

Cómo incorporar el pensamiento computacional en el aprendizaje basado en proyectos (ABP)

2023-09-27

Cómo incorporar el pensamiento computacional en el aprendizaje basado en proyectos (ABP)

Este documento muestra cómo incorporar el pensamiento computacional y la programación de computadores en un ambiente de aprendizaje basado en proyectos, mediante cuatro etapas: a) presentar a los estudiantes el desafío de diseño, la pregunta guía y los objetivos de aprendizaje; b) apoyar la comprensión del pensamiento computacional; c) dar un tiempo de trabajo para diseñar; y, d) reflexionar para activar procesos metacognitivos.

Este documento muestra cómo incorporar el pensamiento computacional y la programación de computadores en un ambiente de aprendizaje basado en proyectos, mediante cuatro etapas: a) presentar a los estudiantes el desafío de diseño, la pregunta guía y los objetivos de aprendizaje; b) apoyar la comprensión del pensamiento computacional; c) dar un tiempo de trabajo para diseñar; y, d) reflexionar para activar procesos metacognitivos.

CÓMO INCORPORAR EL PENSAMIENTO COMPUTACIONAL EN EL APRENDIZAJE BASADO EN PROYECTOS (ABP)

Por Jorge Valenzuela

Al igual que muchos de mis alumnos, mi hija, Anisa, está muy familiarizada con el uso de circuitos electrónicos para crear inventos geniales. Así que cuando le dije a Anisa y a sus compañeros de clase que íbamos a incorporar el pensamiento computacional en una experiencia de aprendizaje basada en proyectos (ABP), llamada “Proyecto del Pensador Computacional”, me sorprendió su pregunta:

"¿Acaso la programación de computadores no es *solo* es utilizada por [programadores informáticos](#) altamente inteligentes?", preguntó con aprensión.

Le expliqué que la programación es, de hecho, una habilidad utilizada por los programadores informáticos. Pero, el pensamiento computacional es un *proceso* que pueden entender tanto las máquinas como los humanos, y aprenderlo la convertiría en una mejor programadora de computadores.

Anisa no está sola. Muchos de sus pares y también educadores, se sienten intimidados por los términos *programación* y *pensamiento computacional*. Sin embargo, una vez que el concepto se entiende como un enfoque sistemático para [resolver problemas](#), se convierte en algo menos desalentador.

Entonces, si eres como Anisa y sus compañeros de clase, no te preocupes por la programación. En su lugar, enfóquese en aprender y aplicar el pensamiento computacional en el contexto familiar de un desafío de diseño STEAM que utiliza circuitos electrónicos para hacer inventos geniales. Aquí le mostramos cómo puede agregar el pensamiento computacional y la programación a una unidad de ABP y agregarlos a la caja de herramientas creativas de sus estudiantes.

ESTÁNDARES ISTE PARA ESTUDIANTES

Para proporcionar contexto e indicadores claros para el pensamiento computacional, yo empiezo presentándole a mi clase los [Estándares ISTE para estudiantes](#) y explicando claramente cómo el pensamiento computacional es una habilidad para resolver problemas.

Anisa asiste a un sistema escolar muy experto en el uso de tecnología en educación ([CCPS](#)) y, aunque ya estaba familiarizada con los conceptos de ser una diseñadora innovadora y ciudadana digital, nunca había visto una [infografía](#) hecha por educadores, ¡especialmente, una hecha para ella y sus compañeros de clase!

Para garantizar que nuestro “Proyecto del Pensador Computacional” sea el vehículo para el aprendizaje previsto del pensamiento computacional y la programación, facilité el proceso en los siguientes cuatro pasos.**

PASO 1: PRESENTE A LOS ESTUDIANTES EL DESAFÍO DE DISEÑO, LA PREGUNTA GUÍA Y LOS OBJETIVOS DE APRENDIZAJE**

El objetivo de Anisa es estudiar botánica, y en su tiempo libre le gusta ayudar a sus compañeros con sus tareas escolares. Incluso, perfeccionó sus habilidades de enseñanza / tutoría al trabajar como asistente de enseñanza voluntaria en el Centro de Innovación de Matemáticas y Ciencias. Por lo tanto, enmarqué este proyecto en torno a su interés en ayudar a sus compañeros a comprender sus asignaturas e incorporando un desafío de diseño STEAM utilizando [littleBits](#) como uno de los productos.

