C+ode
ATRAER | ANIMAR | EDUCAR

COMPUTING AT SCHOOL
EDUCATE · ENGAGE · ENCOURAGE
Part of BCS –The Chartered Institute for IT

Pensamiento Computacional

Guía para profesores

Computing at School desea agradecer a las siguientes organizaciones por su apoyo en la publicación de esta guía:

Hodder Education - the educational division of Hachette UK
Digital Schoolhouse - inspirational computing for kids http://www.digitalschoolhouse.org.uk

Authors

Andrew Csizmadia
School of Education, Newman University, Birmingham

Prof. Paul Curzon
Queen Mary University of London, School of Electronic Engineering and Computer Science
Teaching London Computing Project http://www.teachinglondoncomputing.org

Mark Dorling
CEO and co-founder Progression Pathways http://www.progression-pathways.co.uk
Digital Schoolhouse London Project http://www.digitalschoolhouse.org.uk

Simon Humphreys
National Coordinator for Computing At School http://www.computingatschool.org.uk

Thomas Ng
West Berkshire Council School Improvement Adviser (ICT & Assessment)

Dr Cynthia Selby
University of Southampton http://www.southampton.ac.uk/education/about/staff

Dr John Woollard
University of Southampton http://www.southampton.ac.uk/education/about/staff

Una versión ebook de esta guía, que puede ser compartida libremente con colegas, está disponible en:
http://computingatschool.org.uk/computationalthinking

Contenidos

Introducción 4

La naturaleza de pensamiento computacional 5

Conceptos de pensamiento computacional 6

- Pensamiento algorítmico
- Descomposición
- Generalización (patrones)
- Abstracción
- Evaluación

Técnicas asociadas con pensamiento computacional 10

- Reflexión
- Codificar
- Diseñar
- Analizar
- Aplicar

El pensamiento computacional en el aula 13

Resumen 16

Bibliografía 16

Introducción

Esta guía tiene como objetivo ayudar a desarrollar una comprensión compartida de la enseñanza del pensamiento computacional en escuelas. Presenta un marco conceptual del pensamiento computacional, describe los enfoques pedagógicos para la enseñanza y ofrece guías para la evaluación. Es complementaria a las dos guías publicadas CAS en noviembre de 2013 (Primaria) y junio de 2014 (Secundaria) en el apoyo a la aplicación del nuevo Curriculum Nacional y abraza las descripciones CAS Barefoot y CAS inicio rápido de Informática pensamiento computacional. El pensamiento computacional está en el centro del plan de estudios de computación, sino que también apoya el aprendizaje y el pensamiento en otras áreas del plan de estudios.

Este nuevo plan de estudios de computación tiene un elemento informático enriquecido. La informática es una disciplina académica con su propio cuerpo de conocimiento que puede dotar a los alumnos para convertirse en aprendedores independientes, evaluadores y diseñadores de nuevas tecnologías. En el estudio de la informática, los alumnos adquieren habilidades, conocimientos y una forma única de pensar y resolver problemas: el pensamiento computacional. Permite a los alumnos a comprender el mundo digital de una manera más profunda, al igual que los alumnos comprenden mejor el mundo físico y una lengua extranjera moderna equipa a los alumnos para obtener una comprensión más rica de otras culturas. El pensamiento computacional también da un nuevo paradigma para pensar y entender el mundo más generalmente. Simon Peyton - Jones explica brevemente por qué la informática y el aprendizaje computacional.
El pensar es una habilidad vital básica - además de ser eminentemente transferible - en una charla filmada en TEDxExeter (http://bit.ly/13pJLCR).

Esta guía presenta la naturaleza del pensamiento computacional y proporciona un vocabulario mediante el cual los maestros pueden comunicar, entender y enseñar los conceptos importantes, enfoques y técnicas asociadas con el pensamiento computacional. Se identifica dónde se produce el pensamiento computacional en el plan de estudios de la computación y la forma en que podría ser introducido en el aula.

