

K–12 Estándares para las Ciencias de la Computación

Revisado 2011

Grupo de trabajo de los estándares de CSTA

Deborah Seehorn, Presidente

Departamento de Instrucción Pública de Carolina del Norte

Stephen Carey

Departamento Escolar de Brunswick

Brian Fuschetto

Escuela Secundaria Lyndhurst

Irene Lee

Instituto Santa Fe

Daniel Moix

Universidad de Ouachitas

Dianne O’Grady-Cunniff

Escuela Secundaria Westlake

Barbara Boucher Owens

Universidad Southwestern

Chris Stephenson

Asociación de Maestros de Ciencias de la Computación

Anita Verno

Universidad Comunitaria Bergen

Asociación de Maestros de Ciencias de la Computación

Asociación de maquinaria para computación

2 Penn Plaza, Suite 701

New York, New York 10121-0071

Copyright © 2011 por la Asociación de Maestros de Ciencias de la Computación (CSTA) y la Asociación de Maquinaria para Computación, Inc (ACM). El permiso para hacer copias digitales o impresas de partes de este manual para uso personal o en el aula se concede sin pago, siempre que las copias no se realicen o se distribuyan para fines de lucro o para fines comerciales y que las copias lleven esta notificación completa en la primera página. Los Derechos de autor de componentes de este trabajo que no sean propiedad de la ACM deben ser honradas. La abstracción del crédito está permitida.

Para copiar, volver a publicar, incluirlo en servidores o redistribuir en las listas, se requiere autorización específica previa y / o el pago de una cuota. El permiso para volver a publicar se debe solicitar a: Departamento de Publicaciones. ACM, Inc, Fax +1-212-869-0481 o al correo electrónico permissions@acm.org.

Las copias de los artículos que llevan un código en la parte inferior de la primera o la última página, son permitidas siempre que la cuota por copia, indicada en el código en la parte inferior de la página se pague a través del Centro de Acreditación de los Derechos de Autor, 222 Rosewood Drive, Danvers, MA 01923.

ACM ISBN: # 978-1-4503-0881-6

ACM Numero de Orden: # 104111

Costo: \$15.00

Se pueden ordenar copias adicionales prepago a: Departamento de Pedidos de ACM

Teléfono: 1-800-342-6626

P.O. Box 11405 (U.S.A. and Canada)

Church Street Station +1-212-626-0500

New York, NY 10286-1405 (Todos los demás países)

Fax: +1-212-944-1318

E-mail: acmhelp@acm.org

Agradecimientos

El Grupo de Trabajo de los estándares CSTA agradece a las siguientes organizaciones e individuos. En primer lugar agradecemos a la Fundación Nacional de Ciencias por su apoyo a CSTA como organización y por su compromiso con mejorar la enseñanza de computación. También agradecemos a nuestros patrocinadores corporativos (Google, Microsoft, la Junta Universitaria y el Instituto Anita Borg) por su apoyo continuo.

Estamos especialmente agradecidos a todas las personas que se tomaron el tiempo para leer y revisar este documento, en especial:

Gail Chapman, *ECS Director of National Outreach, Exploring Computer Science Program*

Renee Ciezki, *Instructor de Computación, Universidad Comunitaria Estrella Mountain*

Creighton Edington, *Escuelas Públicas de Deming*

Dr. Barbara Ericson, *Directora del Programa de Extensión de Computación para el Instituto de educación de computación, Georgia Tech*

Dr. Michael Erlinger, *Profesor de Ciencias de la Computación, Universidad Harvey Mudd*

Dave Feinberg, *Profesor de Docentes de Ciencias de la Computación, Universidad Carnegie Mellon*

Baker Franke, *Universidad de Chicago, Secundaria Lab*

Dr. Joanna Goode, *Profesor Asistente de Educación Docente, Universidad de Oregon*

Dr. David Hemmendinger, *Profesor Emérito del Depto. de Ciencias de la Computación, Universidad Unión*

Stephanie Hoepfner, *Escuelas del Noreste de Clermont*

Joe Kmoch, *Escuelas Públicas de Milwaukee*

Carl Lyman, *Oficina de Educación del Estado de UTAH*

Dr. Jane Margolis, *Investigador Senior, Universidad de Posgrado en Educación de UCLA*

