo en Python de las simulaciones descritas.

Desde un enfoque interdisciplinario, la obra integra elementos de **biología del comportamiento, neuroetología, ingeniería de control, ciencias cognitivas, robótica y programación**, constituyéndose en una guía para el diseño de agentes artificiales y físicos capaces de adaptarse, explorar y aprender en entornos complejos.

Actualmente, el libro se encuentra en la etapa final de revisión, unificación del estilo editorial y verificación de los apéndices y ejemplos prácticos.

Atentamente,



Dr. Joel Ricardo Jiménez Cruz
Profesor-Investigador
Departamento de Ingeniería Eléctrica
División de Ciencias Básicas e Ingeniería

EXPLORANDO EL APRENDIZAJE POR DESCUBRIMIENTO EN LA PROGRAMACIÓN DEL ROBOT MBOT EN LA EDUCACIÓN PRIMARIA

EXPLORING DISCOVERY LEARNING IN MBOT ROBOT PROGRAMMING IN PRIMARY EDUCATION

Joel Ricardo Jiménez Cruz

Universidad Autónoma Metropolitana, Unidad Iztapalapa, México
jcjr@xanum.uam.mx

Recepción: 8/diciembre/2024

Aceptación: 24/marzo/2025

Resumen

Este estudio resalta la importancia de un diseño educativo, que se basa en el aprendizaje basado en el descubrimiento y que se aplica a la enseñanza de la robótica utilizando el robot mBot y a la programación empleando el lenguaje de programación visual mBlock. La implementación de este diseño brinda a los estudiantes de educación primaria, prácticas que integran y brindan una visión de la programación y la robótica en las áreas de STEAM.

En un salón con 24 estudiantes de primaria entre 8 y 13 años se realizaron prácticas de introducción a la robótica y a la programación visual basándose en la metodología activa del aprendizaje por descubrimiento. Se formaron seis equipos de trabajo con cuatro integrantes. Se observaron en los estudiantes; una mejor comprensión de conceptos robóticos y computacionales, un aumento en la motivación y una maduración en el aprendizaje de habilidades matemáticas y geométricas.

Palabras Clave: Aprendizaje por descubrimiento, Educación primaria, Programación, Robótica, STEAM.

Abstract

This study highlights the importance of an educational design based on discovery-based learning. This design is applied to teaching robotics using the mBot robot and programming using the mBlock visual programming language. Implementing this

design provides elementary school students with practices that integrate and provide insight into programming and robotics in STEAM areas.

In a classroom with 24 primary school students between 8 and 13 years old, introductory practices were carried out on robotics and visual programming based on the active methodology of discovery learning. Six work teams with four members were formed. A better understanding of robotic and computational concepts, an increase in motivation, and a maturation in the learning of mathematical and geometric skills were observed in the students.

Keywords: *Discovery Learning, Elementary Education, Programming, Robotics, STEAM.*

1. Introducción

La programación y la robótica son disciplinas que se utilizan para apoyar la práctica pedagógica en la educación informal o formal, especialmente en las materias de STEAM (Ciencia, Tecnología, Ingeniería, Arte y Matemáticas). Estas disciplinas promueven la adquisición de conocimientos prácticos sobre tecnología, y fomentan el pensamiento crítico, lógico, algorítmico, computacional, la creatividad, el trabajo en equipo y la resolución de problemas [Pérez, 2024], [Caro, 2023].

La robótica es una ciencia que aglutina varias disciplinas o ramas de la tecnología con el objetivo de diseñar máquinas programadas para realizar tareas de forma automática o autónoma, o para simular el comportamiento humano o animal. La robótica se dedica a la construcción de artefactos que intentan materializar el deseo humano de crear seres a su semejanza para responder a algunas de sus necesidades más complejas y también para liberarse de trabajos tediosos o peligrosos. Su cometido es desarrollar diferentes aspectos de un robot funcional: su autonomía e inteligencia propia, su resistencia y capacidad de operatividad, su programación, sus mecanismos de control y su autoconciencia [Cáceres, 2023].

Por otro lado, la programación es un proceso creativo y lógico que consiste en escribir una serie de instrucciones que una máquina o procesador pueda comprender y ejecutar. A través de la programación se puede optimizar y controlar el comportamiento de un robot.

Para la enseñanza de estas materias se pueden aplicar enfoques didácticos como el aprendizaje basado en proyectos o en problemas, el aprendizaje por descubrimiento, el aprendizaje basado en problemas y las estrategias de aprendizaje colaborativo para permitir una mayor autonomía y experiencias de aprendizaje activo y que enfatizan metodologías centradas en el estudiante [Restrepo, 2005].

A partir de estas estrategias activas de enseñanza-aprendizaje centradas en el estudiante, se puede abordar el pensamiento computacional a partir del análisis lógico de datos, las abstracciones y la resolución de problemas. Todas estas prácticas en un contexto educativo permiten la colaboración e interacciones de los estudiantes y profesores para desarrollar habilidades y resolver problemas cada vez más complejos [Rosero, 2024].

En particular, en este escrito, se analiza el potencial educativo del aprendizaje por descubrimiento como un tipo de aprendizaje activo, práctico y experiencial y como un método fundamental para fomentar una comprensión y un compromiso de aprendizaje más profundo en los estudiantes, en la enseñanza de la programación y la robótica en la educación primaria.

2. Métodos

El aprendizaje por descubrimiento y sus beneficios

El modelo pedagógico del aprendizaje por descubrimiento se inspira y se complementa de diversas teorías del aprendizaje como la teoría del aprendizaje significativo de David Ausubel (1918-2008) que menciona que el aprendizaje es significativo cuando se integran los saberes previos y los nuevos. Jean Piaget (1896-1980) acentúa que el conocimiento surge de la experiencia que, a su vez, provoca nuevos esquemas mentales. Jerome Bruner (1915-2016) y el construccionismo de Seymour Papert con su tortuga logo [Papert, 1980], destacan la importancia esencial de las interacciones sociales en los entornos de aprendizaje desde la perspectiva de las teorías socioculturales. El constructivismo de Lev Vygotsky (1896-1934) subraya la dimensión sociocultural y su incidencia en el proceso de aprendizaje [Carranza, 2024].

Bruner al igual que John Dewey sostiene que el descubrimiento por uno mismo enseña a adquirir información enfocada en la resolución de problemas, es decir; se aprende, haciendo. El aprendizaje según [Papert, 1980] ocurre cuando los alumnos participan en la creación de artefactos significativos que pueden ser investigados y compartidos. La teoría construccionista sostiene que cuando los estudiantes trabajan con materiales que les permiten diseñar y construir artefactos significativos, aprenden mejor. El conocimiento se construye a través del hacer con el uso de herramientas como robots y computadoras [Rosero, 2024].

El aprendizaje por descubrimiento es una técnica pedagógica basada en la averiguación, indagación o exploración y se considera un enfoque educativo constructivista y construccionista que comprende un marco teórico que fundamenta la actividad pedagógica y que concibe el aprendizaje como una elaboración en la que se involucran activamente el profesor y los estudiante; el profesor actúa como facilitador que guía la interconexión entre los saberes previos del estudiante y las nuevas construcciones mentales, y el estudiante interviene activamente en la confección de su propia realidad, conduciendo a la creación de entornos de aprendizaje colaborativos [Gómez, 2018].

El aprendizaje por descubrimiento conlleva muchos de los conceptos del constructivismo en que el estudiante procesa activamente el significado de su aprendizaje, avanza a su propio ritmo sin instrucción deliberada, ni un plan de estudios estrictamente prescrito. Está relacionado con el aprendizaje basado en problemas o en proyectos y el aprendizaje experiencial.

En el aprendizaje por descubrimiento, los estudiantes exploran, construyen y experimentan activamente su conocimiento a través de una comprensión personal de los conceptos. Este enfoque contrasta con los métodos tradicionales en los que la información se transmite principalmente a través de conferencias, pláticas y libros de texto [Espinoza, 2022].

En el aprendizaje por descubrimiento, el docente sirve como mediador o guía que proporciona a los estudiantes, la meta de aprendizaje, apoyándolos con las herramientas necesarias para que el alumno de forma autónoma e independiente recorra el camino para alcanzar el objetivo formativo propuesto. En este sentido, el

contenido educativo no es dado por el educador de forma directa al estudiante, sino que, debe ser descubierto y construido por éste en el desarrollo de las actividades de aprendizaje. Así, las actividades realizadas se convierten en oportunidades para aprender, alcanzar el conocimiento, desarrollar la curiosidad y las estrategias para aprender a aprender en distintos contextos. De esta manera, se mejoran la capacidad de búsqueda, la selección y análisis de la información, la interpretación y lectura de contextos, habilidades propositivas y resolución de problemas, entre otras.

En el aprendizaje por descubrimiento las actividades educativas se orientan a la resolución de problemas en contextos reales donde existen oportunidades de aprendizaje con el entorno. Este enfoque se basa en la indagación para garantizar que el aprendizaje se produzca cuando el sujeto descubre activamente y resuelve problemas. Según estos fundamentos, aprender haciendo implica un enfoque activo de enseñanza-aprendizaje donde el estudiante busca adquirir conocimientos y habilidades a través de la exploración o el descubrimiento.

Resumiendo, el aprendizaje por descubrimiento se caracteriza porque su énfasis está en [Ozdem, 2020]:

- El proceso de construcción del conocimiento y de las habilidades obtenidas.
- El estudiante que asume un papel autónomo y activo en su proceso formativo y el maestro que se convierte en facilitador, motivador y diseñador de oportunidades de aprendizaje.
- Actividades didácticas que proponen tareas factibles y atractivas para los estudiantes.
- Acercar los contenidos curriculares al mundo real donde la meta de formación no radica tanto en los conceptos, sino en las habilidades y competencias involucradas para alcanzar la comprensión de esa concepción y hacer uso de ella en la resolución de problemas o propuestas de innovación.
- Promover una cultura en el aula donde exista un sentido de propósito compartido entre docentes y estudiantes, donde se fomenta una mentalidad abierta al diálogo y donde se anima a los estudiantes a hacer preguntas, esclarecer confusiones y evitar frustraciones.