La naturaleza del pensamiento computacional

El pensamiento computacional proporciona un marco para el estudio de la computación de gran alcance, con aplicación amplia más allá de la computación en sí. Es el proceso de reconocimiento de los aspectos de la computación en el mundo que nos rodea y la aplicación de herramientas y técnicas de computación para entender y razonar sobre sistemas naturales, sociales y procesos artificiales. Permite a los alumnos hacer frente a los problemas, a descomponerlos en partes solucionables y diseñar algoritmos para resolverlos. El término pensamiento computacional fue utilizado por primera vez por Seymour Papert , aunque la profesora Jeannette Wing popularizó la idea en la promoción de pensamiento computacional para todos los nuevos estudiantes universitarios (Wing , 2006) . Ella define el pensamiento computacional como:

" ... Los procesos de pensamiento involucrados en la formulación de problemas y sus soluciones para que la soluciones están representadas en una forma que se puede llevar a cabo eficazmente por un detallado agente de transformación” (Cuny, Snyder, Wing, 2010, cited in Wing, 2011, p.20)

Y:
"La solución puede llevarse a cabo por un ser humano o máquina, o más en general, por las combinaciones de los seres humanos y máquinas”.

(Wing, 2011, p. 20).

El énfasis está claro. Se concentra en los alumnos que realizan un proceso de pensamiento, no en la producción de artefactos oevidencia. El pensamiento computacional es el desarrollo de habilidades de pensamiento y es compatible con el aprendizaje y entendimiento.

Los conceptos de pensamiento computacional

El pensamiento computacional es un proceso cognitivo o pensamiento que implica el **razonamiento lógico** por el cual los problemas se resuelven y procedimientos y sistemas se entienden mejor. Abarca:

● La capacidad de pensar de forma algorítmica;

● La capacidad de pensar en términos de descomposición;

● La capacidad de pensar en generalizaciones, identificando y haciendo uso de patrones;

● La capacidad de pensar en términos abstractos, la elección de buenas representaciones; y

● La capacidad de pensar en términos de evaluación.

Conceptos

Las habilidades de pensamiento computacional permiten a los alumnos acceder a partes de temas de contenidos computacionales que se relacionan con las habilidades de pensamiento y resolución de problemas a través de todo el plan de estudios y a través de la vida en general.

El pensamiento computacional se puede aplicar a una amplia gama de objetos incluyendo: sistemas, procesos, objetos, algoritmos, problemas, soluciones, abstracciones, y colecciones de datos o información. En la siguiente discusión de los conceptos, artefacto se refiere a cualquiera de estos.

Razonamiento lógico

El razonamiento lógico permite a los alumnos a dar sentido a las cosas mediante el análisis y la comprobación de los hechos a través del pensamiento con claridad y precisión. Permite a los alumnos a recurrir a sus propios conocimientos y modelos internos para hacer y verificar las predicciones y sacar conclusiones. Es ampliamente utilizado por los alumnos cuando ponen a prueba, depuración, y algoritmos correctos. El razonamiento lógico es la aplicación de los otros conceptos de pensamiento computacional para resolver problemas.

Diseño y tecnología en los alumnos, el diseño de un modelo de un camión, elegir los materiales para los diferentes elementos del proyecto. Se está empleando generalización cuando reconocen que las propiedades de un material utilizado en una situación que sea adecuado para su uso en otro contexto completamente diferente. Ser capaz de dividir el nuevo proyecto en diferentes partes, lo que requiere diferentes materiales, es un ejemplo de descomposición. El alumno está utilizando el razonamiento lógico para el diseño de un camión.

Los alumnos utilizan el razonamiento lógico cuando se está aprendiendo acerca de la gravedad mediante una cadena ponderada suspendido de la tapa un frasco de vidrio. Antes de inclinar el frasco, los alumnos pueden hacer predicciones sobre el comportamiento de la cadena ponderado. A continuación, pueden evaluar los resultados de sus pruebas. Ellos pueden ser capaces de generalizar el comportamiento a otras situaciones tales como una grúa. El uso novedoso en la comprensión de una propiedad de la gravedad es el razonamiento lógico.

El razonamiento lógico es clave para permitir a los alumnos a depurar su código. Pueden trabajar con sus compañeros para evaluar El código del uno al otro, para aislar errores y sugerir correcciones. Durante este proceso, pueden tener oportunidades emplear la abstracción, la evaluación y el diseño de algoritmos. El uso novedoso en la corrección de errores en el código requiere razonamiento lógico.

