

Marco Referencial

Pensamiento Computacional

Introducción

En los últimos años se ha incrementado tremendamente el impacto de las tecnologías de la información y su presencia en la vida cotidiana. Con ello ha cambiado la forma en la que accedemos al conocimiento, buscamos información, nos relacionamos, aprendemos y enseñamos (redes sociales, plataformas educativas, inteligencia artificial). La tecnología digital evoluciona para convertirse en una herramienta de uso diario en el aula. En 2006, Jeannette Wing publica un artículo que retoma las ideas de Seymour Papert y busca catalizar en la definición del Pensamiento Computacional una nueva competencia a nivel mundial, que debería ser incluida en la formación de todos, ya que representa un ingrediente vital del aprendizaje de la ciencia, la tecnología, la ingeniería y las matemáticas de la actualidad, y que no se acota a la programación (Fraillon et al, 2019; Kong & Abelson, 2019; Shute et al, 2017).

Pensamiento Computacional

El Pensamiento Computacional se basa en el principio de la externalización y la materialización de ideas o creaciones mentales dirigidas a un medio computacional (Kong, Abelson, 2019). Ceibal desde su práctica se acerca a una definición que considera al pensamiento computacional como la habilidad para reconocer aspectos del mundo real que pueden ser modelados como problemas y para diseñar y evaluar soluciones algorítmicas que puedan ser implementadas computacionalmente (Fraillon et al, 2019). Es decir, se entiende al pensamiento computacional como una forma de razonar y resolver problemas desde la lógica de la computación.

En este marco se busca que los estudiantes desarrollen los contenidos fundacionales de las Ciencias de la Computación y aprendan nuevos enfoques para la resolución de problemas aprovechando el potencial del pensamiento computacional, para ser usuarios y creadores de la tecnología de hoy y del futuro.

El programa de Pensamiento Computacional propone trabajar a partir de propuestas o proyectos guías que ponen en juego los siguientes principios y dimensiones. Estos elementos tienen en cuenta la experiencia de CEIBAL en el sistema educativo uruguayo, dialogan con las competencias de la Red Global de Aprendizajes¹, los usos de la Estrategia de Ciudadanía Digital², y consideran los marcos conceptuales de otros países reconocidos a nivel de la comunidad internacional³.

¹ redglobal.edu.uy

² www.ceibal.edu.uy/es/articulo/uruguay-cuenta-con-estrategia-de-ciudadania-digital

³ K-12 CS Framework - CSTA, Computing at School, European Commission Science Hub, Collage Board, Currículo de Referência em Tecnologia e Computação

