ESCENARIO de APRENDIZAJE

Búsqueda de soluciones mediante el pensamiento computacional

Nivel educativo: Educación infantil, primaria y secundaria **Edad:** >6

Autores: Michael Steiner y Hermann Morgenbesser, **Future Learning Lab Viena**

OBJETIVOS Y ASPIRACIONES PRINCIPALES

El objetivo del pensamiento computacional es utilizar métodos analíticos y algorítmicos para formular, analizar y resolver problemas. Entre las prácticas conducentes al desarrollo del pensamiento computacional se pueden citar las siguientes: diseño y creación de simulaciones, modelos y objetos computacionales; producción colaborativa de objetos o artefactos relativos a fenómenos naturales y artificiales; e implementación de técnicas informáticas para la solución de problemas, como codificación, programación y robótica.

DESCRIPCIÓN GENERAL

La única y mejor manera de desarrollar la capacidad de razonamiento es enfrentarse constantemente a nuevos problemas que tengamos que solucionar. El pensamiento computacional busca desarrollar las siguientes competencias:

- Capacidad de enfrentarse a la complejidad con confianza
- Capacidad de resistencia antes problemas complicados
- Tolerancia ante la ambigüedad
- Capacidad de resolver preguntas abiertas
- Capacidad de comunicación y de resolución de problemas en conjunto

Todos estos son valores educativos que cada día adquieren mayor relevancia.

Para desarrollar el pensamiento computacional, se pueden plantear al alumnado problemas abiertos, ambiguos, complejos y tan difíciles que solo se puedan resolver en equipo y con un buen sistema de comunicación. Es importante también resaltar que este tipo de pensamiento se puede aprender y enseñar a cualquier edad, desde la infancia más temprana hasta los exámenes de fin de escolarización obligatoria e incluso más allá; y que se puede hacer a través de una gran variedad de disciplinas, desde las ciencias y las matemáticas hasta la lengua y la literatura.

Podemos poner como ejemplo la construcción de un podómetro manual. El docente plantea al alumnado pensar sobre algún problema que le parezca útil resolver. A este respecto, es importante recordar que el alumnado debe decidir sobre ese problema de acuerdo con sus propios intereses. Se trata de que el alumnado piense en algo que le intrigue y formule su historia. En nuestro ejemplo, un alumno cuenta que su hermana dice que anda 10 000 pasos cada día. Su hermana es profesora de educación infantil y va todos los días andando a trabajar. Según ella, así se mantiene en forma. El alumno busca entender qué significa andar 10 000 pasos. Es imposible contar todos los pasos que da cada día. Además, posiblemente no esté todo el día andando; en algún momento se sentará en el trabajo. Este es el tema que se plantea y debate con el docente y el resto de alumnado. Entre todos, deciden construir un podómetro que cuente los pasos que anda la hermana del alumno que ha hecho la propuesta. La tarea se formula como la construcción de un podómetro que podamos colocarnos en la muñeca o en el tobillo y que cuente los pasos que andamos. A cada paso, el podómetro cuenta un pulso y lo muestra en la pantalla. También tiene que incluir la posibilidad de reiniciarse y volver a contar desde cero.

METODOLOGÍA DE ENSEÑANZA Y APRENDIZAJE

La enseñanza por proyectos y el trabajo en equipo son dos metodologías adecuadas para enfrentarse a otros problemas más complejos de pensamiento computacional.

La denominada «informática sin ordenador» (Computer Science Unplugged) busca resolver problemas para lograr un objetivo y trabaja con conceptos informáticos básicos.

Simulaciones informáticas para explorar distintos fenómenos.

Modelos informáticos que se pueden probar, depurar y refinar.

Juegos informáticos o proyectos de construcción de aplicaciones

EVALUACIÓN

El análisis de patrones de pensamiento computacional, por ejemplo, permite visualizar qué competencias de las nueve específicamente relacionadas con el diseño de juegos ha conseguido dominar el alumnado. Dr. Scratch trabaja siete dimensiones de la competencia en pensamiento computacional.

Análisis de objetos o artefactos, escenarios de resolución de incidencias.

FUNCIONES

ALUMNADO: El alumnado trabaja en colaboración (como equipo para la planificación y el desarrollo de soluciones), con cada miembro del grupo asumiendo una función (programador, analista, desarrollador...) para organizar las tareas.

