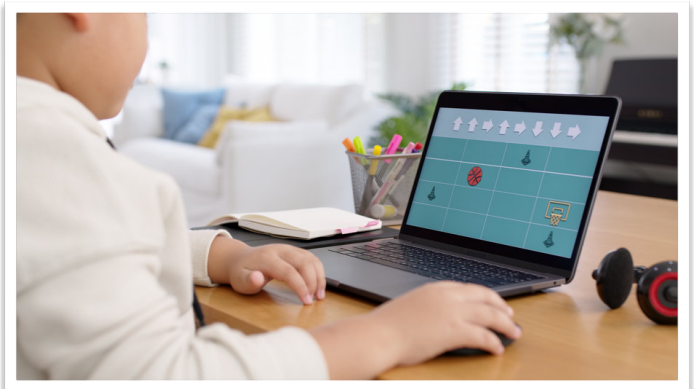
PROGRESIÓN DE APRENDIZAJES EN PENSAMIENTO COMPUTACIONAL

La importancia del pensamiento computacional

Existe un consenso creciente en el mundo en relación a la importancia del Pensamiento Computacional y la necesidad de enseñarlo en todos los niveles educativos para tener una sociedad preparada para interactuar en una sociedad tecnológica y afrontar los retos de la globalización (Barr & Stephenson, 2011; Lee et al., 2011; National Research Council, 2011; The Royal Society, 2012). Como resultado, diferentes países han incorporación prácticas computacionales, científicas y de ingeniería a los currículos de educación a nivel primaria y secundaria (National Research Council, 2011; NGSS Lead States, 2013).

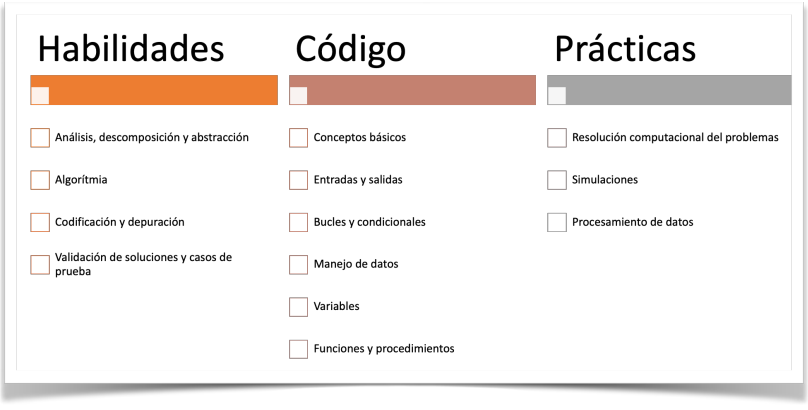


En los currículos de pregrado también ha avanzado la integración de las prácticas de pensamiento computacional mediante la creación de cursos específicos de programación para disciplinas específicas, así como el uso de modelos y simulaciones computacionales con el fin de representar fenómenos complejos que se abordan en diferentes cursos (e.g., Vieira et al., 2018).

En Colombia, como un primer paso, la integración de las prácticas de pensamiento computacional al currículo de primaria y secundaria, en particular en los cursos de las áreas de STEM, pueden aportar al proceso de aprendizaje, teniendo en cuenta que comparten similitudes en sus fundamentos, principios y métodos de conocimiento (Basu at al, 2016).

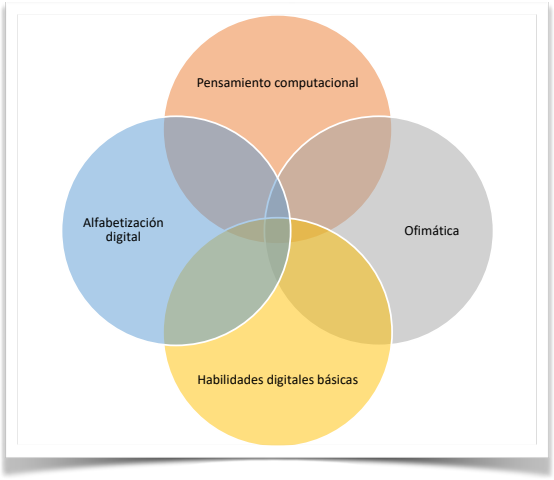
Qué es pensamiento computacional

El pensamiento computacional (PC) ha sido definido como el proceso mental de formular problemas y sus soluciones para representarlas de tal manera que puedan ser llevadas a cabo por un agente de procesamiento de información (Wings 2006). La computación facilita la solución de problemas a través de estrategias tales como el procesamiento de grandes cantidades de datos, hacer



tareas repetitivas de manera eficiente y la representación de fenómenos complejos a través de simulaciones.