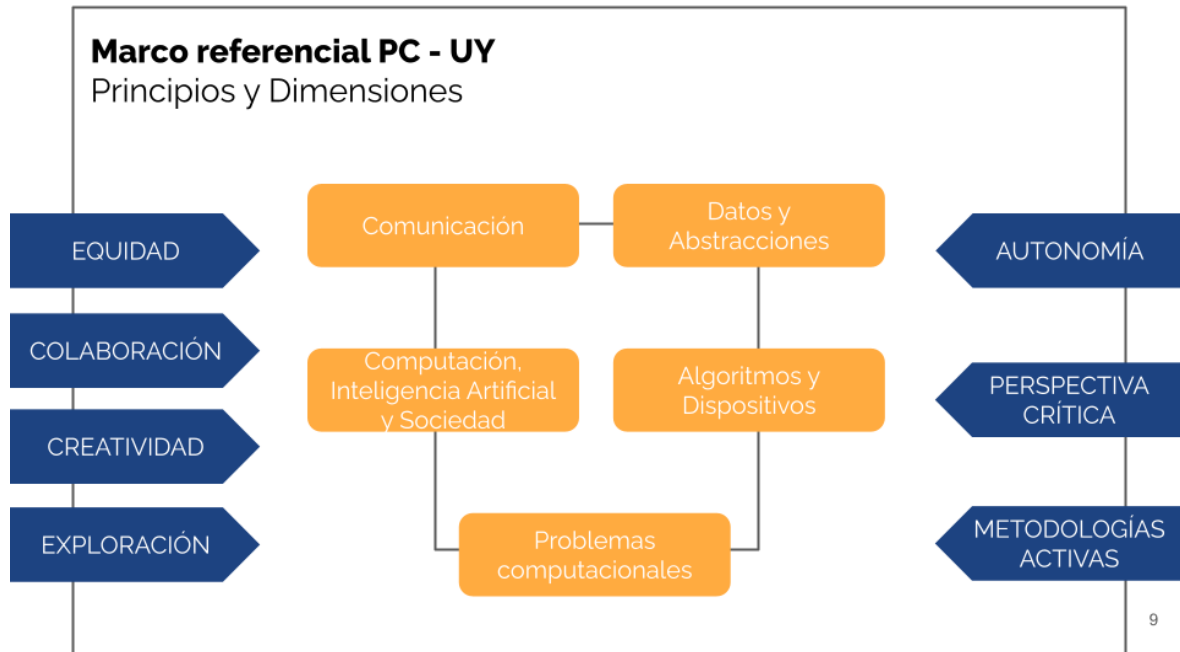


Imagen 1. Principios y dimensiones del programa de pensamiento computacional

Principios

Los principios son características fundamentales y se incorporan de forma transversal a todas las propuestas de trabajo. En este sentido no son elementos intrínsecos del pensamiento computacional, sino pautas que guían la construcción de propuestas y actividades para generar los entornos de aprendizaje.

A. Equidad

Educar teniendo en cuenta las diferencias y necesidades individuales, sin que las condiciones económicas, demográficas, geográficas, étnicas y/o de género repercutan en la educación de los estudiantes.

B. Colaboración⁴

Trabajar en equipo de manera independiente y sinérgica. Desarrollar fuertes habilidades interpersonales. Organizar al equipo para asumir desafíos. Tomar decisiones retadoras y contribuir con el aprendizaje de los demás.

C. Exploración

Valorar, respaldar, acompañar y promover la sorpresa para promover la actitud de búsqueda y de indagación que permitan plantear preguntas, hipótesis y explicaciones por parte de los estudiantes.

⁴ Competencias de la Red global de aprendizajes

D. Creatividad⁵

Tener visión emprendedora, hacer preguntas adecuadas para generar oportunidades e ideas novedosas. Transformar esas ideas en acciones con impacto social.

E. Autonomía

Promover la exploración sin miedo a cometer errores, asumiendo riesgos y tomando la iniciativa como estrategia para involucrarse activamente en el proceso de creación y fomentando la motivación intrínseca de los estudiantes.

F. Perspectiva crítica

Evaluar críticamente la información y los argumentos, identificar patrones y conexiones, desarrollar conocimiento significativo y aplicarlo al mundo real

G. Metodologías activas

Utilizar métodos, técnicas y estrategias que conviertan al estudiante en el centro del proceso de enseñanza/aprendizaje y fomenten la participación activa y la curiosidad de los estudiantes en la construcción de su aprendizaje.

Dimensiones

Las dimensiones son temas, conceptos o ideas poderosas que organizan la actividad del programa de PC y que están entrelazadas con las habilidades promovidas: abstracción, descomposición, generalización, evaluación y pensamiento algorítmico (Dagiené y Sentance, 2016).

1. Comunicación y Colaboración

Documentar los procesos de trabajo en múltiples soportes. Comunicar las ideas de forma clara y entendible. Desempeñar prácticas y procesos de forma grupal o en parejas, que involucren tomar decisiones en conjunto, división de tareas, uso de herramientas para trabajo colaborativo y para presentación de información o ideas.

- a. **Generar** relaciones con personas que posean perspectivas, habilidades y personalidades diferentes para trabajar grupalmente y/o entre pares, creando normas equipo, expectativas y cargas de trabajo equitativas para aumentar la eficiencia y la eficacia.
- b. **Comunicar** las ideas de forma clara, entendible y asertivamente para generar ambientes de trabajo propicios para la interacción del grupo.
- c. **Participar** en procesos donde estén involucradas dos o más personas donde se utilicen herramientas de trabajo colaborativo, que permitan presentar información e ideas a los demás.
- d. **Documentar** la información, las ideas, los procesos de diseño y los procesos de desarrollo en múltiples soportes.