Abstracción

Abstracción facilita pensar sobre los problemas o sistemas. La abstracción es el proceso de hacer un artefacto más comprensible a través de la reducción de los detalles innecesarios. Un ejemplo clásico es el del mapa de metro de Londres, es un sistema muy complejo. La representación de Londres en formas particulares (Por lo general mapas o fotos) ayuda a diferentes usuarios. El mapa del metro de Londres es una abstracción altamente refinada con sólo la información suficiente para que el viajero pueda navegar por la red de metro sin la carga innecesaria de la información como la distancia y la posición geográfica exacta. Es una representación que contiene, precisamente, la información necesaria para planificar una ruta desde una estación a otra - y no hay ¡Más!

La habilidad en la abstracción es en la elección del detalle a ocultar de manera que el problema se vuelve más fácil, sin perder todo lo que es importante. Una parte fundamental de la misma es en la elección de una buena representación de un sistema. Diferentes representaciones hacen diferentes cosas fáciles de hacer.

Por ejemplo, un programa de ordenador que juega al ajedrez es una abstracción. Se trata de un conjunto finito de reglas y precisas llevadas a cabo cada vez que es el turno de la computadora. Se está muy lejos de la analógica, emocional y los procesos mentales distraídos llevadas a cabo por un jugador humano de ajedrez. Es una abstracción porque el elimina detalles innecesarios de esos procesos.

Evaluación

La evaluación es el proceso de asegurar que una solución, ya sea un algoritmo, sistema o proceso, es una buena solución: que es adecuado para el propósito. Varias propiedades de las soluciones deben ser evaluadas. ¿Son correctas? ¿Son lo suficientemente rápido? ¿Utilizan recursos económicos? ¿Son fáciles de usar para las personas? ¿Promueven una experiencia adecuada?. Hay un enfoque específico y con frecuencia extrema atención al detalle en la evaluación basado en pensamiento computacional.

Interfaces de ordenador se están desarrollando continuamente para satisfacer las necesidades de diferentes usuarios. Por ejemplo, si se necesita un dispositivo médico para administrar fármacos de forma automática a un paciente, que tiene que ser programable de una manera sencilla y segura, rápida y sin errores. La solución debe garantizar que las enfermeras podrán establecer la dosis fácilmente sin cometer errores y que no va a ser difícil para los pacientes y enfermeras a utilizar. En el diseño propuesto no sería una compensación que debe hacerse entre la velocidad de ingreso de números (eficiencia) y la evitación de error (eficacia y facilidad de uso). El diseño sería juzgado en la especificación propuesta por los médicos, reguladores y expertos en diseño de dispositivos

médicos (criterios) y las normas generales relativas al buen diseño (heurística). Criterios, la heurística y las necesidades de los usuarios, los juicios que se hagan de manera sistemática y rigurosa.

Estos conceptos del pensamiento computacional se han resumido para la fase primaria (aplicable en todos las etapas clave) por CAS Barefoot Computing http://www.BarefootCAS.org.uk.

Pensamiento algorítmico

Pensamiento algorítmico es una forma de llegar a una solución a través de una definición clara de los pasos. Algunos problemas se resuelven d una sola vez; se aplican soluciones, y lo siguiente es abordar. El pensamiento algorítmico se necesita cuando problemas similares tienen que ser resueltos una y otra vez, Y que no tengan que ser pensados de nuevo cada hora. Se necesita una solución que funcione todo el tiempo. Los algoritmos de aprendizaje para hacer la multiplicación o división en la escuela es un ejemplo. Si las reglas simples se siguen precisamente, por un ordenador o una persona, la solución a cualquier multiplicación se puede encontrar. Una vez que se entiende el algoritmo, que no tiene que ser elaborado a partir rayar para cada nuevo problema.
El pensamiento algorítmico es la capacidad de pensar en términos de secuencias y reglas como una forma de resolver problemas o situaciones de entendimiento. Es un conocimiento esencial que los alumnos desarrollan cuando aprenden a escribir sus propios programas.