Relación del pensamiento computacional con ofimática y alfabetización digital



Dada la corta historia de la educación para la utilización de tecnologías digitales es frecuente que exista un nivel importante de confusión entre varios conceptos como robótica, programación, ofimática, alfabetización digital y pensamiento computacional. Como toda taxonomía, está basada en la selección de características para clasificar y organizar, así como algunas definiciones que son producto de un desarrollo histórico de los conceptos. A continuación se busca presentar una propuesta de organización basada en la definición dada por Wing (2006) para el pensamiento computacional, en la definición dada por OCDE (2019) igualmente sobre el pensamiento computacional en el marco de la prueba PISA y en Department for Education (2019) del Reino Unido sobre lo que significa alfabetización digital.

Esta revisión muestra que un conjunto de aprendizajes básicos no se encuentran ni en pensamiento computacional, ni en habilidades digitales, se asumen como un requisito previo:

- Encender y apagar dispositivos digitales.
- Usar los controles y periféricos de los dispositivos (teclado, pantalla, impresora, ratón ...).
- Gestión básica de información en dispositivos digitales
- Usar un navegador para búsquedas sencillas.
- Usar de forma segura redes públicas.
- Usar de forma segura mecanismos de autenticación (usuarios, claves ...).

Algunas de estas habilidades las desarrollan los estudiantes con el simple contacto en su vida cotidiana con dispositivos digitales cuando el medio se los permite, pero otras habilidades relacionados con el uso seguro requieren de enseñanza explícita en la escuela.

Cuando se aborda el constructo de alfabetización digital se encuentran elementos como:

- Adelantar trámites y compras de forma segura usando Internet.
- Realizar búsquedas de información de calidad en Internet con criterio.
- Comunicar y colaborar (incluye lo que usualmente se conoce como ofimática):
    - Edición de textos y presentaciones
    - Uso del correo electrónico
    - Participación en reuniones y cursos virtuales

- Resolver problemas utilizando la computación (un componente del pensamiento computacional)

En este último elemento se encuentra una de las prácticas del pensamiento computacional referida en la sección anterior.

La robótica, en particular, hace referencia a una tecnología en general costosa sobre la cual se pueden desarrollar solo parcialmente algunos de los aprendizajes antes indicados. Es importante resaltar que para muchos aprendizajes las actividades de computación desconectada son más efectivas sin requerir altas inversiones.

Exploración de algunos programas internacionales que integran el Pensamiento Computacional en el currículo

Para el trabajo que sigue se usaron varios referentes entre los que se encuentran documentos del Reino Unido, de la OCDE y de Australia así como Vieira et al (2019).

Estrategias didácticas en pensamiento computacional

La enseñanza del pensamiento computacional, como sucede en las demás disciplinas, requiere de estrategias y didácticas propias, con las cuales los aprendizajes se logran mejor y en menor tiempo. La utilización de proyectos en el marco de ABP (Aprendizaje basado en proyectos) debe reservarse para etapas avanzadas de aprendizaje donde lo que se busca es que el estudiante aplique conocimiento ya adquirido en la solución de problemas con un nivel de dificultad alto, pero al alcance de lo que ya saben.

Actividades desconectadas

Si bien buena parte de las actividades de aprendizaje del pensamiento computacional son mediadas por dispositivos digitales, tales como computadores o tabletas, existe un creciente uso de lo que se han denominado *actividades de computación desconectada*.

Adelantar actividades de computación sin dispositivos digitales tiene al menos cuatro ventajas:

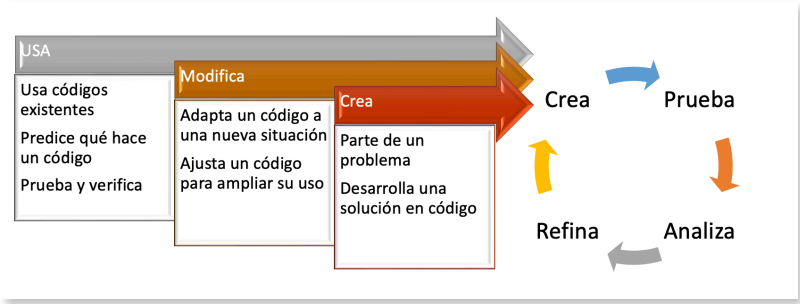
- Facilita el aprendizaje de habilidades y conceptos de base en pensamiento computacional sin la sobrecarga y distracción que representa en las primeras etapas un dispositivo tecnológico.
- Trabajar el pensamiento computacional desde la primera infancia, donde el uso de dispositivos con pantallas está vivamente desaconsejado.
- Desarrollar aprendizajes sobre los algoritmos, la detección de patrones y regularidades o la abstracción.
- Aun en lugares donde no se dispone de tecnologías digitales apropiadas es posible introducir a los estudiantes en el pensamiento computacional.