- Proponer una estrategia de aprendizaje que organiza a los alumnos para trabajar en equipo y despierta el interés con un tema motivador propiciando el trabajo colaborativo.
- Coadyuvar para que los estudiantes establezcan conocimientos fundamentales a través de ejemplos, prácticas y retroalimentación.
- Integrar información adicional y desarrollar habilidades de resolución de problemas y de pensamiento crítico.

Por otro lado, el papel de los docentes en el aprendizaje por descubrimiento es fundamental para el éxito de los resultados del aprendizaje, considerando que:

- Se proporcionen tareas guiadas aprovechando una variedad de técnicas de instrucción.
- Los estudiantes expliquen sus ideas a compañeros y profesores para que se proporcione las retroalimentaciones pertinentes.
- Los maestros proporcionen ejemplos de cómo completar las tareas.
- La orientación docente se encuentra en aprovechar el razonamiento de los estudiantes y la interconexión con sus conocimientos y experiencias.

El aprendizaje por descubrimiento puede ocurrir cuando al estudiante no se le proporciona una respuesta exacta ni completa, sino ideas iniciales y materiales para encontrar la respuesta por sí mismo. Tiene lugar en situaciones de resolución de problemas en las que los alumnos interactúan con su entorno explorando y manipulando objetos, lidiando con preguntas y controversias o realizando experimentos, mientras aprovechan su propia experiencia y conocimientos previos. Es decir, se caracteriza por tener una orientación y explicaciones mínimas por parte del docente, una resolución de problemas con múltiples soluciones, el uso de materiales prácticos, y una mínima repetición y memorización.

El aprendizaje basado en el descubrimiento aplicado en la enseñanza de la programación y la robótica promueve el desarrollo de una mente algorítmica y consecuentemente, la elaboración del pensamiento computacional.

Pensamiento computacional

La importancia de enseñar habilidades de pensamiento computacional desde una edad temprana es un elemento clave para apoyar las tareas cognitivas tecnológicas actuales. Estas prácticas pueden ser una gran ventaja cuando se integran en actividades pedagógicas para mejorar las habilidades en el razonamiento lógico, las matemáticas, la resolución de problemas y el pensamiento crítico.

El pensamiento computacional es un proceso de pensamiento implicado en la formulación de problemas y sus soluciones que puedan llevarse a cabo por un procesador de información ya sea humano o maquinal. El pensamiento computacional involucra la capacidad de entender y solucionar problemas que se pueden automatizar. También involucra una serie de mecanismos del pensamiento, como el abstracto-matemático y el pragmático-ingenieril.

Abarca conceptos, abstracciones y prácticas con entes computacionales tomando en consideración tanto el hardware como el software y considerando los siguientes principios (Figura 1):

- Descomposición sistémica de un problema, proyecto o proceso grande o complejo en unidades más pequeñas.
- Generalización en busca de estándares, llevando a cabo un reconocimiento de patrones repetitivos. Establecimiento de nuevas conexiones y pensamientos sobre otras aplicaciones o usos en otros contextos.
- Abstracción y elección de buenas representaciones para crear "modelos o simulaciones" del problema a resolver, realizando tareas de predicción.
- Elaboración de algoritmos para la resolución de problemas.
- Evaluación de situaciones futuras para mejorar el diseño asegurando predicciones correctas que sean adecuada para sus propósitos, asentando juicios sobre qué hacer, en función de una serie de factores que son parte de un contexto.

Como complemento a los principios o capacidades que configuran el pensamiento computacional, se encuentran una serie de técnicas como la reflexión (juicios argumentados), el análisis (descomposición, abstracción, generalización), el diseño

(satisfacción de necesidades o deseos y adaptación a un contexto), la programación (traducción a un lenguaje computacional y depuración) y la aplicación (satisfacción de necesidades y generalización) [Bordignon, 2020].



Fuente: [Bordignon, 2020]

Figura 1 Principios y técnicas del pensamiento computacional.

Desarrollo del pensamiento computacional con el robot mBot

La programación de robots motiva a los estudiantes a ser curiosos y creativos. Además de ser emocionante, los robots hacen posible que los estudiantes creen programas y los prueben inmediatamente, lo que genera experiencias de aprendizaje colaborativas, creativas y auténticas.

En etapas tempranas o para comenzar a programar se recomienda el uso inicial de lenguajes visuales en los cuales, los programas se van formando con la manipulación de piezas como un rompecabezas que disminuyen los errores de tipo sintáctico de los lenguajes textuales. Codificar con estos lenguajes es más fácil que usar lenguajes de programación tradicionales porque los estudiantes pueden jugar e interactuar con bloques de colores al crear los guiones o scripts de los programas. Los lenguajes de programación visual en bloques (Tickle, Blocky, Scratch, mBlock...), permiten programar y manipular los robots y pueden facilitar la experimentación por parte de los jóvenes debido al carácter intuitivo de este tipo de programación [Cuenca, 2023].

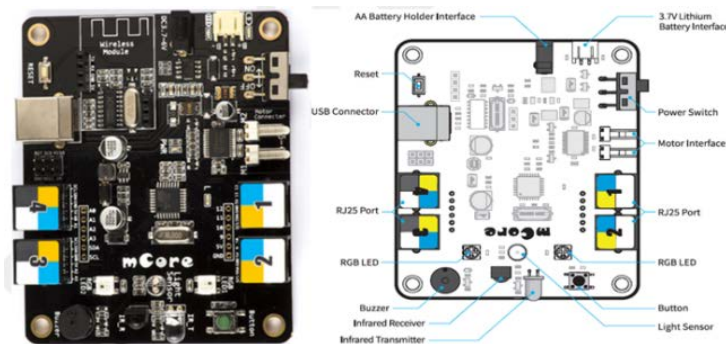
Por otro lado, existen varios robots que se pueden utilizar en la enseñanza como Ozobot, mBot, Lego EV3, Sphero y Edison (Figura 2).



Fuente: [Voštinár, 2024]

Figura 2 Algunos robots que se pueden utilizar en la enseñanza.

De ellos, mBot es un robot relativamente económico y muy atractivo (<https://support.makeblock.com/hc/en-us/sections/360001828973-mBot>). Tiene la posibilidad de trabajar con el lenguaje visual mBlock y otros lenguajes como el de Arduino o Python, utilizando computadoras o dispositivos móviles (iOS y Android). Resulta fácil experimentar con él, y es apto para su uso por personas que atienden por primera vez cursos introductorios de robótica, programación y electrónica basada en el microcontrolador Arduino UNO (Figura 3).



Fuente: [<https://support.makeblock.com/hc/en-us/articles/4412894402967-mCore-Main-Control-Board-of-mBot>]

Figura 3 Detalles de la placa Arduino de mBot.

Una ventaja de mBot es su buena relación costo/beneficio con respecto a otros robots, lo cual lo hace asequible, fácil de usar y ofrece interesantes oportunidades de aprendizaje y entretenimiento. mBot se puede conectar a una computadora u otros dispositivos a través de USB, Bluetooth o Wi-Fi. Esto permite a los programadores transmitir sus programas desde una computadora o teléfono inteligente al robot y controlar su movimiento y actuación.

Algunos recursos del robot mBot comprenden:

- Motores y ruedas: mBot tiene dos ruedas que son impulsadas por motores. A través de ellos se mueve hacia adelante y hacia atrás y hace giros a la izquierda y a la derecha.
- Luces LED: están ubicadas en la parte superior del mBot y se pueden programar para brillar en diferentes colores, lo que permite al robot transmitir señales visuales.
- Buzzer: El robot se puede programar para que emita pitidos o melodías en varios eventos como parte de los proyectos.
- Sensor de distancia ultrasónico: Está ubicado en la parte frontal del mBot y puede medir la distancia a objetos que van desde unos pocos centímetros hasta unos pocos metros. Envía ondas ultrasónicas a los objetos y mide el tiempo que tarda la señal en regresar al sensor después de reflejarse en el objeto. Con la información recibida, mBot calcula la distancia al objeto y puede reaccionar.
- Sensor de luz: el sensor de luz del mBot puede medir la intensidad de la luz en el ambiente. Esto le permite reconocer diferentes niveles de iluminación (el rango es 0-1023) y reaccionar ante los cambios en las condiciones de iluminación.
- Seguidor de línea: este sensor está ubicado en la parte inferior del mBot y le permite seguir una línea recta o curva previamente dibujada de negro.
- Receptor y transmisor de infrarrojos: Permiten que mBot se comunique con otros robots mBot mediante la transmisión y recepción de señales, lo que facilita la cooperación entre robots o la creación de escenas y juegos interactivos.

Además de la configuración básica de mBot, hay disponibles una variedad de accesorios y módulos mBlock que se pueden agregar para ampliar la funcionalidad y realizar tareas más complejas.

En términos de programas de estudio, las prácticas y actividades que se pueden proponer en el aprendizaje inicial de la robótica consideran:

- Comprender y saber utilizar sistemas tecnológicos.
- Conocer y resolver sistemas y aplicaciones.
- Transferir conocimientos para el aprendizaje de nuevas tecnologías.
- Describir y analizar una secuencia de acciones, siguiendo instrucciones (algoritmos).
- Implementar soluciones a problemas utilizando un lenguaje de programación visual basado en bloques.
- Usar pasos básicos en la resolución de un problema algorítmico para diseñar soluciones.
- Aplicar la adquisición de conocimientos y habilidades a las áreas STEAM

3. Resultados

Actividades de programación con el robot mBot

En una clase con 24 estudiantes de primaria se utilizó el robot mBot como la plataforma para aprender robótica y programación utilizando como técnica pedagógica el aprendizaje por descubrimiento. Este robot educativo permitió que los estudiantes pudieran programar los movimientos del robot a través de una interfaz fácil y condescendiente. También permitió que los alumnos capturarán mejor los conceptos abstractos y desarrollaran habilidades cruciales como la resolución de problemas, el pensamiento crítico y creativo, y la colaboración con la guía del docente que brindó las herramientas y el contexto para que los estudiantes llegaran a sus propias conclusiones de programación y manejo de robots (Figura 4).