Deepa Muralidhar, *Escuela Secundaria North Gwinnett*

Joshua Paley, *Escuela Secundaria Henry M. Gunn*

Tammy Pirmann, *Springfield Township High School*

Kelly Powers, *Academia de Matemáticas avanzadas y ciencias*

Beth Richtsmeier, *Meridian Technical Charter HS*

Dr. Eric Roberts, *Profesor de Ciencias de la Computación, Universidad de Stanford*

Esther Romero, *Escuelas Públicas de Portland*

Cameron Wilson, *Director de Políticas Públicas, ACM*

Nancy Yauneridge, *Escuela Saint Benedict*

En especial queremos agradecer al Dr. Allen Tucker por su visión y liderazgo en la creación de los primeros estándares CSTA / ACM K-12 y por su apoyo al comienzo de este proyecto.

También agradecemos a nuestro maravilloso diseñador Bob Vizzini y nuestro exigente corrector de estilo Kate Conley. Gracias al Presidente de CSTA Dr. Steve Cooper y a la Presidenta del Consejo Asesor, Dra. Debra Richardson y a todos los miembros de la Junta Directiva y el Consejo Asesor.

Finalmente, a todo el personal y líderes voluntarios de ACM, que ayudaron a CSTA a existir y para que continúen apoyándonos todos los días.

Deborah Seehorn, *Departamento de Instrucción Pública de Carolina del Norte*
Stephen Carey, *Departamento Escolar de Brunswick*
Brian Fuschetto, *Escuela Secundaria Lyndhurst*
Irene Lee, *Instituto Santa Fe*
Daniel Moix, *Universidad de Ouachitas*
Dianne O'Grady-Cunniff, *Escuela Secundaria Westlake*
Barbara Boucher Owens, *Universidad Southwestern*
Chris Stephenson, *Asociación de Maestros de Ciencias de la Computación*
Anita Verno, *Universidad Comunitaria Bergen*

Resumen Ejecutivo

Durante las últimas décadas, las computadoras han transformado el mundo y la fuerza laboral profundamente. Como resultado, las ciencias de la computación y las tecnologías necesarias ahora se encuentran en el corazón de nuestra economía y la forma en que vivimos nuestras vidas. Para ser ciudadanos bien educados en un mundo de computación intensiva y estar preparados para las carreras del siglo 21, los estudiantes deben tener un claro entendimiento de los principios y prácticas de las ciencias de la computación. Ningún otro tema abrirá tantas puertas en el siglo 21 como la ciencia de la computación, independientemente del campo de estudio u ocupación de la persona.

Como el informe Corriendo Vacío: El fracaso para enseñar Ciencias de la Computación en la era digital ([Http://csta.acm.org / Comunicaciones / sub / documents.html](http://csta.acm.org/Comunicaciones/sub/documents.html)) deja claro que el estado actual de la educación en ciencias de la computación, es inaceptable en momentos en que la computación está impulsando el crecimiento del empleo y el nuevo descubrimiento científico. Aproximadamente dos tercios de los cincuenta estados no tienen los estándares de ciencias de la computación para la educación secundaria. Aun cuando existen, los estándares del nivel K-8 a menudo confunden la ciencia de la computación y el uso de aplicaciones. A pesar de su importancia como campo académico, pocos estados cuentan las ciencias de la computación como una materia académica básica para graduarse. Las reglas para la certificación de maestros en ciencias de la computación varían ampliamente de un estado a otro y suelen ser ajenas a las necesidades de la enseñanza de esta disciplina. Estos son los fracasos nacionales y los que no nos podemos permitir en esta era digital.

Este documento proporciona los estándares integrales para la educación en ciencias de la computación de K- 12 diseñada para fortalecer la fluidez en la computación y la competencia en todas las escuelas primarias y secundarias. Está escrito en respuesta a la imperiosa necesidad de dar coherencia académica entre los cursos y el rápido crecimiento de la computación y la tecnología en el mundo moderno, junto con la necesidad de un público educado, que puede utilizar y construir la tecnología más eficaz para beneficio de la sociedad.