Sin embargo, un error que se debe evitar es interpretar las actividades de computación desconectada como la alternativa cuando no hay computadores. Aun en instituciones que disponen de tecnologías digitales apropiadas, las actividades desconectadas son fundamentales.

Usa - modifica - crea

Una vez se ha trabajado con actividades de computación desconectada y es el momento de pasar a la implementación, se recomienda una estrategia llamada ***usa-modifica-crea***.

Esta estrategia sugiere que para evitar sobrecargas cognitivas, primero hay que brindarle a los estudiantes la posibilidad de usar las herramientas y representaciones computacionales para que se familiaricen con éstas. Por ejemplo, suministrar una solución computacional a un problema en particular. En esta etapa los estudiantes exploran el código suministrado, tratan de anticipar lo que hará y posteriormente lo prueban y verifican sus predicciones.



En una segunda etapa, los estudiantes deben modificar estas representaciones computacionales para resolver un problema existente o para ampliar una solución, de tal manera que comiencen a desarrollar una comprensión sobre cómo están construidas y cómo pueden ser adaptadas a otros contextos o problemas.

Finalmente, una vez los estudiantes hayan desarrollado los conocimientos necesarios para hacerlo, es posible pedirles que creen nuevos artefactos y soluciones desde cero, si bien continuarán utilizando código ya escrito, pero en este caso le corresponde al estudiante seleccionar el código apropiado.

Un error común en el uso de esta estrategia es asumir que cada actividad debe pasar por todas las etapas. Es importante reconocer que al comienzo, durante muchas sesiones, probablemente las actividades se centren en el USA y paulatinamente se irá pasando a Modifica. El CREA implica que el estudiante ya tienen suficientes conocimientos para abordar la etapa de CREA. El uso del ABP implica un componente importante del paso CREA.

El pensamiento computacional y las demás áreas

Habilidades y comprensiones propias de las matemáticas y el lenguaje son sustento central para el desarrollo del pensamiento computacional y, en consecuencia, deben asegurarse.

El uso del pensamiento computacional en áreas como ciencias naturales y sociales, por ejemplo, puede ayudar al aprendizaje en estas áreas desde las prácticas de manejo de datos y de simulación. El uso de simulaciones computacionales parametrizables en ciencias naturales son una herramienta valiosa en la aproximación de fenómenos que no pueden ser abordados de otra forma. En ciencias sociales, el uso de sistemas geográficos y de datos relativos a poblaciones, permiten comprender diferentes aspectos de la sociedad y su desarrollo.

Tomado de https://phet.colorado.edu

Una estrategia que se debe evitar es pretender enseñar habilidades y comprensiones básicas de pensamiento computacional desde otras áreas, dado que esta acción puede distraer de los aprendizajes de la respectiva área, logrando aprendizajes fragmentados, poco estructurados y poco desarrollados en pensamiento computacional. Los espacios propios para el pensamiento computacional son irremplazables. Sin embargo, utilizar aprendizaje de pensamiento computacional en otras áreas, ayuda a aprendizajes en estas áreas y le da la posibilidad al estudiante de aplicar, conectar y profundizar aprendizajes ya adquiridos en pensamiento computacional.

Propuesta de progresión de aprendizajes

A continuación se presentan, grado a grado, los aprendizajes sugeridos en pensamiento computacional. Las tablas se organizaron de acuerdo a las siguientes categorías, sobre las cuales se hace una breve descripción:

Componente conceptual

Gran idea

Representa una o pocas ideas que dan cuenta de lo que se quiere lograr en términos de formulación de los aprendizajes que debe lograr el estudiante.

Conceptos asociados

Lista los conceptos en los cuales se enfocan los aprendizajes en el grado respectivo, si bien pueden estar asociados otros conceptos adquiridos previamente, los cuales no se mencionan para simplificar la presentación.

Habilidades

Corresponden a un conjunto de habilidades o destrezas que desarrollan los estudiantes en cada nivel. Estas habilidades son agrupadas en: Interpretar y Diseñar Algoritmos, y Programar, Depurar y Utilizar Estructuras de Datos.

Interpretar y diseñar algoritmos

Corresponde a interpretar, diseñar y evaluar algoritmos en diferentes representaciones para resolver problemas. En general este tipo de aprendizajes se logran mejor con actividades desconectadas.

Programar, depurar y utilizar estructuras de datos

Utilizar diferentes lenguajes de programación y sus diferentes tipos de abstracción como funciones, estructuras de control, operadores y estructuras de datos. Estos aprendizajes en general se logran con actividades conectadas.