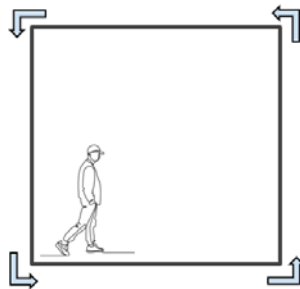
Fuente: elaboración propia

Figura 4 Actividades de programación utilizando el aprendizaje por descubrimiento.

Se formaron seis grupos de cuatro estudiantes para qué de manera colaborativa programaran al robot. Al trabajar en equipos, se observó que los estudiantes aprenden a dividir tareas, asignar responsabilidades y solucionar problemas en forma conjunta. Este esfuerzo colaborativo refleja escenarios de otras materias en su formación posterior y del mundo real, donde el trabajo en equipo es esencial para preparar a los estudiantes a desafíos futuros en entornos familiares, académicos o profesionales.

A continuación, se comentan las actividades y prácticas efectuadas con el robot mBot. Las actividades se realizaron con los principios del pensamiento computacional y se apoyaron en la didáctica del aprendizaje por descubrimiento para que los estudiantes fueran descubriendo por sí mismos la forma de programar el robot, considerando la descomposición de los programas, la abstracción para conceptualizar y simplificar, la generalización en las instrucciones, el desarrollo de los algoritmos y la evaluación para detectar fallas.

Antes de iniciar la codificación de las instrucciones en el lenguaje visual mBlock y dando una teoría mínima de los conceptos fundamentales como programación, computadoras, robots, algoritmos, se les pidió a los estudiantes que dieran instrucciones verbales al profesor, para que, actuando como robot, se moviera como una persona en marcha, que caminara en línea recta y girara un ángulo de 90° . Para ello, los estudiantes empezaron a dar diversos tipos de órdenes, por ejemplo, camina y gira, y el profesor no se movía y se realimentaba verbalmente, diciéndoles que no sabía cómo caminar. De ahí, iban surgiendo las ideas entre todos, de la precisión de los algoritmos de una manera natural (Figura 5).



Fuente: [<https://es.vecteezy.com/arte-vectorial/27764408-continuo-linea-vector-ilustracion-de-persona-en-sombrero-caminando>]

Figura 5 Caminar en línea recta y girar 90° y luego caminar describiendo un cuadrado.

Se procedió a dar la instrucción de dar un paso que consistía en alzar y mover una pierna hacia adelante y luego de alzar y mover hacia adelante la otra pierna. Luego dar varios pasos durante cierto tiempo y finalmente dar la instrucción de girar 90° que se podría actuar dando una pirueta o un giro brusco hacia la izquierda o hacia la derecha.

La siguiente indicación fue pedirles a los estudiantes que, teniendo estas instrucciones ya asimiladas, comunicaran al profesor, cómo debía procederse para trazar y caminar la forma de un cuadrado. Esta divertida introducción prepara el escenario para que los alumnos aprendan los comandos fundamentales en el entorno mBlock del robot que consisten en (Figura 6):

- Caminar: que se traduce en el comando "Avanzar durante un cierto tiempo".
- Girar 90° a la derecha/izquierda: que se codificarán con los comandos de mBot "Girar a la derecha durante cierto tiempo" o "Girar a la izquierda durante cierto tiempo", respectivamente.

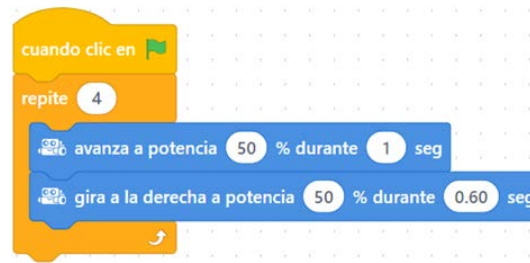


Fuente: elaboración propia

Figura 6 Instrucciones para avanzar y girar 90° y luego formar un cuadrado.

Una vez que a los estudiantes se les mostró, cuáles eran los comandos básicos dentro del entorno mblock para avanzar y girar, los estudiantes por medio del descubrimiento, la ayuda de sus compañeros y su intuición empiezan a generar el código para indicarle al robot que trace un cuadrado (Figura 6).

Una vez asimilada la forma en la que el robot describe un cuadrado repitiendo la secuencia 4 veces, guiados por el profesor, los estudiantes pueden realizar la misma acción utilizando un bucle (Figura 7).



Fuente: Elaboración propia

Figura 7 El uso de un ciclo para que el robot describa el cuadrado.

Estos comandos se ejecutaron en línea y en tiempo real, sin descargar el código en el robot. Las acciones del robot se realizan con el cable USB conectado al robot. Esta manera inicial es más útil y sencilla para que los niños observen lo que la programación produce en el robot. Esta tarea aparentemente simple los obliga a aplicar sus conocimientos de geometría, visualizar la forma del cuadrado, el número de lados y los ángulos rectos y codificarlos en una secuencia de comandos específicos para que el robot describa el cuadrado.

Posteriormente, y de la misma manera, se pide a los grupos de estudiantes que extiendan los conceptos aprendidos para que el robot describa un triángulo y luego un círculo.

Otras actividades que se les puede pedir a los estudiantes que programen son que [Sáez, 2019]:

- El robot se mueva hacia adelante, atrás, a la izquierda o a la derecha de acuerdo con la presión de las teclas.
- El robot evite obstáculos utilizando el sensor ultrasónico.

De acuerdo con los resultados observados, que están en concordancia con lo reportado en la literatura, se encuentra que el aprendizaje por descubrimiento es muy eficaz debido a:

- La retroalimentación que se proporciona, los ejemplos trabajados, el andamiaje y las explicaciones obtenidas por parte de compañeros y el profesor [Alfieri, 2011],
- La mejora en las habilidades de pensamiento crítico de los estudiantes [Hidayatul, 2024].

- La mejora en el desempeño de los estudiantes en la adquisición de conocimiento factual, conceptual y procedimental si se llegaran a utilizar videos de YouTube acompañados del aprendizaje por descubrimiento [Koto, 2020].

4. Discusión

Los conceptos de secuencias, bucles y condicionales en la programación permiten a los estudiantes desarrollar competencias relacionadas con las matemáticas, las ciencias y la tecnología, fomentando un enfoque interdisciplinario. Los estudiantes se interesan por la respuesta del sensor, mostrando motivación, interés y voluntad en el proceso. En la programación visual de mBot, los estudiantes en equipo se embarcan en un viaje para desarrollar un conjunto diverso de habilidades [Desislava, 2023], como las que se explican a continuación.

Aplicación de conceptos geométricos

La programación de los movimientos de formas geométricas como cuadrados, triángulos o círculos requiere que los estudiantes apliquen conceptos matemáticos (vértices, lados, ángulos, medidas y relaciones espaciales) en un contexto práctico pues se requieren calcular e ingresar instrucciones para que el robot avance una cierta distancia y gire un determinado ángulo.

Al aplicar los conceptos geométricos se mejora la visualización espacial ya que al programar el robot se requiere que los estudiantes visualicen mentalmente sus movimientos, giros y trayectorias, mejorando sus habilidades de orientación y representación espacial [Sáez, 2023].

La precisión es clave: la programación exige instrucciones y comandos precisos para que el robot funcione según lo previsto, lo que promueve la atención al detalle y la meticulosidad.

La descomposición de problemas

En la programación se aplica el principio de descomposición al dividir los problemas en pasos más pequeños y procesables, perfeccionando el pensamiento

lógico y las habilidades sistemáticas de resolución de problemas. Los estudiantes crean secuencias ordenadas de instrucciones para lograr un objetivo específico, sentando las bases para comprender algoritmos y estructuras de programación. Aprenden a traducir conceptos matemáticos abstractos en acciones tangibles y secuencias de comandos que el robot mBot puede ejecutar. Este proceso no sólo refuerza su comprensión de las matemáticas, sino que también mejora sus habilidades de pensamiento computacional.

El mejoramiento de habilidades computacionales

Al escribir código para controlar los movimientos y acciones del mBot, los estudiantes desarrollan el pensamiento algorítmico, estructurando una secuencia de pasos para lograr el resultado deseado. Este proceso fomenta el razonamiento lógico para la resolución de problemas, que es fundamental tanto en matemáticas, como en informática.

La programación del robot mBot presenta a los estudiantes los principios básicos del pensamiento computacional, incluida la descomposición de problemas, la secuenciación de instrucciones y la resolución algorítmica de problemas.

Los estudiantes profundizan en conceptos básicos de programación como condicionales y bucles para controlar el comportamiento del robot, sentando las bases para aprender lenguajes de programación más complejos.

La programación grupal fomenta un entorno colaborativo donde los estudiantes trabajan juntos, se comunican de manera efectiva y logran un objetivo común.

5. Conclusiones

El robot mBot sirve como una poderosa herramienta educativa en las escuelas primarias, facilitando el desarrollo de habilidades matemáticas, geométricas y computacionales a través del aprendizaje por descubrimiento y las actividades de programación visual grupal. Al participar en la exploración práctica y la resolución colaborativa de problemas con mBot, los estudiantes profundizan la comprensión de conceptos abstractos y cultivan habilidades esenciales que son vitales en un mundo impulsado por la tecnología actual.

En las prácticas realizadas se observó que el uso de la robótica y la programación apoyados por el aprendizaje por descubrimiento mejora la comprensión de matemáticas, geometría y pensamiento computacional, allanando el camino para comprender la tecnología digital en constante evolución y se fomenta la creatividad, potenciando la interconexión de las distintas disciplinas STEAM.

Algunos beneficios del aprendizaje por descubrimiento que se observaron y/o se esperan son:

- Efectos positivos en la retención de información después de la instrucción en comparación con la instrucción directa tradicional.
- Desarrollo de mentes inquisitivas que se interesen en un aprendizaje permanente y autónomo.
- Exploración conjunta en la resolución de problemas, creando un círculo virtuoso de motivación.
- Recuerdo más eficiente de los conceptos si los descubren por sí mismos en comparación con aquellos que se les enseñan directamente.
- Los enfoques constructivistas son benéficos para el aprendizaje por descubrimiento

Algunas limitaciones del aprendizaje por descubrimiento:

- En algunas situaciones se pueden encontrar retroalimentación inconsistente o engañosa, errores de codificación, atribuciones causales erróneas y elaboraciones inadecuadas, comparado con instrucciones directas.
- Se requiere que el alumno investigue o realice procedimientos mientras recibe poca o ninguna ayuda.
- Las demandas cognitivas en los niños pequeños pueden obstaculizar el aprendizaje por sus conocimientos limitados al integrar información adicional.
- La instrucción directa proporciona ejemplos prácticos, técnicas de andamiaje, explicación explícita y retroalimentación que son beneficiosas para el aprendizaje.
- El descubrimiento o la capacidad de encontrar soluciones necesita algún conocimiento básico sobre el tema.