2. Computación, Inteligencia Artificial, Sociedad y Equidad

⁵ Competencias de la Red global de aprendizajes

Identificar los impactos de la computación en las personas y la sociedad. Describir, entender y explicar los usos frecuentes de la tecnología en la vida cotidiana, cómo las redes sociales o la inteligencia artificial.

- a. **Comprender y reflexionar** sobre la existencia de diferentes perspectivas, propias y ajenas, que pueden existir al enfrentarse a diferentes problemas o al diseñar y/o desarrollar dispositivos.
- b. **Abordar** durante el proceso de diseño, la diversidad de los usuarios finales para producir dispositivos con amplia accesibilidad y usabilidad.
- c. **Identificar** los impactos de la computación en las personas y la sociedad. Pudiendo describir, entender y explicar los usos frecuentes de la tecnología en la vida cotidiana.
- d. **Reflexionar** sobre el uso de internet y las redes sociales, entendiendo la diversidad de personas que navegan y la necesidad de protegerse ante los posibles ataques.
- e. **Comprender** el concepto de inteligencia artificial y el impacto que puede tener socialmente.

3. Problemas computacionales

Solucionar problemas desde un enfoque computacional involucra identificar y definir el problema, dividirlo en partes, implementar soluciones y evaluar viabilidad y alcance. Se trata de problemas reales y que involucran abordajes y conocimientos de diferentes disciplinas.

- a. **Identificar** problemas reales que puedan ser solucionados desde un enfoque computacional, involucrando abordajes y conocimientos de diferentes disciplinas.
- b. **Dividir** el problema en subproblemas manejables, a partir de la integración de los datos disponibles sobre el problema.
- c. **Buscar e implementar** soluciones nuevas o existentes a los subproblemas para solucionar el problema. Evaluando la viabilidad y el alcance de las soluciones al problema

4. Análisis de datos y Abstracciones

En el análisis de datos resulta fundamental la identificación de aspectos importantes y selección de información relevante. Entender y elegir distintas formas de representación según la situación para modelar el problema. Reconocer características comunes y elaborar generalizaciones y conclusiones, re-utilizando soluciones y gestionando la complejidad.

- a. **Analizar** los datos requiere identificar los aspectos importantes, seleccionar la información relevante y buscar patrones. Así mismo seleccionar la mejor forma de representación y sistematización de los datos ayudará a modelar el problema de manera óptima.
- b. **Abstraer** requiere reconocer las características comunes de un conjunto de procesos o fenómenos relacionados, realizando generalizaciones. Modelar soluciones que puedan aplicarse a múltiples situaciones que permitan reducir la complejidad del proceso y/o solución, llegando a conclusiones fundamentadas.

5. Algoritmos, programas y dispositivos

El proceso de creación de algoritmos, programas o dispositivos involucra poner en práctica la creatividad y la exploración en la materialización de abstracciones para la resolución de problemas. El desarrollo de las habilidades que componen este proceso implican la puesta en práctica e iteración de todas las etapas.

- a. **Planificar** el desarrollo y/o diseño de un dispositivo, algoritmo y/o programa es un proceso iterativo que incluye exploración, reflexión, revisión y modificación del plan, teniendo en cuenta sus características, el tiempo y los recursos del que se dispone y las expectativas de los usuarios.
- b. **Crear** un dispositivo, algoritmo y/o programa requiere programar una secuencia de pasos (algoritmo) para alcanzar el objetivo, por lo que, la creación parte de una intención que puede ser práctica, expresión personal y/o para abordar un problema social.
- c. **Modificar** un dispositivo, algoritmo y/o programa es un proceso de iteración y automatización que tiene por objetivo mejorar o personalizar uno existente.
- d. **Experimentar** y comprender los conceptos, posibilidades y aplicaciones del aprendizaje automático y la inteligencia artificial.
- e. **Evaluar**, probar y ajustar como método iterativo para la mejora de las creaciones y procesos. Implica identificar errores, comparar versiones, evaluar efectos en la robustez, accesibilidad, usabilidad y eficiencia de los procesos.