Descomposición

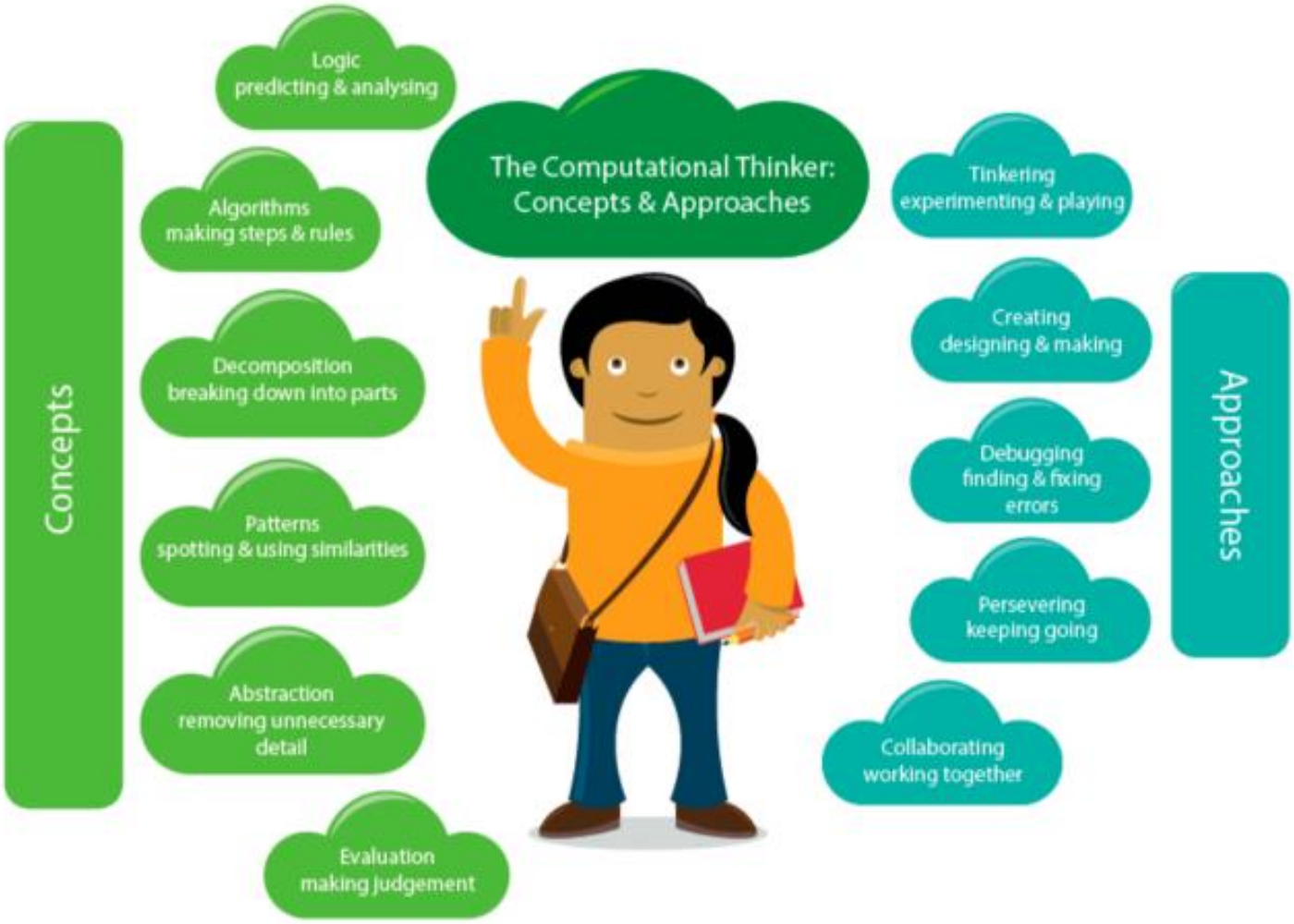
La descomposición es una manera de pensar acerca de los artefactos en términos de sus partes y componentes. Cada pieza debe entenderse, solucionarse, desarrollarse y evaluarse por separado. Esto hace más fácil de resolver problemas complejos, y grandes sistemas más fáciles de diseñar. Por ejemplo, haciendo el desayuno se puede dividir, o descomponer, en actividades separadas , tales como hacer tostadas ; Haz Te; hervir el huevo ; etc. Cada uno de estos, a su vez, podría también ser dividido en una serie de pasos. A través de la descomposición de la tarea original cada parte puede ser desarrollada e integrada más tarde en el proceso. Considerar el desarrollo como un juego: diferentes personas pueden diseñar y crear los diferentes niveles de forma independiente, siempre que los aspectos clave se acuerdan de antemano. Un nivel de árcade sencillo también puede ser descompuesto en varias partes, tales como el movimiento realista de un personaje , el fondo de desplazamiento y el establecimiento de las normas sobre cómo interactúan los personajes.

Generalización (patrones)

La generalización se asocia con la identificación de patrones, similitudes y conexiones, y la explotación de las características. Es una forma de resolver rápidamente los nuevos problemas sobre la base de las soluciones en los problemas anteriores, y la construcción en la experiencia previa. Haciendo preguntas tales como "¿Esto es similar a un problema que ya he solucionado?"