En el artículo se plantea la necesidad de implementar nuevos modelos de enseñanza que sean acordes a los nuevos entornos tecnológicos y culturales en los que se desenvuelven los niños y jóvenes. De manera que un joven competente en tecnología o alfabetizado digitalmente:

- Disponga de habilidades para el acceso a la información y el uso de recursos tecnológicos.
- Posea las capacidades cognitivas para transformar la información en conocimiento.
- Sea capaz de utilizar formas expresivas para difundir la información y comunicarse con otros compañeros.

Se consideran muy útiles las metodologías activas del aprendizaje como el aprendizaje por descubrimiento que se revisó en esta ocasión u otras estrategias como el aprendizaje basado en problemas, el aula invertida, o la gamificación, que colocan al estudiante y al aprendizaje como los protagonistas del acto formativo.

La programación del robot mBot proporciona una experiencia de aprendizaje interdisciplinario que une las matemáticas, la física y la robótica.

A medida que los estudiantes interactúan con el robot, exploran las aplicaciones prácticas de los conceptos teóricos que aprenden en el aula. Por ejemplo, pueden observar cómo los cambios en el movimiento del robot afectan su trayectoria, vinculando la geometría con principios físicos como el movimiento, la velocidad y la fuerza.

Los estudiantes son testigos de primera mano de cómo los conceptos matemáticos se traducen en movimientos en el mundo real.

Este enfoque interdisciplinario enciende la curiosidad y alimenta la pasión por el aprendizaje STEAM.

6. Bibliografía y Referencias

- [1] Alfieri, L., Brooks, P., Aldrich, N., Tenenbaum, H. Does discovery-based instruction enhance learning?. *Journal of Educational Psychology*, Vol 103(1), 1-18, 2011.

- [2] Bordignon, F., Iglesias, A. Introducción al pensamiento computacional. Editorial Universitaria, EDUCAR S.E, 2020. <https://biblioteca-repositorio.clacso.edu.ar/bitstream/CLACSO/2379/1/introduccion-pensamiento-computacional.pdf>
- [3] Cáceres, L. La robótica. Issuu, 2023. https://issuu.com/luisfcaceres/docs/roboticaluieseeeyrhf.docx_e8506833bd4d9e.
- [4] Caro, D. Enfoque STEAM: Retos y oportunidades para los docentes. Revista Internacional de Pedagogía e Innovación Educativa, 3(1), 229-244, 2023. <https://editic.net/journals/index.php/ripie/article/view/144/131>.
- [5] Carranza, M. Mejora del rendimiento académico en instituciones de nivel superior a través de la implementación del aprendizaje basado en proyectos. MQRInvestigar, 8(1), 5388-5406, 2024. <https://www.investigarmqr.com/ojs/index.php/mqr/article/view/1176/4322>.
- [6] Cuenca, Y. From Lego blocks to block programming languages, a current mediato include the teaching and learning of programming in schools and society. Rev. Tecnológica, la Paz, 19(25), 2023. http://revistasbolivianas.umsa.bo/scielo.php?pid=S1729-75322023000100005&script=sci_arttext&lng=en.
- [7] Desislava G, Tsvetanka G. Developing Mathematical Competencies Through Makeblock mBot Programming in Computer Modelling Education. TEM Journal, 12(4), 2437-2447, 2023. https://www.temjournal.com/content/124/TEMJournalNovember2023_2437_2447.pdf.
- [8] Espinoza, E. Aprendizaje por descubrimiento Vs aprendizaje tradicional. Revista Transdiciplinaria de Estudios Sociales y Tecnológicos, 2(1), 73–81, 2022. <https://revista.excedinter.com/index.php/rtest/article/view/38/35>.
- [9] Gómez, A., Ostos, O. El constructivismo y el construccionismo. Revista Interamericana De Investigación Educación Y Pedagogía RIIEP, 11(2), 115-120, 2018. <https://revistas.usantotomas.edu.co/index.php/riiep/article/view/4777>.
- [10] Hidayatul, A. Growth of critical thinking ability in middle-school students: overview of discovery learning model. Indonesian Journal of Research and

- Educational Review, 3(3), 139 - 147, 2024. <https://etdci.org/journal/ijrer/article/view/1724/1034>.
- [11] Koto, I. Teaching and Learning Science Using YouTube Videos and Discovery Learning in Elementary School. *Mimbar Sekolah Dasar*, 7(1), 106-118, 2020. <https://files.eric.ed.gov/fulltext/EJ1264962.pdf>.
- [12] Ozdem, Y., Bilican, K. Discovery Learning—Jerome Bruner. In: Akpan, B., Kennedy, T.J. (eds) *Science Education in Theory and Practice*. Springer Texts in Education, Cham, 2020. https://doi.org/10.1007/978-3-030-43620-9_13.
- [13] Papert, S. *Mindstorms*, Nueva York, Basic Books, 1980. <https://dl.acm.org/doi/pdf/10.5555/1095592>.
- [14] Pérez, M., Frutos, L., Ferrer, E. La robótica educativa y la educación tecnológica. *Órbita Científica*, 30(128), 2024. <http://revistas.ucpejv.edu.cu/index.php/rOrb/article/view/2565/2928>.
- [15] Restrepo, B. Aprendizaje basado en problemas (ABP): una innovación didáctica para la enseñanza universitaria. *Educación y educadores*, No. 8, 9-20, 2005. <https://educacionyeducadores.unisabana.edu.co/index.php/eye/article/view/562/654>.
- [16] Rosero, O. Fundamentos Teóricos del uso de la Robótica Educativa. **Ciencia Latina**: Revista Multidisciplinar, 8(1), 6364-6375, 2024. <https://ciencialatina.org/index.php/cienciala/article/view/9979/14680>.
- [17] Sáez, J., Sevillano, M., Vazquez, E. The effect of programming on primary school students' mathematical and scientific understanding: educational use of mBot. *Education Tech Research Dev* 67, 1405–1425, 2019. <https://doi.org/10.1007/s11423-019-09648-5>.
- [18] Sáez, J., Buceta, R. El robot mBot para el aprendizaje de coordenadas cartesianas en Educación Secundaria. *Pixel-Bit, Revista de Medios y Educación*, No. 66, 2023. <https://redined.educacion.gob.es/xmlui/bitstream/handle/11162/241615/Robot.pdf?sequence=1&isAllowed=y>.
- [19] Voštinár P. Teaching programming using eduScrum methodology. *Peer J Computer Science* 10: e1822, 2024. <https://doi.org/10.7717/peerj-cs.1822>. <https://peerj.com/articles/cs-1822/>.

La sinergia entre el microaprendizaje, los chatbots y el aprendizaje autodirigido

Joel Ricardo Jiménez Cruz

UAM-I

XXXVII Simposio Internacional SOMECE 2024 y en las Jornadas Internacionales de Sistemas y Ambientes Educativos (JISAE) 2024, 15 pp.

Resumen

Este artículo explora el potencial de la inteligencia artificial (IA), en particular el microaprendizaje proporcionado a través de chatbots, para revolucionar la educación al apoyar el aprendizaje autodirigido. El artículo examina las definiciones y los beneficios de estos conceptos, basándose en la investigación actual para ilustrar su potencial sinérgico. Además, aborda las limitaciones y las ventajas en torno a la IA generativa en la educación. Como un ejercicio demostrativo se muestra la aplicación de estos conceptos en la programación en Python.

Introducción

El panorama educativo se está transformando rápidamente, impulsado por los avances tecnológicos y las necesidades cambiantes de los estudiantes. El microaprendizaje, caracterizado por unidades de aprendizaje concisas y enfocadas, ofrece un enfoque prometedor para abordar la reducción de la capacidad de atención y mejorar la retención de conocimientos. Al mismo tiempo, los chatbots, impulsados por IA, brindan una plataforma para experiencias de aprendizaje personalizadas a través de interacciones similares a las humanas. Estas innovaciones, cuando se combinan con los principios del aprendizaje autorregulado, donde los estudiantes toman un papel activo y responsable en su proceso de aprendizaje, pueden crear una poderosa sinergia para una educación efectiva y atractiva. Este artículo examina la intersección de estos conceptos transformadores y explora su potencial para reformular la educación moderna.

A continuación se revisan los conceptos sinérgicos del microaprendizaje, los chatbots y el aprendizaje autodirigido, acompañados por el marco de la agencia del estudiante y el diseño estructural.

El Microaprendizaje

El microaprendizaje es una estrategia de enseñanza que pone énfasis en la entrega de contenido de aprendizaje en unidades pequeñas y digeribles, que suelen durar alrededor de 10 minutos. Este enfoque se alinea con las preferencias cognitivas de los estudiantes contemporáneos que están acostumbrados a acceder a la información, que proporciona Internet, de manera rápida y eficiente.

Los estudios revelan que el microaprendizaje puede mejorar significativamente la retención de conocimientos y mejorar el desempeño educativo en comparación con los métodos de aprendizaje tradicionales de mayor duración (Zhang, West, 2020).

El microaprendizaje se ha implementado en diversos contextos educativos y de entrenamiento, lo que ha generado el desarrollo de varios productos multimedia, incluidos videos de aprendizaje, podcasts, juegos interactivos y módulos en línea.

También se ha abordado el desarrollo de modelos instruccionales que sirven como marco o base para los procesos de enseñanza aprendizaje y para la creación de productos ad-hoc para los diferentes formas aprendizaje, incluido el microaprendizaje.