Propuestas Pedagógicas

Las propuestas pedagógicas son proyectos que promueven en los estudiantes los objetivos definidos para cada nivel en la siguiente sección. Cada propuesta se divide en varias etapas, en las que se plantea una secuencia de actividades que con un hilo conductor de la temática del proyecto acompaña el desarrollo y materialización de un dispositivo o programa informático. La duración promedio de las propuestas es de 8 semanas de trabajo. A su vez cada etapa tiene actividades que articulan aspectos definidos de algún contenido curricular (lengua, matemática, ciencias, etc) con elementos del pensamiento computacional. En la Imagen 2 se muestra el esquema del recorrido para cada uno de los niveles de pensamiento computacional.



Imagen 2. Esquema de las propuestas de PC y contenidos de los niveles.

Para ilustrar un recorrido de una de las propuestas, en la imagen 3 se observan las líneas principales de cada etapa de la propuesta de La máquina de dibujar.

→ **Síntesis del recorrido en Etapas**



Imagen 3. Ejemplo de recorrido de la propuesta La máquina de Dibujar

Objetivos por niveles

Nivel 4 - 1er año de PC

Comunicación	Computación, Sociedad y Equidad	Resolución de problemas
<p>Participar de un proyecto grupal realizando las consignas propuestas.</p> <p>Explorar varias modalidades para documentar y comunicar ideas.</p> <p>Utilizar herramientas digitales colaborativas para potenciar sus ideas creativas.</p>	<p>Comprender la dualidad de roles a desempeñar: usuarios y programadores.</p> <p>Reconocer que el comportamiento de las computadoras y, por lo tanto, el uso que podemos darles, es el resultado de la ejecución de un programa.</p> <p>Entender que todas las acciones llevadas a cabo por las computadoras dependen de las instrucciones que les damos los humanos.</p> <p>Reflexionar sobre la existencia de diferentes perspectivas, propias y ajenas, que pueden existir al visualizar las distintas soluciones que surgen de la resolución de problemas.</p> <p>Reflexionar sobre el uso de internet, cómo se transfieren y comparten datos</p> <p>Comprender que los datos compartidos en internet inciden en los demás.</p>	<p>Identificar, descomponer y resolver problemas sencillos de programación que requieran de una variedad acotada de instrucciones.</p> <p>Comprender la necesidad de probar/iterar con diferentes procedimientos para alcanzar una solución.</p>
Datos y Abstracciones	Algoritmos, programas y dispositivos	
<p>Reconocer que una instrucción, objeto o fenómeno puede tener múltiples representaciones</p> <p>Identificar los aspectos importantes y reflexionar sobre la información relevante de los datos de un problema sencillo.</p>	<p>Comprender que el resultado de la ejecución de un programa depende tanto de sus instrucciones como de eventos de entrada originados por el usuario.</p> <p>Experimentar y comprender la lógica de la programación por bloques, y sus opciones básicas.</p> <p>Seguir un plan para crear programas utilizando instrucciones simples, repetición y eventos.</p> <p>Comprender y explicar los comportamientos de sus propios programas.</p> <p>Encontrar y reflexionar sobre los errores cometidos en el proceso de resolver problemas simples.</p>	

Nivel 5 - 2do año de PC

Comunicación	Computación, Sociedad y Equidad	Resolución de problemas
<p>Participar de forma proactiva en un proyecto grupal.</p> <p>Transmitir y escuchar ideas dentro del grupo de trabajo.</p> <p>Integrar el uso de herramientas de documentación de la información y el proceso de desarrollo del proyecto.</p>	<p>Reflexionar sobre la seguridad de los datos compartidos en internet y redes sociales.</p> <p>Conocer y experimentar que se puede utilizar las computadoras para extraer información variada a partir de un conjunto de datos</p> <p>Comprender que ciertos problemas sociales del entorno pueden ser abordados desde una perspectiva computacional.</p> <p>Entender que un dispositivo automatizado puede interactuar con el entorno.</p>	<p>Utilizar la estrategia de división de un problema en subproblemas.</p> <p>Resolver problemas computacionales utilizando algunas herramientas básicas de programación (como la alternativa condicional, las repeticiones, las variables, etc).</p> <p>Recuperar soluciones construidas en experiencias anteriores para adaptarlas a nuevos problemas.</p>
Datos y Abstracciones	Algoritmos, programas y dispositivos	
<p>Identificar los aspectos importantes y seleccionar la información relevante de los datos de un conjunto de problemas sencillos.</p> <p>Reconocer las generalidades de los comportamientos de sus programas, en términos abstractos, para que puedan ser adaptados, combinados y/o reutilizados.</p> <p>Comprender que los modelos son representaciones de diferentes escenarios, y permiten al usuario experimentar con distintas condiciones y sus consecuencias.</p>	<p>Planificar, crear y modificar, con ayuda del docente, un programa y/o solución tecnológica particular que interactúe con el entorno.</p> <p>Controlar aspectos gráficos de sus programas.</p> <p>Utilizar en sus programas de forma independiente o combinada bloques de control, variables, sensores, eventos y operadores.</p> <p>Reconocer el uso del método de desarrollo incremental.</p> <p>Probar, ajustar y corregir mediante la iteración durante el desarrollo de las creaciones.</p> <p>Identificar las diferencias entre versiones y resultados de dispositivos y/o programas.</p>	