Y "¿Cómo es diferente? “Son importantes aquí, como es el proceso de reconocimiento de patrones, tanto en los datos utilizados y se utilizan los procesos / estrategias. Algoritmos que resuelven algunos problemas específicos se pueden adaptar para resolver toda una clase de problemas similares. Entonces cada vez que se presenta un problema de esa clase, la solución el general puede ser aplicada.

Por ejemplo, un alumno utiliza una tortuga de piso para dibujar una serie de formas, tales como un cuadrado y un triángulo. Los alumnos escriben un programa para dibujar las dos formas. Luego, quieren dibujar un octágono y una forma de 10 caras. A partir del trabajo con el cuadrado y el triángulo, detectaron que hay una relación entre el número de las dos formas y los ángulos involucrados. A continuación, pueden escribir un algoritmo que expresa esta relación y utilizarlo para dibujar cualquier polígono regular.



Técnicas asociadas con el pensamiento computacional

Hay una serie de técnicas empleadas para demostrar y evaluar el pensamiento computacional. Pensar en esto como "tarea computacional". Estos son el equivalente del "método científico "en "Ciencias computacionales". Estas son las herramientas con las que el pensamiento computacional se opera en el aula, lugar de trabajo y el hogar. Se reflejan en la clave de la práctica de la etapa 3 del aula.

Reflexión

La reflexión es la habilidad de hacer juicios (evaluación) que son justos y honestos en situaciones complejas que no están libres de valores. Dentro de la informática esta evaluación se basa en criterios que se utilizan para especificar el producto, heurística (o reglas de oro) y las necesidades del usuario para guiar los juicios.

Codificación

Un elemento esencial del desarrollo de cualquier sistema informático traduce el diseño en forma de código y evaluarlas de manera a garantizar que funcione correctamente en todas las condiciones. La depuración es la aplicación sistemática de las habilidades de análisis y evaluación utilizando como prueba, la localización, y el pensamiento lógico para predecir y verificar los resultados.

Diseño

Diseños involucrados en trabajar la estructura, apariencia y funcionalidad de los artefactos. Se trata de la creación del diseño, incluidas las representaciones legibles por humanos, tales como diagramas de flujo, guiones gráficos,

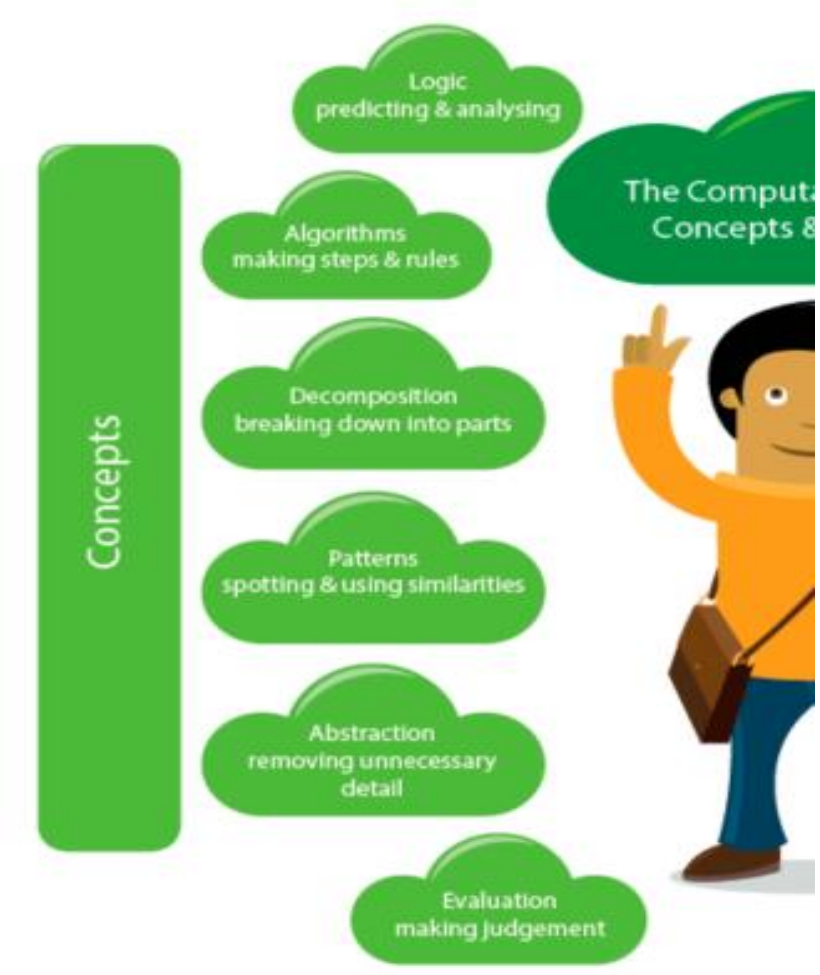
pseudocódigo, diagramas de sistemas, etc. Se trata de nuevas actividades de descomposición, y la abstracción en el diseño de algoritmos.