La duración de la implementación del microaprendizaje sigue siendo una consideración crítica. Si bien se puede adaptar en función de los objetivos y objetos de aprendizaje específicos, la investigación sugiere un período mínimo de implementación de 14 días o 2-3 sesiones para evaluar adecuadamente su impacto en los resultados de aprendizaje.

Los chatbots o agentes conversacionales

Los chatbots han surgido como una herramienta poderosa para la comunicación y la difusión de información en varios sectores, incluida la educación. Impulsados por los avances en inteligencia artificial, en particular del procesamiento del lenguaje natural, los chatbots pueden comprender y responder al lenguaje humano, simulando conversaciones naturales.

En entornos educativos, los chatbots se están integrando en plataformas de aprendizaje para personalizar la experiencia del aprendizaje y brindar apoyo personalizado a los estudiantes.

Los chatbots pueden funcionar como asistentes de enseñanza, ofreciendo orientación y respondiendo preguntas, o incluso asumiendo el papel de líderes de proyectos, ayudando a los estudiantes a mantenerse organizados y al día con sus tareas. Esta versatilidad permite a los educadores crear entornos de aprendizaje más interactivos y atractivos que se adapten a las necesidades individuales de los estudiantes.

La concurrencia del microaprendizaje y los chatbots

La combinación de la tecnología del microaprendizaje y los chatbots presenta una oportunidad particularmente atractiva para mejorar la educación. El microaprendizaje proporciona el marco ideal para ofrecer contenido de aprendizaje en porciones pequeñas a través de chatbots.

La naturaleza conversacional de los chatbots permite a los estudiantes volver a revisar los materiales de aprendizaje, solicitar aclaraciones y recibir comentarios instantáneos, lo que promueve el aprendizaje activo y una comprensión más profunda.

Este enfoque se alinea con los principios establecidos de la psicología cognitiva, que sugieren que el cerebro procesa la información de manera más efectiva cuando se presenta en fragmentos más pequeños y manejables. Además, el elemento interactivo del microaprendizaje basado en chatbots puede mejorar la motivación y el compromiso de los estudiantes (que se verá más adelante), lo que conduce a mejores resultados de aprendizaje.

Dimensiones del aprendizaje autodirigido

La agencia de un individuo considerando el aspecto estudiantil se refiere al empoderamiento de los estudiantes para dar forma a su trayectoria educativa, haciéndose cargo de su proceso de aprendizaje al tomar decisiones y dirigir sus caminos académicos. Esto implica un fuerte sentido de influencia personal, que abarca intencionalidad, previsión, autorreactividad y

autorreflexión en sus acciones. En el corazón de la agencia estudiantil están la voz, la elección, el interés, el significado, el esfuerzo y el control dentro del entorno del aula. Con estos ingredientes, los estudiantes participan, crecen en confianza y se convierten en estudiantes eficaces. El creer en su capacidad de influencia en los resultados, pueden mejorar significativamente su motivación (Main, 2024).

La agencia del estudiante está influenciada por el aprendizaje autodirigido que incluye dos dimensiones (figura 1): (1) la teoría de la autodeterminación (motivación y desempeño); y (2) el aprendizaje autorregulado (automonitoreo y autogestión). La autogestión está relacionada con el control de tareas. Se centra en actividades externas que influyen en el proceso de aprendizaje, como la implementación de objetivos de aprendizaje y la gestión de recursos y apoyo para el aprendizaje. Estas actividades se evalúan y negocian constantemente (Yin, Goh, Yang, Xiaobin, 2021).

Por otro lado, el automonitoreo implica procesos cognitivos y metacognitivos que incluyen la capacidad de monitorear las propias estrategias de aprendizaje y pensar sobre el pensamiento. El automonitoreo significa que el alumno es responsable de la construcción del aprendizaje personal. El automonitoreo interno por sí solo no es suficiente para promover la mejora cognitiva. En consecuencia, para desarrollar de manera efectiva las habilidades del aprendizaje autodirigido, los instructores deben proporcionar retroalimentación externa para apoyar el automonitoreo de los alumnos (Zhu, Bonk, Doo, 2020).

La segunda dimensión del modelo es la motivación y el desempeño que están incluidas en la teoría de la autodeterminación, que se refiere al involucramiento y las tareas a realizar. La motivación puede iniciar y mantener el esfuerzo hacia el aprendizaje y la consecución de objetivos cognitivos. Los estudiantes motivados pueden mejorar su rendimiento académico. La motivación de los estudiantes influye en el aprendizaje autorregulado cuyos componentes cognitivos del autocontrol y la autogestión se influyen entre sí (Stover, Bruno, Uriel, Fernández, 2017).

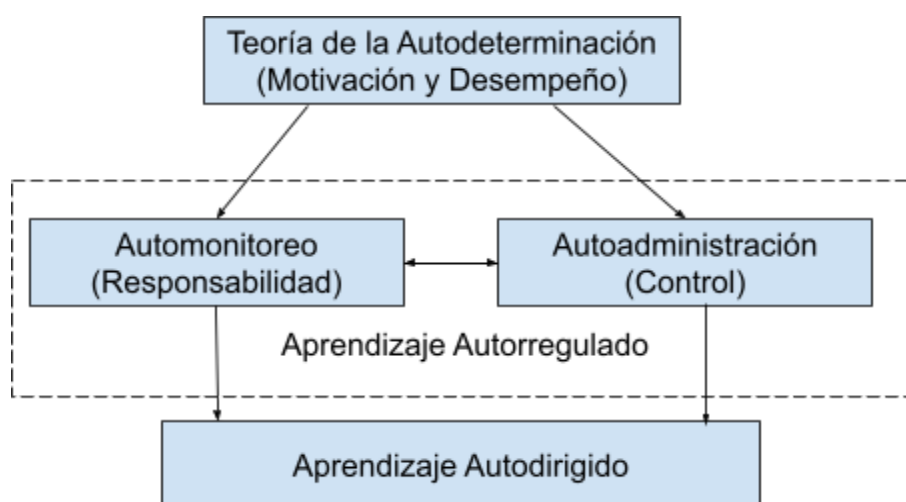


Figura 1. Dimensiones del Aprendizaje Autodirigido

La teoría de la autodeterminación postula que el interés por el estudio y el rendimiento en el aprendizaje puede aumentar la motivación, la cual puede ser intrínseca y extrínseca.

Un estudiante intrínsecamente motivado realiza una actividad por su satisfacción, diversión o desafío inherente. La motivación intrínseca obedece a tres necesidades psicológicas; autonomía, competencia y relación afectiva. La autonomía se refiere a acciones selectivas y autoiniciadas. La competencia se refiere a la percepción de que el individuo ha realizado una tarea de manera efectiva y con confianza. La relación afectiva se define como el apoyo que un individuo recibe o brinda a otros (que podría ser el agente conversacional), durante las interacciones (Stover, Bruno, Uriel, Fernández, 2017).

Por otro lado, un estudiante extrínsecamente motivado realiza una actividad debido a fuerzas externas, presiones o recompensas (teoría de la recompensa de Pavlov).

Los estudiantes con motivación intrínseca manejan tareas desafiantes de manera persistente y más satisfactoria a medida que adquieren competencia en los nuevos conocimientos.

La motivación por aprender genera una mayor asimilación de conocimiento y por ende un mejor logro de los objetivos de estudio.

El aprendizaje autorregulado puede guiar el aprendizaje a través de la planificación, el establecimiento de metas, la organización, el autocontrol y la autoevaluación. Estos procesos son importantes en la resolución de problemas. Los procesos de autorregulación como el establecimiento de metas, el control de la comprensión, las estrategias y las fases (previsión, desempeño y reflexión) se utilizan como heurísticas para describir las interacciones entre los estudiantes, el profesor y ahora la IA (Hartley, Hayak, Ko, 2024).

La IA no solo puede apoyar la instrucción, sino también puede promover el desarrollo de habilidades para el aprendizaje autorregulado y posiblemente sus características pedagógicas. Estas nuevas herramientas de IA pueden centralizar muchos dispositivos, sistemas y procesos en un solo lugar, de la misma manera como lo hacen los teléfonos inteligentes.

De la misma manera se espera que un sistema de microaprendizaje autorregulado y apoyado por un chatbot pueda exhibir comportamientos de motivación y rendimiento similares.

El agente conversacional también podría facilitar el microaprendizaje entre pares, conectando a los estudiantes con intereses comunes con objetivos y objetos de aprendizaje o moderando debates y actividades grupales.

Sobre el Diseño Instruccional

La implementación del microaprendizaje con la tecnología del chatbot tiene dos dimensiones esenciales: el desenvolvimiento y capacidades del chatbot para brindar respuestas precisas, y el modelo instruccional empleado. El chatbot interactúa con el usuario a través de la fluidez del lenguaje natural a través de texto o voz e incorpora elementos multimedia.

El microaprendizaje con sus características de corto plazo, fácil digestión y adaptabilidad, desarrollan, a su vez, las habilidades de corto plazo de los estudiantes. El aumento en la alta disponibilidad de recursos ha proporcionado a los estudiantes múltiples vías para el aprendizaje independiente, que les requieren un papel más activo y responsable en el proceso de

aprendizaje, en concordancia con el aprendizaje autorregulado y autodeterminado (Putri, Mustaji, Sumarno, 2024).

De acuerdo a estas características, el modelo instruccional basado en sistemas de microaprendizaje debe apoyar un contenido conciso y de fácil comprensión, maximizando la eficiencia y la accesibilidad al aprendizaje, considerando y adecuando el modelo propuesto por Baumgartner (2013) que establece tres fases: absorción, adquisición y construcción del conocimiento.

- En la fase de absorción, se presenta a los estudiantes la planificación y el contenido básico de los materiales didácticos que les permita absorber el conocimiento mínimo requerido y lleven a cabo las primeras actividades introductorias (tiene ciertos rasgos derivados del conductismo).
- En la fase de adquisición, el diseño instruccional asume que el aprendizaje es un proceso activo. Los estudiantes interactúan con el chatbot para adquirir y construir conocimiento a través de la retroalimentación instantánea, evaluación y reflexión, formando así trozos (chunks) de experiencias de aprendizaje que retiene la memoria a corto plazo o memoria de trabajo (tiene cierto parentesco con el cognitivismo).
- En la fase de construcción, los estudiantes y el chatbot y/o el profesor interactúan para crear una solución conjunta que resuelve un problema puntual o una parte del problema general. Se intenta lograr una solución clara y sencilla dentro del tiempo limitado por el objeto de aprendizaje (tiene fuertes vínculos con el constructivismo social).