Prácticas

Modelar y simular

Abstraer regularidades de un fenómeno con el fin de simular aspectos de su funcionamiento. Se pueden iniciar estos aprendizajes con actividades desconectadas, para luego proceder a utilizar dispositivos de cómputo.

Resolver problemas a través de la computación

Utilizar herramientas computacionales para resolver problemas por medio de unidades de procesamiento.

Procesar y analizar información

Utilizar herramientas computacionales para procesar y analizar información.

Aprendizajes grado a grado

Preescolar

Grandes ideas	• Una tarea se puede realizar a través de una secuencia de pasos que se pueden describir
Conceptos asociados	• Secuencias de pasos e instrucciones • Regularidades y patrones
Interpretar y diseñar algoritmos	• Define un problema sencillo • Sigue una secuencia de pasos necesarios para resolver problemas simples • Genera instrucciones orales para llevar a cabo tareas de orientación espacial (guiar a sus compañeras y compañeros hacia un objetivo usando indicaciones como izquierda, derecha, adelante, atrás …) • Reconoce las tareas diarias como secuencias de pasos que se repiten (lavarse las manos, atarse los cordones, empacar su mochila) • Identifica patrones en secuencias sencillas • Reconoce similitudes entre objetos y figuras según sus características • Realiza afirmaciones a partir de lo que observa: hay más objetos rojos que verdes, el cuadrado es más grande que el círculo

Primer grado

Grandes ideas	• Una tarea se puede realizar a través de una secuencia de pasos que se pueden describir identificando partes que se repiten o se parecen • Los pasos se pueden expresar usando símbolos o dibujos
Conceptos asociados	• Secuencias de pasos e instrucciones • Regularidades y patrones • Descomposición
Interpretar y diseñar algoritmos	• Sigue secuencias de instrucciones previamente diseñadas para resolver un problema nuevo • Define instrucciones orales y gráficas para llevar a cabo tareas simples (salir de un laberinto sencillo, encontrar un tesoro, lograr una secuencia de movimientos) • Divide tareas en una serie de tareas más simples (proponer divisiones para sus tareas cotidianas, proponer instrucciones sencillas para preparar un alimento) • Elige el orden correcto en el que se deben ejecutar las instrucciones de una secuencia • Repite una secuencia de instrucciones un número dado de veces • Reconoce patrones en objetos e información usando características comunes • Responde preguntas sobre los grupos de objetos
Programar, depurar y utilizar estructuras de datos	• Da instrucciones sencillas a un elemento para que se desplace (mover una animación prediseñada en un lenguaje por bloques con las flechas adelante, atrás) • Logra predecir el comportamiento del elemento a partir de una corta secuencia de instrucciones. • Reconoce cuando las instrucciones no corresponden a las acciones

Segundo grado

Grandes ideas	• Una tarea descrita por instrucciones o pasos puede tener errores que deben detectarse y corregirse • Los pasos se pueden expresar usando símbolos o dibujos
Conceptos asociados	• Patrones • Entorno de programación • Programación por bloques • Propiedades y características de un objeto
Interpretar y diseñar algoritmos	• Desarrolla soluciones a problemas uniendo partes pequeñas (completar una tarea extensa por etapas) • Repite patrones de instrucciones para apoyar la solución de problemas nuevos • Reconoce errores sencillos en una serie de instrucciones (identifica pasos equivocados para ir de un lugar a otro, al ordenar objetos, o escribir palabras) • Elimina y corrige errores dentro de un conjunto de instrucciones
Programar, depurar y utilizar estructuras de datos	• Reconoce el entorno de programación del lenguaje de programación por bloques • Crea un objeto y le asigna propiedades durante una actividad de diseño de juegos (elegir un personaje en un lenguaje de programación por bloques, cambiar su apariencia, tamaño, color) • Realiza modificaciones sencillas a las instrucciones programadas (cambiar textos, selección de valores) • Reconoce que se pueden crear representaciones gráficas de los objetos y sus cantidades (utiliza tablas de conteo y pictogramas)

Tercer grado

Grandes ideas	• Una tarea se puede expresar como un conjunto de pasos descritos en un algoritmo utilizando un lenguaje gráfico.
Conceptos asociados	• Algoritmo • Lenguaje de programación • Programa • Información
Interpretar y diseñar algoritmos	• Reconoce y utiliza el concepto de algoritmo para nombrar las secuencias de instrucciones que llevan a resolver problemas. • Identifica conjuntos de instrucciones que se repiten dentro de un algoritmo. • Sigue y crea algoritmos para resolver un problema • Identifica errores en sus algoritmos y los de sus compañeros • Organiza estrategias para corregir errores dentro de sus instrucciones
Programar, depurar y utilizar estructuras de datos	• Entiende los lenguajes de programación como un lenguaje formal designado para comunicar instrucciones a un agente de computación • Predice el comportamiento de un programa sencillo • Reconoce un programa como una secuencia de comandos para realizar una tarea que tiene un inicio, eventos, metas y resultados esperados • Usa y modifica un programa diseñado previamente • Reutiliza partes de programas que fueron creados previamente para la solución de un problema nuevo • Reconoce diferentes tipos de información (texto, imágenes, números) y cómo un objeto se puede describir por sus características