Analizar

Analizar consiste en dividir en partes todos los componentes (descomposición), reduciendo la innecesaria complejidad (abstracción), la identificación de los procesos (algoritmos) y la búsqueda de elementos comunes o patrones (generalización). Se trata de utilizar el pensamiento lógico tanto para comprender mejor las cosas como de evaluar su adecuado propósito.

Aplicar

La aplicación es la adopción de soluciones pre-existentes para satisfacer las necesidades de otro contexto. Es generalización - la identificación de patrones, similitudes y conexiones - y explotar aquellas características de la estructura o función de artefactos. Un ejemplo incluye el desarrollo de un subprograma o algoritmo en un contexto en el que se pueden volver a utilizarse en un contexto diferente.




ational Thinker:
& Approaches
Tinkering
experimenting & playing
Creating
designing & making
Debugging
finding & fixing
errors
Persevering
keeping going
Collaborating
working together
Approaches

El pensamiento computacional en el aula

Cada uno de los conceptos de pensamiento computacional descritos anteriormente (pensamiento algorítmico, descomposición etc....) se identifican distintos comportamientos que se pueden observar en el aula.

Pensamiento algorítmico

El pensamiento algorítmico es la capacidad de pensar en términos de secuencias y las reglas como una forma de resolver problemas. Es un conocimiento esencial que los alumnos desarrollan cuando aprenden a escribir sus propios programas de ordenador. Lo siguiente puede observarse en el aula

● El primer grupo consiste en formular instrucciones para lograr un efecto deseado.

● La formulación de las instrucciones a seguir en un orden determinado (secuencia).

● La formulación de instrucciones que utilizan operaciones aritméticas y lógicas.

● Escribir secuencias de instrucciones que almacenan, mueven y manipulan los datos (variables y asignación).

● Escribir instrucciones que eligen entre diferentes instrucciones constituyentes (selección).

● Escribir instrucciones repiten grupos de instrucciones constituyentes (bucles / iteración).

● Agrupación y nombramiento de un conjunto de instrucciones que realizan una tarea bien definida para hacer una nueva instrucción (subrutinas, procedimientos, funciones, métodos).

● Escribir instrucciones que implican subrutinas que utilizan copias de sí mismos (recursividad).

●Escribir conjuntos de indicaciones que se pueden seguir al mismo tiempo por diferentes agentes (computadoras / personas, pensamiento y procesamiento paralelo, concurrencia).

● Escribir un conjunto de reglas declarativas (codificación en Prolog o un lenguaje de consulta de base de datos).

También implica:

● El uso de una notación adecuada para escribir código para representar cualquiera de los anteriores.

● Creación de algoritmos para probar una hipótesis.

● Creación de algoritmos que dan soluciones basadas en la experiencia (heurística).

● Creación de descripciones algorítmicas de los procesos del mundo real con el fin de comprender las mejor (Modelado computacional).

● El diseño de soluciones algorítmicas que tengan en cuenta las capacidades, limitaciones y deseos de las personas que los utilizará.

Descomposición

● El desglose de los artefactos en sus distintos componentes para que sean más fáciles de trabajar.

● El desglose de un problema en las versiones más simples del mismo problema que puede ser resuelto de la misma manera

Generalización (patrones)

● La identificación de patrones y los puntos comunes en los artefactos.

● La adaptación de soluciones, o partes de las soluciones, por lo que se aplican a toda una clase de problemas similares.

● La transferencia de ideas y soluciones de un problema a otro.

Abstracción

La abstracción es el proceso de hacer un artefacto más comprensible ocultando detalles.

● Reducción de la complejidad mediante la eliminación de detalles innecesarios.