Faltaría considerar una cuarta fase de una perspectiva conectivista que reconoce que el aprendizaje ocurre dentro de redes de interacciones y que los espacios dialógicos pueden ser habilitados por la IA generativa, entre los que se encuentran: el aprendizaje colaborativo, la conectividad del conocimiento y la integración de la teoría y la práctica (Pratschke, 2024). El aprendizaje conectivista potencia las herramientas de la IA generativa, no solo como facilitadoras de las interacciones sino también como agentes de interacción que crean nuevas interacciones de aprendizaje en diferentes niveles (Liang & Bai, 2024).

En estas fases se van sugiriendo, empleando y dispensando instrumentos o recursos que se pueden categorizar en (Hartley, Hayak, Ko, 2024):

- materiales didácticos (por ejemplo, libros de texto, recursos multimedia),
- herramientas de contenido (por ejemplo, generadores de mapas conceptuales, IDEs),
- retroalimentación y evaluaciones (por ejemplo, ejercicios resueltos, scripts de código)
- herramientas de planificación (por ejemplo, calendario, objetivos, tareas).

Por otra parte, los estudiantes autorregulados y autodeterminados se pueden evaluar y monitorear, mientras que la planificación y el establecimiento de objetivos corresponden a responsabilidades compartidas.

Convergencia entre el microaprendizaje y el agente conversacional

Es deseable que el diseño instruccional basado en un agente conversacional contemple estrategias motivacionales y educativas se cumplan con una funcionalidad conversacional verificable, con metas educativas y con roles pedagógicos que fortalezcan la capacidad de los estudiantes para aprender de manera autónoma, autodirigida y a su propio ritmo.

El sistema de microaprendizaje basado en un chatbot contiene dimensiones educativas motivacionales a (Dulyarak, Boonlue, Narmluk, 2024):

- Mostrar diferentes contenidos multimedia como texto, imágenes y videos que intentan captar el interés y el disfrute de los estudiantes durante las interacciones.
- Presentar el contenido simple y preciso para permitir una rápida absorción que minimice cualquier presión y estrés innecesarios.
- Incluir elementos de control con retroalimentación afectiva y positiva que apoyan la autoevaluación, la construcción de conocimiento y la competencia.
- Permitir la práctica repetida y las consultas libres sobre un concepto específico que fomentan la interacción entre el alumno, el contenido y la IA y la valoración del autoaprendizaje.
- Ofrecer enlaces a contenidos multimedia y elementos de autoevaluación voluntaria.

Sobre un ejercicio demostrativo

Con la incorporación de la IA al modelo de microaprendizaje, como ya se ha mencionado, los estudiantes utilizan diversos instrumentos que cuando son adecuados, fundamentan el aprendizaje de los estudiantes. Este cambio hacia un enfoque multimodal del aprendizaje refleja el panorama rápidamente cambiante de la educación y la necesidad de considerar las estrategias de autorregulación y autodeterminación, que se han comentado. Los estudiantes al gestionar múltiples instrumentos, reflejan mejor sus necesidades y capacidades individuales.

A continuación se presenta un ejercicio demostrativo para el microaprendizaje del lenguaje de programación de Python que utiliza el diseño instruccional comentado que contempla las fases de absorción, adquisición y fase de construcción y que se llevan a cabo utilizando los instrumentos: herramientas de planeación, materiales didácticos, herramientas de contenido, y retroalimentación y evaluación.

Para llevar a cabo los ejercicios se utilizará el agente conversacional de ChatGPT para realizar las consultas y el IDE de Replit para evaluar y modificar los códigos realizados.

Ejemplo de una cápsula de **materiales didácticos** de instrucción: el uso de la instrucción de selección if-then

prompt: ¿Cómo uso la instrucción if-then en el lenguaje Python para controlar el flujo de un programa?

Respuesta de ChatGPT:

El agente conversacional presenta:

- La estructura básica de if-elif-else:
- Un ejemplo 1: Instrucción if-else simple
- Un ejemplo 2: Usando if-elif-else

- Un ejemplo 3: Instrucciones if anidadas
- Puntos clave a recordar: indentación (sangrado) y operadores lógicos y de comparación

Análisis:

- Integridad: como se señaló en el análisis anterior, ChatGPT brindó una explicación integral de las declaraciones if-then, que abarca los conceptos básicos de las declaraciones condicionales, la sangría y el uso de "else" y "elif" para manejar múltiples condiciones.
- Sintaxis específica de Python: la respuesta demostró con precisión el uso de la sintaxis de Python, incluidos los dos puntos después de las declaraciones condicionales y la sangría para definir bloques de código. Esto es fundamental para los estudiantes, ya que los errores de sintaxis son comunes en la programación y pueden causar frustración.
- Ejemplo práctico: el ejemplo de código proporcionado, aunque simple, ilustra de manera eficaz cómo se puede utilizar una declaración if-then para determinar si un número es par o impar. Los ejemplos prácticos (fase de construcción) son esenciales para ayudar a los estudiantes a comprender los conceptos de programación y ver cómo se pueden aplicar.

Herramientas de contenido: Pruebas y depuración

Prompt: ¿Qué entorno de desarrollo (ide) en línea puedo usar para probar y depurar mi código Python mientras aprendo?

Respuesta de ChatGPT:

Menciona algunos de los **IDEs más populares** en línea para principiantes, con sus ventajas y desventajas, aquí un resumen que proporciona:

Replit: Ideal para colaboración en tiempo real y fácil de usar. Es versátil y permite compartir proyectos fácilmente.

Google Colab: Perfecto para ciencia de datos y aprendizaje automático, ideal para quienes trabajan con Jupyter Notebooks.

PythonAnywhere: Excelente para proyectos Python puros, incluidos alojamientos web.

Trinket: Muy amigable para principiantes y útil en entornos educativos.

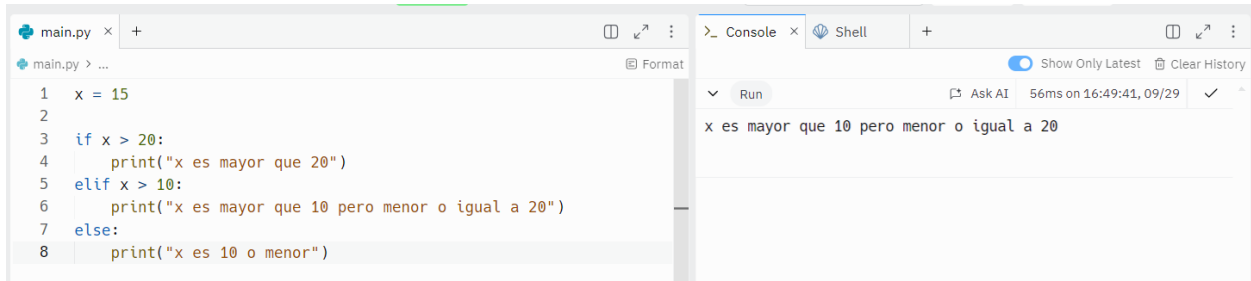
JDoodle: Ideal para pruebas rápidas de código sin necesidad de registro.

Glitch y CodeSandbox: Perfectos para proyectos web y colaboración, aunque no están tan centrados en Python.

Si estás empezando, Replit y Trinket son los más recomendados por su simplicidad. Si te interesa la ciencia de datos o proyectos interactivos, Google Colab es una excelente opción.

Como muestra, si se corre el ejemplo 2 que proporciona ChatGPT (fase de construcción para retroalimentación y evaluación), la respuesta en el IDE de Replit

(<https://replit.com/languages/python3>) para modificarla si se desea, es:

A screenshot of a Replit IDE interface. The left pane shows a Python file named 'main.py' with the following code:

```
1 x = 15
2
3 if x > 20:
4     print("x es mayor que 20")
5 elif x > 10:
6     print("x es mayor que 10 pero menor o igual a 20")
7 else:
8     print("x es 10 o menor")
```

The right pane shows the 'Console' tab with the output of the script: 'x es mayor que 10 pero menor o igual a 20'. Above the console, there are tabs for 'Console' and 'Shell', and a 'Run' button. The top bar shows 'main.py' and a 'Format' button.

Análisis:

- Variedad de herramientas: ChatGPT ofreció una lista útil que se adapta a diferentes estilos y preferencias de aprendizaje.
- Recomendaciones contextuales: Para un principiante, los intérpretes en línea son los recomendados ya que los estudiantes reciben más apoyo en las etapas iniciales y pasan gradualmente a un aprendizaje más independiente a medida que desarrollan sus habilidades.

Análisis más profundo:

- Integración de IA en IDE: En Replit se cuenta con la asistencia inteligente en la depuración en tiempo real y sugerencias de codificación que pueden mejorar significativamente el proceso de aprendizaje.
- Equilibrio entre retroalimentación y exploración: si bien las herramientas de depuración impulsadas por IA son útiles, es fundamental lograr un equilibrio en que se tengan oportunidades para lidiar con errores, desarrollar estrategias propias de resolución de problemas y desarrollar habilidades de depuración. La dependencia excesiva de la asistencia de IA podría obstaculizar el desarrollo de estas habilidades esenciales.

Retroalimentación y evaluación: desafío if-then

Prompt: Dame un problema para resolver que requiera el uso de una declaración if-then. Y evalúa la precisión de mi respuesta.

ChatGPT: Aquí tienes un problema que requiere el uso de una declaración if-then en Python: Escribe un programa en Python que le pida al usuario ingresar un número. Si el número es mayor o igual a 100, el programa debe imprimir "El número es grande". Si el número es menor a 100, el programa debe imprimir "El número es pequeño".

Pasos:

Solicita al usuario que ingrese un número.

Utiliza una declaración if-else para verificar si el número es mayor o igual a 100.

Imprime el mensaje adecuado según la comparación.

Ejemplo de salida:

Si el usuario ingresa 150, el programa debe imprimir: "El número es grande".