Nivel 6 - 3er año de PC

Comunicación	Computación, Sociedad y Equidad	Resolución de problemas
<p>Colaborar y construir conjuntamente un proyecto grupal.</p> <p>Reflexionar sobre los diferentes roles que pueden existir dentro de un grupo de trabajo.</p>	<p>Reflexionar sobre los sesgos (género, nivel socioeconómico, etc.) que existen en la computación.</p> <p>Identificar problemas de seguridad de los datos compartidos en internet y redes sociales.</p> <p>Reconocer distintos usos de los algoritmos de inteligencia artificial en la computación.</p>	<p>Definir diferentes soluciones computacionales que pueden resolver un problema dado.</p> <p>Incorporar la estrategia de división de un problema en subproblemas.</p> <p>Seleccionar las herramientas de programación adecuadas para el subproblema a resolver.</p> <p>Recuperar estrategias de solución y herramientas de programación construidas en experiencias anteriores para adaptarlas a nuevos problemas.</p>
Datos y Abstracciones	Algoritmos, programas y dispositivos	
<p>Analizar los datos importantes y la información relevante de un problema dado.</p> <p>Reconocer las generalidades de los comportamientos de sus programas⁶, en términos abstractos, para que puedan ser adaptados, combinados y/o reutilizados.</p> <p>Explorar distintas formas de representación y sistematización de los datos para modelar un problema.</p>	<p>Planificar, crear y modificar un programa y/o solución tecnológica particular que interactúe con el entorno.</p> <p>Utilizar las herramientas de programación adecuadas para implementar sus programas.</p> <p>Utilizar el método de desarrollo incremental.</p> <p>Comprender la posibilidad de reutilizar partes de programas ya creados en la construcción de programas nuevos.</p> <p>Identificar y corregir, con ayuda del docente, errores mediante un proceso sistemático.</p> <p>Comparar las diferencias entre versiones y resultados de dispositivos y/o programas.</p>	

⁶ Se entiende que los programas de este nivel serán más complejos que los de N-5

Referencias

- College Board. (2020). AP Computer Science Principles. Course And Exam Description. Retrieved from <https://apcentral.collegeboard.org/pdf/ap-computer-science-a-course-and-exam-description.pdf?course=ap-computer-science-a>
- Csizmadia, A., Curzon, P., Dorling, M., Humphreys, S., Ng, T., Selby, C., & Woollard, J. (2015). Computational thinking-A guide for teachers.
- Dagienė, V., & Sentance, S. (2016). It's computational thinking! Bebras tasks in the curriculum. In International conference on informatics in schools: Situation, evolution, and perspectives (pp. 28-39). Springer, Cham.
- Frailon, J., Ainley, J., Schulz, W., Duckworth, D., & Friedman, T. (2019). *IEA International Computer and Information Literacy Study 2018 assessment framework*. Springer.
- ISTE & CESTA (2011). Computational Thinking in K–12 Education leadership toolkit. *Computer Science Teacher Association*
- Juškevičienė, A., & Dagienė, V. (2018). Computational thinking relationship with digital competence. *Informatics in Education, 17(2), 265-284*.
- K–12 Computer Science Framework. (2016). Retrieved from <http://www.k12cs.org>.
- National Research Council. (2010). *Report of a workshop on the scope and nature of computational thinking*. National Academies Press.
- Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review, 22, 142-158*.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49(3), 33-35*