La pregunta guía para el proyecto fue: "¿Cómo podemos nosotros, como pensadores computacionales, diseñar un **juego para niños** que enseñe a los compañeros más jóvenes habilidades básicas de programación y sobre el uso de **circuitos electrónicos**?"

Siendo algo sin novedad en el proceso, mis alumnos escribieron en cursiva nuevo vocabulario (temas) en la pregunta guía o conductora y subrayaron los elementos que ya conocían. Mi objetivo era hacer que se enfocaran en el nuevo aprendizaje y cómo se conecta con cualquier conocimiento previo para el diseño del juego.

Le di a los estudiantes una breve lista de ideas de diseño de juegos para elegir, y Anisa se decidió por [Hot Potato](#).

Algunos de los objetivos de aprendizaje que quería que los estudiantes alcanzaran fueron:

- Puedo descomponer cada paso del proceso de programación en detalles minuciosos para poder explicarlo a los demás.
- Puedo reconocer patrones (similitudes o diferencias comunes) que me ayudarán a hacer predicciones sobre qué “bits” (entrada, salida, potencia, etc.) son apropiados para el diseño de mi juego.
- Puedo abstraer cualquier información innecesaria mientras construyo mi programa.
- Puedo desarrollar un algoritmo paso a paso para una tarea personal de mi elección (pintarme las uñas, pasear a mi perro, etc.).
- Puedo desarrollar, paso a paso, algoritmos para el programa del juego.
- Puedo definir y aplicar bucles (estructura repetitiva) al programar el juego.
- Puedo usar variables para almacenar datos que puedan referenciarse en el programa del juego.
- Puedo definir y aplicar lógica condicional (estructura condicional) en el programa del juego.
- Puedo aplicar el paso de remezcla en el Ciclo de invención “littleBits” para crear un nuevo código para mi juego.

Nota para maestr@s: Los objetivos de aprendizaje utilizados para este trabajo fueron la reescritura de algunos indicadores enumerados para el capítulo de pensamiento computacional en los Estándares ISTE para estudiantes, la [lección “Hot Potato of Doom”](#) y los objetivos de aprendizaje de pensamiento computacional para estudiantes de [Bloomfield Hills School](#).

Los objetivos de aprendizaje deben estar escritos en un lenguaje amigable para los estudiantes y son una excelente estrategia para ayudarles a enfocar su aprendizaje, desarrollar vocabulario y solicitar retroalimentación específica. Realmente recomiendo conocer la [rúbrica](#) para diseñar objetivos de aprendizaje, para comenzar.

PASO 2: APOYE LA COMPRENSIÓN DEL PENSAMIENTO COMPUTACIONAL

Para comenzar a abordar los objetivos de aprendizaje y el andamiaje del aprendizaje del pensamiento computacional, le presenté a los estudiantes lo esencial con un video pegajoso [de JULES](#). El video apoyó su comprensión porque proporcionó una imagen tangible tanto de los conceptos clave ([los cuatro elementos del pensamiento computacional](#): Descomposición, reconocimiento de patrones, abstracción & algoritmos) como de su aplicación con y sin computadoras, pero dentro de contextos con los que ya estaban familiarizados.

Para apoyar aún más su comprensión del pensamiento computacional, presenté a Anisa y a los demás el trabajo del increíble [Dr. Shuchi Grover](#). Ahora no solo tienen otro punto de referencia al que pueden volver a consultar mientras trabajan en el desafío de diseño STEAM, sino que también tienen otro modelo femenino a seguir, como [Oprah](#) , [Ellen](#), [Ayah Bdeir](#), [Kim Lane](#), [Emma Gonzales](#) y [Tarana Burke](#) (entre muchas otras mujeres inspiradoras), ¡y yo también!