Si el usuario ingresa 50, el programa debe imprimir: "El número es pequeño".

Prueba a escribir el código, y cuando lo tengas, envíamelo para evaluar su precisión.

Prompt: Aquí está mi solución

```
numero = int(input("Ingrese un número: "))
```

```
if numero >= 100:  
    print("El número es grande.")  
else:  
    print("El número es pequeño.")
```

ChatGPT: ¡Tu solución es precisa! Has implementado correctamente el uso de la declaración if-else en Python. Aquí te doy una evaluación detallada:... Lo que has hecho bien...

ChatGPT: Aunque el programa es correcto, podrías agregar un pequeño manejo de errores para asegurarte de que el usuario realmente ingresa un número válido, por ejemplo, usando un bloque try-except. Esto evitará que el programa se bloquee si el usuario introduce algo que no sea un número.

ChatGPT: Ejemplo con manejo de errores:

```
try:  
    numero = int(input("Ingrese un número: "))  
  
    if numero >= 100:  
        print("El número es grande.")  
    else:  
        print("El número es pequeño.")  
except ValueError:  
    print("Error: Por favor, ingresa un número válido.")
```

ChatGPT: Esto mejorará la robustez del programa, pero tu solución inicial ya es totalmente funcional. ¡Bien hecho!

Esta sugerencia hace que el estudiante vaya aprendiendo más, paso a paso, como lo sugiere el método de microaprendizaje.

Análisis:

- Relevancia del problema: El problema planteado, aunque simple, se alineaba con el requisito de la indicación de usar una declaración if-then. Sin embargo, como se señaló en el análisis anterior, la suposición con respecto a la entrada del usuario resalta la necesidad de instrucciones más explícitas y consideraciones sobre el nivel de conocimiento actual del alumno.
- Buena retroalimentación: La evaluación de ChatGPT sugirió una mejora al código.

Cuando ChatGPT haga algo que no le hemos pedido:

- Se trata de mejorar la especificidad del prompt: Al indicar explícitamente los conocimientos limitados de programación, se puede pedirle al sistema que proporcione un problema que se

ajuste a sus capacidades actuales. Proporcionar un código básico con instrucciones para obtener la información del usuario puede guiar aún más a los alumnos y evitar que se queden atascados debido a conocimientos asumidos.

- Incorporación de la participación humana: para abordar las limitaciones de las evaluaciones basadas en IA, podría resultar beneficioso un enfoque en que se de la participación humana. ChatGPT se podría utilizar para generar una variedad de posibles problemas de evaluación, que un educador humano podría revisar, refinar y seleccionar, garantizando la precisión, la idoneidad y la alineación con los objetivos de aprendizaje. El educador también podría proporcionar comentarios más matizados sobre las respuestas de los estudiantes, abordando conceptos erróneos y brindando orientación más allá de lo que ChatGPT puede ofrecer actualmente.

Planificación: preparación para el examen PCEP de Python

Prompt: Quiero aprobar el examen PCEP del instituto Python dentro de tres meses.

Proporcióname una lista de temas que deba dominar y un cronograma semanal.

ChatGPT: ¡Es excelente que te prepares para el examen **PCEP (Python Certified Entry-Level Programmer)**! Este examen evalúa los conocimientos básicos de Python, y es el primer nivel de certificación del Python Institute. Para ayudarte a organizar tu estudio, aquí tienes una lista de temas clave que debes dominar y un cronograma de estudio semanal que cubre los próximos tres meses.

Temas que debes dominar para el PCEP:

1. **Sintaxis básica y estructura de Python**

- Estructura del programa Python (indentación, comentarios).
- Variables, tipos de datos, y operadores básicos.
- Funciones `print()` y `input()`.

2. **Tipos de datos y operaciones**

- Números (int, float).
- Cadenas de texto (`str`).
- Listas, tuplas, conjuntos, y diccionarios.
- Indexación, rebanado (slicing), y operadores en colecciones.
- Operadores aritméticos, de comparación, lógicos y de asignación.

3. **Control de flujo**

- Declaraciones condicionales (`if`, `else`, `elif`).
- Bucles (`while`, `for`).
- Control de bucles (`break`, `continue`).

4. **Funciones y módulos**

- Definición y uso de funciones.
- Argumentos y retorno de valores.
- Módulos integrados (como `math`, `random`) y cómo importarlos.

5. **Entrada y salida de datos**

- Lectura de datos del usuario con `input()`.

- Salida de datos con `print()`.
 - Manejo básico de archivos (abrir, leer, escribir, cerrar archivos).
 - 6. **Errores y excepciones**
 - Manejo de excepciones (`try`, `except`).
 - 7. **Conceptos básicos de programación orientada a objetos (OOP)**
 - Clases y objetos.
 - Métodos y atributos básicos.
-

Cronograma de estudio para 3 meses

Te propongo un cronograma semanal de estudio, de manera que cubras todos los temas con tiempo para la práctica y repaso. Asumo que estudiarás entre 4 a 6 horas por semana (si puedes más, mejor).

Mes 1: Fundamentos de Python

- **Semana 1:**
 - Introducción a Python: Sintaxis básica, variables, tipos de datos (enteros, flotantes, cadenas).
 - Operadores aritméticos y de comparación.
 - Ejercicios de práctica simples (por ejemplo, cálculos con operadores).
 - **Semana 2:**
 - Estructuras de datos: Listas y tuplas.
 - Operaciones con listas (agregar, eliminar, indexar, rebanar).
 - Práctica con listas: Crear y modificar listas.
 - **Semana 3:**
 - Estructuras de datos: Conjuntos y diccionarios.
 - Métodos importantes en conjuntos y diccionarios.
 - Ejercicios prácticos con diccionarios (almacenar datos, acceder a valores).
 - **Semana 4:**
 - Estructuras de control: `if`, `else`, `elif`.
 - Operadores lógicos.
 - Práctica de condicionales.
-

Mes 2: Control de Flujo y Funciones

- **Semana 5:**
 - Bucles `for` y `while`.
 - Control de bucles: `break`, `continue`.

- Ejercicios prácticos: Contadores, sumas y operaciones repetitivas.
 - **Semana 6:**
 - Definir y usar funciones.
 - Parámetros y argumentos de funciones.
 - Retorno de valores.
 - Ejercicios: Crear funciones con diferentes tipos de parámetros.
 - **Semana 7:**
 - Módulos y bibliotecas estándar.
 - Importar módulos (`import`, `from...import`).
 - Explorar módulos como `math` y `random`.
 - Práctica: Crear pequeños programas que usen funciones de bibliotecas estándar.
 - **Semana 8:**
 - Archivos: Apertura y cierre de archivos.
 - Leer y escribir archivos de texto.
 - Práctica: Crear un programa que lea y escriba archivos.
-

Mes 3: Errores, Excepciones y Repaso

- **Semana 9:**
 - Manejo de errores con `try`, `except`.
 - Tipos de excepciones comunes.
 - Práctica: Crear programas que manejen errores y validen entradas de usuario.
 - **Semana 10:**
 - Conceptos básicos de Programación Orientada a Objetos (POO).
 - Crear clases, objetos, métodos y atributos.
 - Ejercicios: Definir una clase simple y crear objetos a partir de ella.
 - **Semana 11:**
 - Repaso de todos los temas cubiertos hasta ahora.
 - Hacer exámenes de práctica o ejercicios de evaluación.
 - **Semana 12:**
 - Práctica final: Resolver ejercicios de nivel PCEP.
 - Repaso de temas que consideres más débiles.
 - Simulacro de examen para medir el tiempo y mejorar tu rendimiento.
-

Recursos útiles:

1. **Python Institute Learning Platform** (material oficial):
Python Institute
2. **W3Schools Python Tutorial**:
W3Schools Python

3. Exámenes de práctica:

Busca simuladores de exámenes en línea o preguntas de prueba.

ChatGPT: ¡Con este cronograma, estarás bien preparado para el examen PCEP! Si en algún momento sientes que necesitas ajustar el plan o más ayuda con algún tema, no dudes en pedírmelo.

Análisis:

- Plan integral: ChatGPT demostró su fortaleza en la planificación al generar un plan de estudio detallado de 12 semanas, describiendo los temas del examen PCEP y asignando tiempo para cada uno. Las herramientas de IA pueden respaldar eficazmente los procesos de planificación de los estudiantes al dividir los objetivos en subobjetivos y tareas, un aspecto fundamental del aprendizaje autoregulado.
- Se puede pedir a ChatGPT el detalle que se desee, como un cronograma diario, o cualquier nivel de granularidad que puede ser beneficioso para los estudiantes que tienen dificultades con la administración eficaz del tiempo y necesitan o necesitan dividir los objetivos de aprendizaje más grandes en partes manejables para estructurar su proceso de aprendizaje.

Consideraciones para la implementación:

- Flexibilidad y adaptación: si bien los planes estructurados son beneficiosos, los estudiantes deben poder adaptarse y ajustarse en función de su propio progreso y comprensión. De acuerdo a la teoría de la autodeterminación se necesita que los estudiantes se motiven para seguir el plan generado por IA como punto de partida y modificarlo según sea necesario, asumiendo la responsabilidad del proceso de aprendizaje.
- Integración del seguimiento del progreso: la incorporación de mecanismos para que los estudiantes realicen un seguimiento de su progreso y realicen ajustes en función de su desempeño sería crucial. Esto podría implicar el uso de IA para generar cuestionarios o preguntas de práctica alineadas con el plan de estudio, brindar retroalimentación sobre su desempeño y sugerir áreas en las que podrían necesitar concentrar más atención.
- Motivación y responsabilidad: Las herramientas de IA carecen de la influencia de los instructores humanos en términos de asignar calificaciones y brindar motivación externa. Alentar a los estudiantes a establecer sus propios objetivos, reflexionar sobre su progreso y encontrar una motivación intrínseca será esencial para el éxito a largo plazo.