● La elección de una forma de representar un artefacto, para permitir que sea manipulado en formas útiles.

● Cómo ocultar la complejidad de un artefacto (que oculta la complejidad funcional).

● Ocultación de la complejidad en los datos, por ejemplo mediante el uso de estructuras de datos.

● Identificar las relaciones entre las abstracciones.

● Filtrado de la información en el desarrollo de soluciones.

Evaluación

La evaluación es el proceso de garantizar una buena para determinado propósito.

● valorar si un artefacto es adecuado para el propósito.

● Evaluar si un artefacto hace lo correcto (corrección funcional).

● El diseño y la ejecución de planes de prueba y la interpretación de los resultados (de pruebas).

● Evaluar si el rendimiento de un artefacto es lo suficientemente bueno (utilidad: eficacia y eficiencia).

● Al comparar el rendimiento de los artefactos que hacen lo mismo.

● Hallar una compensación entre demandas conflictivas.

● Evaluar si un artefacto es fácil para la gente a usar (usabilidad).

● Evaluar si un artefacto da una experiencia positiva cuando se emplea adecuadamente (experiencia del usuario).

● Evaluación de cualquiera de los anteriores con las especificaciones y criterios establecidos.

● Paso a paso, mediante procedimientos o algoritmos / código paso a paso para trabajar a cabo lo que hacen (en seco / rastreo).

● El uso de argumentación rigurosa para justificar que un algoritmo funciona (prueba).

● El uso de argumentación rigurosa para comprobar la capacidad de uso o el rendimiento de un artefacto (evaluación analítica).

● El uso de métodos que implican la observación de un artefacto en uso para evaluar su capacidad de uso (evaluación empírica).

● La evaluación de si un producto cumple los criterios de rendimiento generales (heurística).

Resumen

El pensamiento computacional es una habilidad importante para la vida, que ahora todos los alumnos necesitan desarrollar. Es fundamental en el mundo en el que vivimos y para entender el enriquecimiento digitalmente. Por lo tanto, es un concepto central en el nuevo Computing Programme of Study. El marco que se presenta aquí se explica el concepto e ilustra con experiencias de aprendizaje eficaces que se desarrollan las habilidades, además de ser una forma de evaluar su desarrollo.

Hay muchas fuentes existentes de ideas para las actividades de clase que potencialmente desarrollan el pensamiento computacional con ejemplos incluidos:

En español: http://www.codemas.org

- Barefoot Computing http://www.barefootcas.org.uk
- Computing at School http://www.computingatschool.org.uk
- CS4FN http://www.cs4fn.org/
- CS Insidehttp://csi.dcs.gla.ac.uk/
- CSTA Computational Task Force http://www.csta.acm.org/Curriculum/sub/CompThinking.html
- CS Unplugged http://csunplugged.org/
- Digital Schoolhouse http://www.digitalschoolhouse.org.uk
- Teaching London Computing http://www.teachinglondoncomputing.org

Bibliografía

BCS, The Chartered Institute for IT. 2014. Call for evidence - UK Digital Skills Taskforce. Available: http://bit.ly/1Li8mdn [Accessed 15-04-2015].

Department for Education. 2014. The National Curriculum in England, Framework Document. Reference:DFE-00177-2013. Available:https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/335116/Master_final_national_curriculum_220714.pdf [Accessed 15-04-2015].

Dorling, M. & Walker, M. 2014. Computing Progression Pathways. Available: http://www.hoddereducation. [Accessed 28-02-14].

Dorling, M., Selby, C. & Woollard, J. 2015. Evidence of Assessing Computational Thinking. IFIP 2015, A New Culture of Learning: Computing and Next Generations. Vilnius, Lithuania. Available: http://eprints.soton. [Accessed 01-07-2015].

Selby, C. & Woollard, J. 2013. Computational thinking: the developing definition. Definition Available: http://eprints.soton.ac.uk/356481/ [Accessed 01-04-2014].

Wing, J. 2006. Computational Thinking. Commun. ACM, 49, 3, 33-35. Available: http://dl.acm.org/citation.

[Accessed 15-04-2015].

Wing, J. 2011. Research Notebook: Computational Thinking - What and Why? The Link. Pittsburgh, PA: Carneige Mellon. Available: http://www.cs.cmu.edu/sites/default/files/11-399_The_Link_Newsletter-3.pdf

[Accessed 15-04-2015].