Creo que los estudiantes están más inclinados a involucrarse en la ciencia de la computación y actividades STEAM cuando ven y aprenden de personas involucradas en trabajos altamente colaborativos e innovadores que se parecen a ellos. Inspirar a los estudiantes con excelentes modelos a seguir tampoco debería estar limitado por el mundo de la educación tecnológica “Edtech” o la ciencia de la computación “CS”. Los defensores de los movimientos que dan voz a las poblaciones sub representadas

(niñas, afroamericanos, hispanos, estudiantes LGBTQ y otros) a menudo cuentan historias muy convincentes con las que muchos de nuestros jóvenes pueden inspirarse y relacionarse.

PASO 3: MOMENTO DE DISEÑO (TIEMPO DE TRABAJO)

Para transferir sus nuevas habilidades de pensamiento computacional y también ayudar a otros, Anisa le dio su propio toque al popular juego para niños [Hot Potato](#), al diseñar un modelo que ayudaría a los niños más pequeños a comprender la importancia de los circuitos electrónicos, la programación y las tecnologías que se encuentran en la vida cotidiana mientras disfrutan del juego familiar.

Usando el [Kit de programación “LittleBits”](#), Anisa construyó la base de su juego Hot Potato basado en instrucciones retocándolo con un diseño original. Al no cambiar la premisa de Hot Potato (sostener un objeto que se pasa en círculo y cuando la música / sonido se detiene, se elimina la persona que lo sostiene), le permitió reconocer patrones, funciones y propósitos tanto del hardware como del programa.

Al dominar la nueva tecnología y varios principios de programación (aplicar [bucles](#), usar [variables](#) para almacenar datos para ser referenciados y lógica condicional), ella resolvió su problema (juego) y la solución paso a paso (algoritmo) que codificó usando Google [“Blockly-based code”](#). Ella usó su código para controlar el temporizador, el mensaje y la visualización de imágenes en la matriz LED de su diseño de juego [“Hot Potato”](#). Antes de decidirse por una solución automatizada final, probó y rehízo su diseño algorítmico varias veces hasta que fue de su agrado.

Nota para maestr@s: durante el tiempo de trabajo, es esencial que facilite el aprendizaje. Al trabajar con grupos de estudiantes, es importante estar bien versado en el uso del Kit de códigos (o cualquier tecnología que esté utilizando en las clases / proyectos) y su aplicación. Utilice la aplicación para ayudarse a realizar mini clases y andamiajes de los conceptos de codificación que sus alumnos más necesitan (ya sea individualmente o en grupos).

No olvide involucrar a los estudiantes en un [proceso de diseño](#). En este proyecto, Anisa utilizó el [ciclo de invención littleBits](#) y la [lista](#) de verificación del registro de invención que se encuentra en la página 16 del kit de código [“Kit invention](#)

[log](#)” para evaluar su trabajo. También recomiendo tener la [lista de verificación de depuración](#) y la [tabla de comentarios](#). Además, use [los protocolos de retroalimentación](#) y los objetivos de aprendizaje para ayudar a los estudiantes a solicitar y recibir retroalimentación con el fin de mejorar su trabajo.

PASO 4: REFLEXIÓN PARA LA METACOGNICIÓN

Con Anisa y mis alumnos yo siempre hago hincapié sobre la importancia de reflexionar para la metacognición, brindándoles oportunidades para reflexionar sobre sus pensamientos y desarrollar una mejor comprensión de cómo ellos aprenden mejor para que puedan convertirse en aprendices de por vida. Con este propósito, incorporé la reflexión a lo largo de los diversos pasos del desafío de diseño (tanto en el diario de registro de la invención como en nuestras sesiones de preguntas).

Una de las respuestas de Anisa a las preguntas de reflexión fue sobre cómo los Estándares ISTE apoyaron su aprendizaje del pensamiento computacional. Ella escribió en su diario: "Encuentro que los Estándares ISTE proporcionan buenas pautas para ayudarme a aprender y aplicar los conceptos / prácticas del pensamiento computacional y otros estándares que me permiten saber cómo combinar mi uso de la tecnología (de manera creativa y responsable) con lo académico". También me gusta saber que se puede confiar en los estándares porque fueron escritos por educadores que saben lo que los estudiantes más necesitan. ¡Más escuelas deberían implementar estos estándares para sus estudiantes!"