Cuarto grado

Grandes ideas	• Cuando en una tarea se detectan pasos que se repiten se pueden describir como bucles para simplificar la presentación e interpretación del algoritmo. • Existen formas de programar los algoritmos en un dispositivo usando un lenguaje de bloques
Conceptos asociados	• Bucle o ciclo • Algoritmo • Descomposición • Abstracción • Variables
Interpretar y diseñar algoritmos	• Reconoce y utiliza el concepto de bucle o ciclo para describir secuencias repetidas de instrucciones • Compara diferentes procesos y extrae sus características comunes • Descompone problemas en sub-problemas más pequeños y manejables para facilitar su solución • Extrae lo esencial de un problema eliminando toda la complejidad innecesaria para su comprensión • Identifica que algunas soluciones se pueden generalizar para dar solución a diferentes problemas • Utiliza tablas de 1 y 2 entradas para guardar información • Crea representaciones gráficas de los datos recolectados (pictogramas, conjuntos, imágenes)
Programar, depurar y utilizar estructuras de datos	• Reconoce que los algoritmos son planes que se pueden implementar en un lenguaje de programación para dar soluciones a problemas • Identifica y describe secuencias repetidas en datos o código • Modifica un código para reemplazar instrucciones repetidas por un bucle • Identifica y corrige errores en un programa que incluya secuencias y bucles. • Utiliza variables sencillas para guardar información
Resolver problemas a través de la programación	• Crea, evalúa y corrige un programa en un lenguaje de bloques utilizando secuencias de instrucciones

Quinto grado

Grandes ideas	• Cuando la tarea sube en complejidad, se puede resolver con varios algoritmos, cada uno a cargo de una parte de la tarea, los cuales se pueden programar en un lenguaje • Algunas tareas requieren toma de decisiones para lo cual se condiciona la ejecución de ciertos pasos a situaciones determinadas utilizando elementos particulares de un lenguaje de programación • Los algoritmos se pueden representar de forma gráfica utilizando por ejemplo diagramas de flujo
Conceptos asociados	• Depuración • Descomposición • Proceso de diseño • Tipos de bucles • Variables lógicas • Tipos de datos • Lenguaje de programación • Diagrama de flujo
Interpretar y diseñar algoritmos	• Usar diagramas para representar un algoritmo • Descompone los problemas en sub-problemas más pequeños y manejables para facilitar su solución • Identifica las decisiones que interrumpen un bucle a través de ejemplos (saltar tres veces seguidas, en comparación a saltar hasta que se escuche una palabra clave) • Compara múltiples algoritmos para la misma tarea y determina cuál es el más apropiado. • Identifica y describe patrones en visualizaciones de datos, como cuadros o gráficos, para responder preguntas
Programar, depurar y utilizar estructuras de datos	• Anticipa los resultados de un programa sencillo de desarrollo propio en algunas situaciones • Describe los pasos y las decisiones que se tomaron durante el proceso de diseño y desarrollo de un programa que incluye bucles (por qué se eligieron ciertas instrucciones o un tipo de bucle, cómo se corrigieron los errores) • Depura programas sencillos • Reconoce y agrupa diferentes datos según su tipo (datos numéricos, de texto, listas)
Resolver problemas a través de la programación	• Identifica un problema que requiere el uso de bucles para su solución (calcular la potencia de un número, crear una animación, componer una melodía) • Diseña y refina un programa para dar solución a un problema