La incorporación de las respuestas reales de Python generadas por ChatGPT y el IDE de Replit, junto con un análisis más profundo, fortalecen la idea de brindar ejemplos concretos y resaltar tanto el potencial como las limitaciones del microaprendizaje y las teorías de autorregulación y autodeterminación respaldadas por la IA en el contexto del aprendizaje de la programación en Python. El análisis enfatiza la importancia de la IA, la supervisión humana y un enfoque equilibrado que aproveche la asistencia de la IA al tiempo que fomenta la autonomía de los estudiantes, el pensamiento crítico y las habilidades de aprendizaje autorregulado y autodeterminado.

Conclusión

La integración del microaprendizaje, los chatbots y los principios del aprendizaje basado en la inteligencia artificial representa un cambio de paradigma en la educación. Estas tecnologías ofrecen el potencial de personalizar el aprendizaje, mejorar la participación y empoderar a los estudiantes para que se apropien de sus recorridos educativos. Sin embargo, como ocurre con cualquier tecnología disruptiva, se deben considerar cuidadosamente las implicaciones éticas y los posibles inconvenientes. Las investigaciones futuras deberían centrarse en:

- Comprobar la eficacia y las limitaciones de los agentes conversacionales de las diferentes plataformas que existen hoy en día como Gemini, Copilot, ChatGPT, etc.
- Desarrollar métricas de evaluación estandarizadas para permitir comparaciones objetivas entre diferentes sistemas de IA y hacer un seguimiento de su progreso a lo largo del tiempo.
- Investigar el impacto a largo plazo del microaprendizaje respaldado por IA en la motivación, la participación y el rendimiento de los estudiantes.
- Abordar las preocupaciones éticas relacionadas con la privacidad de los datos, el sesgo algorítmico y el acceso equitativo para garantizar una implementación responsable e inclusiva de la IA en la educación.

A medida que la tecnología de IA continúa evolucionando, se tiene el potencial de transformar la educación en una experiencia más personalizada, atractiva y eficaz para todos los estudiantes. Sin embargo, hacer realidad este potencial requiere investigación, desarrollo y un examen crítico continuos tanto de los beneficios como de los desafíos de integrar la IA en el proceso del microaprendizaje.

Reconocimiento

Se hace un reconocimiento a la ayuda prestada por Google NotebookLM para la escritura de este documento.

Referencias

Baumgartner, P. (2013). Educational dimensions of microlearning—towards a taxonomy for microlearning. In M. S. Martina Roth & Peter A. Bruck (Ed.), Designing microlearning experiences—building up knowledge in organisations and companies. Innsbruck University Press.

https://portfolio.peter-baumgartner.net/files/pdf/2013/Baumgartner_2013_Educational%20Dimensions%20for%20MicroLearning.pdf

Dulyarak, D., Boonlue D., Narmluk, M. (2024). Investigating the Components of Chat-Bot-Driven Micro-Learning: Enhancing Problem-Solving Competency in Modern Teaching. Conference Proceedings ICTAECT#2.

https://www.researchgate.net/profile/Dulyarak-Chumdoem-2/publication/383119749_Investigating_the_Components_of_Chart-Bot-Driven_Micro-Learning_Enhancing_Problem-Solving_Competency_in_Modern_Teaching/links/66bd6e8b311cbb09493947b6/Investigating-the-Components-of-Chat-Bot-Driven-Micro-Learning-Enhancing-Problem-Solving-Competency-in-Modern-Teaching.pdf

Hartley, K., Hayak, M., Ko, U. (2024). Artificial Intelligence Supporting Independent Student Learning: An Evaluative Case Study of ChatGPT and Learning to Code. *Educ. Sci* 14, 120. <https://doi.org/10.3390/educsci14020120>

Liang, E. S., & Bai, S. (2024). Generative AI and the future of connectivist learning in higher education. *Journal of Asian Public Policy*, 1–23. <https://doi.org/10.1080/17516234.2024.2386085>

Main, P. (2024). The Role of Student Agency in Fostering Lifelong Learners. <https://www.structural-learning.com/post/student-agency#>

Pratschke, B.M. (2024). The New Hybrid. In: *Generative AI and Education*. SpringerBriefs in Education. Springer, Cham. https://doi.org/10.1007/978-3-031-67991-9_3 <https://arxiv.org/pdf/2309.12468>

Putri, L., Mustaji, M., & Sumarno, A. (2024). How Developing and Implementation Microlearning in Educational Setting?: A Literature Review. *Jurnal Teknologi Pendidikan : Jurnal Penelitian dan Pengembangan Pembelajaran*, 9(3), 432-441. doi:<https://doi.org/10.33394/jtp.v9i3.11698>

Stover , J., Bruno, F., Uriel, F., & Fernández L. (2017). Teoría de la Autodeterminación: una revisión teórica. *Revista de Psicología y Ciencias Afines*, 14(2),105-115. <https://www.redalyc.org/articulo.oa?id=483555396010>

Yin, J., Goh, T.-T., Yang, B., & Xiaobin, Y. (2021). Conversation Technology With Micro-Learning: The Impact of Chatbot-Based Learning on Students' Learning Motivation and Performance. *Journal of Educational Computing Research*, 59(1), 154-177. <https://doi.org/10.1177/0735633120952067>

Zhang, J., & West, R. E. (2020). Designing Micro learning Instruction for Professional Development Through a Competency Based Approach. *Tech Trends*, 64(2), 310-318.

Zhu, M., Bonk, C.J. & Doo, M.Y. (2020). Self-directed learning in MOOCs: exploring the relationships among motivation, self-monitoring, and self-management. *Education Tech Research Dev* 68, 2073–2093 <https://doi.org/10.1007/s11423-020-09747-8>

Modelos computacionales de agentes robóticos bioinspirados: del movimiento aleatorio a la conducta adaptativa

Autor: Joel Ricardo Jiménez Cruz

Universidad Autónoma Metropolitana - Iztapalapa

Contenido

Prefacio

Capítulo 1. Caminatas aleatorias y exploración básica

Capítulo 2. Estrategias de búsqueda y forrajeo

Capítulo 3. Comportamientos emergentes: los vehículos de Braitenberg

Capítulo 4. Modelando el comportamiento con máquinas de estado finito (FSM)

Capítulo 5. Selección de acción en agentes autónomos

Capítulo 6. Aprendizaje asociativo en abejas

Capítulo 7. Forrajeo y regreso al nido en hormigas

Capítulo 8. Aprendizaje por reforzamiento

Capítulo 9. Enseñanza Animal - Robot

Apéndices del libro

Apéndice A. Herramientas matemáticas y computacionales

Apéndice B. Extensiones y proyectos avanzados

Apéndice C. Código Python completo

Referencias bibliográficas

Prefacio

El presente libro ofrece una **visión integradora** sobre cómo la inteligencia y el comportamiento pueden **emerger** en sistemas **naturales y artificiales**.

A través de modelos computacionales y simulaciones en **Python**, se analiza la evolución de la conducta desde los **movimientos aleatorios más simples** hasta las **formas adaptativas de aprendizaje e inteligencia**.

El texto está dirigido a estudiantes, investigadores y entusiastas de la **robótica bioinspirada**, la **neurociencia computacional** y la **inteligencia artificial**, interesados en comprender los mecanismos que subyacen al comportamiento autónomo y autoorganizado de los agentes.

El enfoque es **interdisciplinario**, combinando fundamentos **biológicos, matemáticos y computacionales**.

Cada capítulo desarrolla un **modelo computacional de un agente robótico**, inspirado en un proceso biológico o cognitivo, y se acompaña de **explicaciones teóricas, simulaciones prácticas y ejercicios** diseñados para fomentar el **pensamiento computacional** y la **comprensión del comportamiento emergente**.

Estructura del libro

Cada capítulo incluye:

- **Sección teórica**, donde se explica el principio biológico y su formalización matemática.
- **Ejemplo de simulación**, implementado en Python, que demuestra el modelo.
- **Ejercicios teóricos y prácticos**, orientados al análisis y la experimentación.
- **Reflexiones finales**, que relacionan el modelo con fenómenos biológicos reales.

Al final del libro se incluyen tres **apéndices** que sirven como referencia técnica y conceptual:

● **Apéndice A. Herramientas matemáticas y computacionales**

Contiene conceptos matemáticos y explicaciones de código necesarias para comprender los modelos. En el texto, aparece un círculo rojo que indica la referencia a este apéndice para que se pueda ahondar en información relevante.

● *Si necesita repasar un concepto, siga la referencia antes de continuar.*

● **Apéndice B. Extensiones y proyectos avanzados**

Amplía los temas tratados en cada capítulo y propone proyectos adicionales de estudio, investigación o simulación. Las referencias a este apéndice aparecen en un círculo verde en el texto.

● Si desea profundizar en un tema, vaya a la referencia.

● Apéndice C. Código Python completo

Contiene los programas íntegros de todos los modelos descritos en el libro. Las figuras del texto o algún párrafo indican con un círculo azul, el enlace correspondiente al código completo.

● Si desea revisar el código completo siga esta referencia.

Enfoque pedagógico

El libro adopta un **enfoque minimalista (y progresivo)**, donde cada modelo sirve como punto de partida para un estudio posterior más profundo o para abordar el siguiente tema.

El lector puede **seguir la secuencia completa** —desde los agentes más simples hasta los sistemas con aprendizaje adaptativo— o bien **estudiar capítulos específicos** de manera independiente.

Gracias a la disponibilidad de abundante material en Internet y a las herramientas de **Inteligencia Artificial Generativa (IAG)**, el texto principal se mantiene conciso, permitiendo al lector **profundizar libremente** en los temas de mayor interés mediante la exploración asistida o autónoma.

Este enfoque hace del libro un recurso ideal tanto para **cursos trimestrales o semestrales**, o para el **estudio independiente de manera autodidacta**.

Reconocimientos

Este libro ha surgido de los cursos de IA, Cibernética y Robótica que he impartido en la UAM-I. Agradezco a los estudiantes de esos cursos y a mis mentores Rolando Lara e Ismael Espinosa, que han contribuido a la discusión de ideas que dieron forma a este material, así como a las tecnologías que hoy nos permiten simular y comprender la inteligencia natural y artificial desde nuevas perspectivas.

El libro se ha pulido y organizado con la ayuda de ChatGpt.

ChatGpt se ha utilizado como un ayudante de investigación para la elaboración de este material.