Nota para maestr@s: El trabajo descrito en esta publicación está muy influenciado por las prácticas (Inteligencia Digital, reflexión, etc.) que se encuentran en el blog “[Gold Standard PBL](#)” del [Instituto Buck para la Educación](#) y en el [Aprendizaje Basado en Proyectos de Alta Calidad](#) . Esto se debe principalmente a mi trabajo con BIE “Buck Institute for Education”.

Es importante tener en cuenta que Anisa enseñando sus nuevas habilidades de pensamiento computacional y de programación a sus compañeros más jóvenes, y presentando públicamente su trabajo también fueron productos del Proyecto del Pensador Computacional.

Para los maestros de STEM / STEAM que desean mejorar su uso del aprendizaje basado en proyectos “ABP” recomiendo los trabajos de algunos expertos en ABP muy notables:

- [Aaron Eisberg](#)
- [Andrew K. Miller](#)
- [BIE blog](#)
- [I Do School](#) ([Charity Moran](#))
- [Michael Niehoff](#)
- [Project ARC](#)
- [Remix Education](#) (Shayla Adams-Stafford)
- [Suzie Boss](#)
- [TweenTeacher](#) (Heather Wolpert-Gawron)

RECURSOS A LOS QUE PUEDE RECURRIR PARA OBTENER AYUDA:

- [Code.org](#) ha desarrollado una excelente [lección para principiantes en](#) pensamiento computacional con correlaciones a ELA, matemáticas y los estándares [ISTE para estudiantes](#).
- El [Marco de Ciencias de la Computación K-12](#) ofrece una amplia visión general del [pensamiento computacional](#), junto con recursos y una explicación en profundidad de las correlaciones entre la informática, la ciencia, la ingeniería y las prácticas matemáticas.

- [BBC Bitesize](#) alberga una variedad de recursos de enseñanza en su sitio web para [presentar el pensamiento computacional](#) a los estudiantes.
- [*Pensamiento computacional {y codificación} para cada estudiante: la guía de inicio del maestro de*](#) Jane Krauss y Kiki Prottzman es un gran recurso.
- littleBits : [Code Kit Educator Resources](#) ofrece lecciones, diapositivas de presentación, videos, la aplicación Code Kit y folletos para maestros.
- Consulte los [recursos](#) desarrollados por ISTE y una [guía práctica](#) para educadores que enseñan pensamiento computacional.

CRÉDITOS:

Traducción al español del artículo “[Embed computational thinking into PBL](#)”, escrito por Jorge Valenzuela y publicado por ISTE en la [Revista Empowered Learner](#). **Anisa Valenzuela** es una estudiante de noveno grado en la escuela de Manchester en Chesterfield, Virginia. Anisa ya obtuvo su Certificado de Junior Master Gardeners y le gustaría asistir a la universidad para continuar aprendiendo botánica. Quiere trabajar como asistente de docente durante un curso de verano en el Centro de Innovación de matemáticas y ciencias en Richmond, Virginia, y a menudo es tutora de sus compañeros en ciencia, geometría y diseños STEAM. **Jorge Valenzuela** es un formador de maestros y un asistente de enseñanza de posgrado en la Universidad Old Dominion. También es el formador principal de Lifelong Learning Defined, Inc., una facultad nacional del Instituto Buck para la Educación, es un formador nacional de efectividad docente en la Asociación Internacional de Educadores de Tecnología e Ingeniería (ITEEA) y hace parte del programa de educadores líderes para littleBits. Puede conectarse con Jorge en Twitter [@JorgeDoesPBL](#) para continuar la conversación.

Esta traducción se realiza en el marco de la Alianza establecida entre ISTE y la Universidad Icesi para realizar actividades internacionales conjuntas que promuevan la constante actualización de la comunidad académica hispanoamericana. La presente

traducción no es obra de ISTE y no deberá considerarse traducción oficial de dicha organización. ISTE no responderá por el contenido ni por posibles errores de la traducción.

Publicación de este documento en EDUTEKA: Febrero 9 de 2021.

Última actualización de este documento: Septiembre 27 de 2023.