Sexto grado

Grandes ideas	• Existen soluciones algorítmicas sencillas para retos importantes como por ejemplo recorrer un laberinto u ordenar unos datos • La programación en bloques permite introducir condicionales y lazos para controlar la ejecución de instrucciones • El uso de computadores permite realizar operaciones aritméticas y estadísticas a gran velocidad
Conceptos asociados	• Diagramas de flujo • Seudocódigo • Condicionales • Variables • Estructuras de control
Interpretar y diseñar algoritmos	• Reconoce el concepto de condicionales para definir las decisiones que se toman durante la ejecución de los algoritmos • Usa diagramas para representar algoritmos que incluyen condicionales y bucles • Explica el flujo de los algoritmos y cómo cambian las salidas a partir de las decisiones que se toman
Programar, depurar y utilizar estructuras de datos	• Comprende que diferentes lenguajes de programación permiten resolver diferentes problemas, y que estas soluciones pueden ser generalizadas a problemas similares • Anticipa los resultados de un programa simples hechos por otros • Detecta y corrige errores en programas simples hechos por otros • Reconoce y utiliza el concepto de variable para definir el espacio en memoria donde se guardan datos dentro de un programa • Crea variables que representan diferentes tipos de datos y hace operaciones sobre sus valores • Utiliza condicionales para comparar una variable con un valor
Resolver problemas a través de la programación	• Crea un juego o animación interactiva que incluye conceptos de programación básica como estructuras de control, variables, bucles
Procesar y analizar información	• Define una pregunta que se pueda responder utilizando datos (¿cuál es la edad promedio del curso?, ¿cuál es la estatura promedio de mis compañeras y compañeros?, ¿cómo varía la temperatura del salón durante el día?) • Define el proceso de toma de datos y su almacenamiento en medios digitales • Utiliza herramientas computacionales para responder a la pregunta planteada

Séptimo grado

Grandes ideas	• Para manejar la información en un computador se usan variables que pueden ser usadas y cambiadas por el programa desarrollado • Partes de una solución que se repiten en una misma solución o en otras soluciones se pueden definir como funciones para facilitar el desarrollo de nuevas soluciones • La programación permite utilizar condicionales y lazos para controlar la ejecución de instrucciones a partir de los valores que tomen las variables
Conceptos asociados	• Funciones • Entradas y salidas • Variables • Casos de prueba • Estructuras de control • Operadores de comparación (mayor que, menor que, igual a)
Interpretar y diseñar algoritmos	• Resuelve problemas descomponiéndolos en partes más pequeñas • Define funciones que den solución a los problemas más pequeños y luego une las funciones • Implementa condicionales usando operadores de comparación (mayor, menor, igual) en sus algoritmos • Diseña algoritmos para la solución de problemas incluyendo condicionales, variables y bucles
Programar, depurar y utilizar estructuras de datos r	• Utiliza secuencias, selección y repetición en sus programas • Utiliza diferentes entradas, salidas, y variables en los programas que crea • Hace pruebas y ajusta sus programas usando casos de prueba • Justifica la selección de estructuras de control (diferentes tipos de bucles, condicionales) cuando se deben tomar decisiones respecto a la implementación, desempeño, calidad, costo, tiempo y accesibilidad del código • Recolecta y maneja datos usando funciones y estructuras de datos sencillas
Modelar y simular	• Realiza simulaciones utilizando material concreto y cálculos manuales con el apoyo de tablas para registrar la evolución de variables
Resolver problemas a través de la programación	• Diseña, escribe y depura programas que logren objetivos específicos, incluyendo el control o la simulación de sistemas físicos
Procesar y analizar información	• Explica cómo los sistemas digitales representan texto, imágenes y audio en binario • Responde preguntas a partir del análisis información recolectada

Octavo grado

Grandes ideas	• Los proyectos tecnológicos se pueden resolver a través de un proceso estructurado de pasos • Los proyectos tecnológicos buscan solucionar necesidades con el apoyo de la tecnología, y tienen restricciones • Crear variables permite guardar datos para ser utilizados posteriormente • Las funciones permiten hacer una tarea varias veces sin repetir las líneas de código necesarias para realizarla
Conceptos asociados	• Prototipo • Proceso de diseño • Interfaz de usuario • Librerías • Operadores lógicos (Y, O, NO) • Usabilidad
Interpretar y diseñar algoritmos	• Describe los operadores lógicos y cómo se usan para crear expresiones lógicas y evaluar condiciones • Desarrolla prototipos en papel y digitales e implementar mejoras a partir de retroalimentación recibida por docentes y pares
Programar, depurar y utilizar estructuras de datos	• Incorpora código, librerías o extensiones existentes en programas originales. • Usa diferentes tipos de variables dentro del desarrollo de un programa • Sigue y explica los cambios de valor de las variables durante la ejecución de un programa • Considera las diferentes maneras en que las interfaces de usuario afectan o no la usabilidad de sus aplicaciones
Modelar y simular	• Simula fenómenos y procesos utilizando bucles que permiten iterar el cálculo de expresiones matemáticas sencillas para obtener la evolución de una o varias variables
Resolver problemas a través de la programación	• Sigue un proceso estructurado de resolución de problemas para diseñar soluciones que usan tecnologías computacionales • Considera el proceso de diseño como una forma de resolución de problemas que prioriza las necesidades de un usuario • Resuelve problemas utilizando diferentes aplicativos para diseñar y crear programas, sistemas (por ejemplo, AppInventor, Arduino, code.org)
Procesar y analizar información	• Responde preguntas a partir del análisis de información recolectada por diferentes fuentes (sensores, encuestas …)