PROFESORADO: El profesorado debe aprender a gestionar un aula en la que los ordenadores sirven tanto como medio principal para demostrar la ejecución como de material de apoyo ocasional. Pueden distribuirse las actividades por las distintas zonas de aprendizaje del aula. Además, como apoyo a la colaboración en grupo, es necesario un proceso de andamiaje.

OTROS: Se pueden utilizar herramientas de videoconferencia para invitar a personas expertas externas como apoyo a diversas actividades.

ENTORNO DE APRENDIZAJE

Es importante la interacción entre el profesorado y el alumnado a lo largo de la experiencia de aprendizaje, mientras el docente da instrucciones y feedback y dirige el proceso. El alumnado trabaja en equipos, lo que le permite intercambiar de forma activa sus opiniones sobre las tareas del proyecto y la división de funciones. Prepara una propuesta de solución con el apoyo del docente. Cada equipo ofrece sus soluciones. Pueden compartir distintas tareas para crear un producto. Por ejemplo, un equipo puede ocuparse de la programación, otro de trabajar con el micro:bit y otro de los materiales. Después, el alumnado muestra sus avances al docente, quien sigue asesorando sobre los siguientes pasos y dando algunas posibles pistas, además de plantear las posibles dificultades o errores. Nuevamente vuelve a haber interacción. Por último, el alumnado presenta su trabajo o los avances que haya logrado y reflexiona sobre cómo lo ha ido desarrollando y cómo ha resultado el trabajo en equipo.

POSIBLES RETOS

Cuando se integra en la educación obligatoria, se plantea la cuestión de qué tipo de evaluación podría explicitar la capacidad de pensamiento computacional y solución de problemas por parte del alumnado en contextos reales.

También hay que desarrollar ciertas conductas entre el alumnado, que le permitan trabajar con otros y hacer frente a la frustración.

Hay que escoger actividades adecuadas a cada edad, pero también hay que responder a los intereses del alumnado y, para no desmotivar a las alumnas, es necesario proponer actividades que encajen también en sus gustos.

Sobre todo, las actividades para el desarrollo del pensamiento computacional necesitan de profesorado capacitado para diseñar y evaluar experiencias en el aula relacionadas con este tipo de pensamiento que se centren en sus conceptos clave.

VIDEO DE ESCENARIO DE APRENDIZAJE

https://www.youtube.com/watch?v=Z7xg1yZGeW0

ACTIVIDADES DE APRENDIZAJE

Las actividades de aprendizaje relacionadas con el pensamiento computacional se construyen en torno a los conceptos clave de este tipo de pensamiento, como la abstracción, el pensamiento algorítmico, la automatización, la descomposición, la depuración y la generalización. La codificación y la programación son dos de los elementos que integran el pensamiento computacional, pues concretizan sus conceptos y, por tanto, pueden convertirse en una herramienta de aprendizaje. No obstante, aún más importante es el proceso que precede a la codificación y programación: el análisis y la descomposición de problemas.

Algunas de las actividades principales en el caso de la construcción del podómetro serían las siguientes:

El docente da instrucciones sobre cómo diseñar prototipos UML/pseudocódigo o cómo desarrollar una solución.

Los equipos desarrollan propuestas de soluciones.

Se hace una presentación general de los programas ejecutables o componentes.

Se reflexiona sobre el proceso y se realizan los cambios necesarios.

También se pueden introducir mejoras. La última fase podría consistir en rehacer o mejorar la solución.

RECURSOS

lenguaje de programación (Python, Scratch, SNAP)
micro:bit
telas viejas, hilo y aguja. Velcro

REFERENCIAS BIBLIOGRÁFICAS DE APOYO

Wing, J. (2006): Computational thinking. Communications of the ACM, 49(3), 33-35.

Computational Thinking Task Force: https://csta.acm.org/Curriculum/sub/CompThinking.html

Computer Science Unplugged: https://csunplugged.org/en/

K-12 Framework: https://k12cs.org/wp-content/uploads/2016/09/Computer-Science-Framework.pdf

Coding with microbits: https://padlet.com/eis/dlplwien

Co-funded by the Erasmus+ Programme of the European Union

The learning scenario is created by the DesignFILS project (http://designfils.eba.gov.tr), funded by EU's Erasmus+ KA2 (grant agreement 2019-1-TR01-KA201-076567). The contents of the publication are the sole responsibility of the authors, and the EC or Turkish National Agency cannot be held responsible for any use which may be made of the information contained therein.