Noveno grado

Grandes ideas	• Los procesadores (o dispositivos tecnológicos) se pueden comunicar con el mundo a través de entradas y salidas • Es posible simular eventos aleatorios a través de la computación
Conceptos asociados	• Código binario • Estructuras de datos • Bases de datos • Visualización de datos • Entradas y salidas • Usabilidad
Interpretar y diseñar algoritmos	• Reconoce el proceso de iterar sobre un conjunto de datos para transformar uno por uno • Utiliza contadores para seguir el proceso dentro de un ciclo
Programar, depurar y utilizar estructuras de datos	• Utiliza estructuras de datos y medios para visualizarlos • Explica los elementos de la representación binaria y las maneras comunes en que varios tipos de información son representados en códigos binarios • Utiliza la programación para simular la lectura de variables físicas • Evalúa las ventajas y desventajas de diferentes estructuras de datos, formas de almacenamiento de datos • Comprende que no existen soluciones perfectas que sean útiles en todos los casos, porque optimizar algunas características implica disminuir otras (trade-off)
Modelar y simular	• Simula eventos con componentes aleatorias reconociendo las restricciones de la simulación (efectos, variables y situaciones que no contempla) • Reconoce la importancia de la computación para modelar fenómenos de la naturaleza
Resolver problemas a través de la programación	• Sigue un proceso estructurado de resolución de problemas para diseñar soluciones que usan tecnologías computacionales. • Considera el proceso de diseño como una forma de resolución de problemas que prioriza las necesidades de un usuario y considera las restricciones económicas, ambientales, sociales, técnicas y de usabilidad • Resuelve problemas utilizando diferentes aplicativos para diseñar y crear programas, sistemas (por ejemplo, AppInventor, Arduino, Code.org) • Identifica las necesidades del usuario y evalúa en qué medida sus diseños las atienden
Procesar y analizar información	• Recolecta, analiza y visualiza datos de diferentes fuentes

Décimo grado

Grandes ideas	• Es posible utilizar herramientas de simulación para solucionar problemas vinculados a la realidad y tomar decisiones a partir de lo que se observa • Las simulaciones son útiles para representar los fenómenos físicos, químicos y biológicos, describir sus propiedades y sus relaciones con el entorno • Es posible utilizar simulaciones para comprender y poder hacer predicciones de fenómenos complejos • Las estructuras de datos permiten organizar datos en una computadora para que puedan ser utilizados de manera eficiente
Conceptos asociados	• Tipos de variables • Visualización de datos • Procedimientos • Módulos • Objetos
Interpretar y diseñar algoritmos	• Diseña un algoritmo para desarrollar tareas y lo evalúa en términos de eficiencia, modularidad y claridad • Descompone tareas en componentes más pequeños a través de la detección de patrones y de un análisis sistemático, usando elementos tales como procedimientos, módulos y/o objetos.
Programar, depurar y utilizar estructuras de datos	• Diseña un programa y simula las instrucciones en un lenguaje de programación basado en texto • Usa listas para simplificar soluciones, generalizando problemas computacionales en lugar de usar variables simples de manera repetida • Reconoce estructuras de datos como matrices y vectores para capturar y procesar información • Evalúa diferentes técnicas para procesar, almacenar y validar datos cualitativos y cuantitativos de diferentes fuentes de datos
Modelar y simular	• Utiliza simulaciones interactivas para identificar, reconocer y describir diferentes fenómenos asociados a la física, química y biología • Genera predicciones sobre el comportamiento de algunos fenómenos y las comprueba a partir del uso de simulaciones
Resolver problemas a través de la programación	• Resuelve problemas utilizando diferentes aplicativos para diseñar y crear programas, sistemas (por ejemplo, AppInventor, Arduino, TinkerCAD) • Evalúa cómo las soluciones y sistemas de información existentes representan riesgos, pero a la vez oportunidades para la innovación y el emprendimiento.
Procesar y analizar información	• Utiliza herramientas y técnicas de visualización de datos para la solución de problemas de la vida cotidiana. • Utiliza herramientas y técnicas de visualización de datos para la representación de fenómenos complejos.

Ondécimo grado

Grandes ideas	• Los fenómenos naturales están compuestos de situaciones que no se pueden predecir y por eso es necesario incluir variables aleatorias en su simulación • Cuando los programas suben de complejidad es necesario dividirlos en módulos que cumplen funciones específicas que se pueden reutilizar • Es posible optimizar el comportamiento de los programas utilizando estructuras de datos adecuadas, funciones y objetos • Existen aplicativos computacionales que permiten simular una amplia variedad de fenómenos de la naturaleza
Conceptos asociados	• Modularización , funciones y procedimientos • Clases y objetos • Estructuras y bases de datos
Interpretar y diseñar algoritmos	• Utiliza e interpreta diferentes formas de expresar un algoritmo (diagramas de flujo, seudocódigo ...). • Explica el funcionamiento de diferentes algoritmos para adelantar una tarea (ordenamiento, búsqueda ...) • Compara algoritmos alternativos para una misma tarea • Diseña algoritmos a través de diagramas de flujo y seudocódigo y explicar el paso a paso de su ejecución
Programar, depurar y utilizar estructuras de datos	• Compara lenguajes de programación y encuentra algunas ventajas y desventajas para su uso en una situación dada. • Desarrolla código modular creando y utilizando funciones, componentes, módulos, clases u objetos en un lenguaje de programación. • Utiliza estructuras de datos como matrices para capturar y procesar información • Utiliza variables de generación aleatoria • Documenta el código que escribe para que sea más fácil de depurar
Modelar y simular	• Valida modelos matemáticos contra datos recolectados • Valida y ajusta un modelo computacional a partir de sus resultados • Utiliza modelos programables o parametrizables para simular fenómenos y responder a preguntas
Resolver problemas a través de la programación	• Analiza un problema e identifica los patrones generales que puedan ser aplicados para lograr la solución del problema usando un agente de cómputo • Usa y adapta un algoritmo para resolver un problema usando computación • Implementa y depura una solución para un problema creando sus propios componentes como procesos, módulos u objetos • Validar la solución a través de seguimiento a la ejecución y el diseño de casos de prueba
Procesar y analizar información	• Organiza información utilizando un agente de computación • Procesa información (estadísticas descriptivas ...) • Visualizar datos para encontrar patrones y regularidades

Referencias de consulta

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: what is Involved and what is the role of the computer science education community? *Acm Inroads*, *2*(1), 48-54.

Basu, S., Biswas, G., Sengupta, P., Dickes, A., Kinnebrew, J. S., & Clark, D. (2016). Identifying middle school students' challenges in computational thinking-based science learning. *Research and practice in technology enhanced learning*, *11*(1), 13.

Burning Glass Technologies. (2017). Beyond Tech. In https://www.burning-glass.com/research-project/beyond-tech/

Canu, M. Gómez, M. Duque, M. (2020) Innovación educativa, Programa STEM-Academia, AFFECYN

Department for Education. (2013). The national curriculum in England: Framework document. Disponible en: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/239033/PRIMARY_national_curriculum_-_Computing.pdf

Department for Education. (2019). Guidance: Essential digital skills frameworf. Department for education – UK. Disponible en: https://www.gov.uk/government/publications/essential-digital-skills-framework/essential-digital-skills-framework

Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., ... & Werner, L. (2011). Computational thinking for youth in practice. *Acm Inroads, 2*(1), 32-37.

MEN (2004) Estándares básicos de competencias en ciencias Naturales y ciencias sociales: formar en ciencias: ¡el desafío!, serie guía 7.

MEN (2016) Derechos básicos de aprendizaje: grado 1 a 11.

MEN (2016) Derechos básicos de aprendizaje: transición.

National Research Council. (2011). *Successful K-12 STEM education: Identifying effective approaches in science, technology, engineering, and mathematics*. National Academies Press.

NGSS Lead States. (2013). Next generation science standards: For states, by states. Washington, DC: The National Academy Press

OECD (2019) Pisa 2021 ICT Framework, OECD, Paris

OCDE. (2020). OECD Education and Skills Today. Disponible en:_https://oecdedutoday.com/computer-science-and-pisa-2021/

Royal Society (2012). Shut Down Or Restart?: The Way Forward for Computing in UK Schools. *Royal Society*

The Australian Curriculum. Technologies Foundation (2018). Disponible en: https://www.acara.edu.au/

Vieira, C., Magana, A.J., Garcia, R.E., Jana, A., and Krafcik, M. (2018) Integrating computational science tools into a Thermodynamics course. *Journal of Science Education and Technology (JOST) 27*(1) 1-12 DOI: 10.1007/s10956-017-9726-9

Vieira, C, Duque, M. (2019) Pensamiento computacional en el currículo, documento interno de trabajo.

Webb, M., Davis, N., Bell, T., Katz, Y. J., Reynolds, N., Chambers, D. P., & Sysło, M. M. (2017). Computer science in K-12 school curricula of the 2lst century: Why, what and when?. *Education and Information Technologies*, *22*(2), 445-468.

Wing, J. (2006) Computational Thinking, Communications of the ACM, 49(3).