Estas normas proporcionan un marco de tres niveles para las ciencias de la computación. Los dos primeros niveles están dirigidos a los grados K- 6 y 6-9 respectivamente. Esperamos que los resultados del aprendizaje en el nivel 1 se aborden en el contexto de otras materias académicas. Los resultados del aprendizaje en el nivel 2, puede abordarse a través de otras materias o en cursos discretos de ciencias de la computación. El nivel 3 se divide en tres campos diferentes: Ciencias de la Computación en el mundo moderno, los principios de las ciencias de la computación, y Temas de Ciencias de la computación. Los estándares establecidos en Ciencias de la Computación en Mundo Moderno reflejan contenido que debe ser dominado por todos los estudiantes; los principios y Temas en Ciencias de la Computación son cursos diseñados para estudiantes interesados especialmente en ciencias de la computación y otras carreras de computación , ya sean encaminados a la universidad o no.

Estas recomendaciones no se hacen en el vacío. Somos conscientes de las serias limitaciones con las cuales están funcionando los distritos escolares y la dura batalla que enfrentan las ciencias de la computación, a la luz de otras prioridades educativas. Por lo tanto, concluimos este informe con una serie de recomendaciones que tienen el objetivo de

proporcionar ayuda para una evolución a largo plazo en las ciencias de la computación en las escuelas de K-12. Se han dado avances significativos desde que el Plan Modelo de Estudios de ACM para K-12 de Educación en Ciencias de la Computación fue publicado por primera vez en 2003 y revisado en 2006. Aún se necesitan muchos esfuerzos de seguimiento, sin embargo, para mantener el impulso que estos estándares generan. La formación docente, la innovación curricular, recursos para la enseñanza y la difusión son sólo algunos de estos retos.

Estos estándares de aprendizaje servirán como catalizadores para la adopción generalizada de la enseñanza de computación para todos los estudiantes de K-12. Le recomendamos que lea este documento y luego participe en el esfuerzo para poner en práctica estos estándares de manera que beneficie a usted y a la comunidad educativa de K-12. Busque información sobre las actividades que se desarrollan para apoyar la enseñanza de computación de K-12, en el sitio Web de la Asociación de Maestros de Ciencias de la Computación (csta.acm.org).

Contenido

Agradecimientos	i
Resumen ejecutivo	iii
1. Introducción.	1

2.	Ciencias de la computación como disciplina fundamental.	2
2.1	Las ciencias de la computación son intelectualmente importantes	2
2.2	Las ciencias de la Computación nos llevan a varias carreras.	3
2.3	Las ciencias de la computación enseñan a resolver problema.	3
2.4	Las ciencias de la computación sirven de apoyo y enlace a otras ciencias.	4
2.5	Las ciencias de la computación atraen a todos los estudiantes.	4
3.	Definiendo Términos.	5
4.	Organización de los resultados del aprendizaje: Niveles y guías.	7
4.1	Niveles.	7
4.2	Guías	9
4.2.1	Pensamiento Informático.	9
4.2.2	Colaboración.	10
4.2.3	Prácticas de Computación y programación.	11
4.2.4	Dispositivos informáticos y de comunicación.	11
4.2.5	La comunidad, impactos globales y éticos.	11
5.	Estándares integrales de ciencias de la computación para k–12.	12
5.1	Nivel 1: Las ciencias de la computación y yo.	12
5.2	Nivel 2: Las ciencias de la computación y la comunidad.	15
5.3	Nivel 3: aplicación de conceptos y creación de soluciones reales.	17
5.3.A	Las ciencias de la Computación en el mundo moderno.	18
5.3.B	Principios de las Ciencias de la computación.	20
5.3.C	temas en las ciencias de la computación.	22
5.3.C.1	AP ciencias de la computación A.	22
5.3.C.2	Cursos basados en proyectos.	22
5.3.C.3	Cursos que conducen a la certificación de la industria.	24
6.	Desafíos de Implementación.	25
7.	Llamado a la acción	26
8.	Actividades.	27
A.1.	Ejemplo de actividades para el nivel 1: ciencias de la computación y yo.	27
A.2.	Ejemplo de actividades para el nivel 2: ciencias de la computación y la comunidad.	32
A.3.	Ejemplo de actividades para el nivel 3: aplicación de conceptos y creación de soluciones reales.	44
A.4.	Ejemplo de actividades para el nivel 3c: temas en las ciencias de la computación.	50
A.5.	Recursos adicionales para el nivel 3.C.2 y 3.C.3	51
	Referencias.	54
	k–12 Cuadros escalonados de los estándares	55

K–12 Estándares Nacionales para las Ciencias de la Computación

1. Introducción

Hay una necesidad urgente para mejorar el nivel de comprensión pública de las ciencias de la computación como un campo académico y profesional, incluyendo las

diferencias en los sistemas de administración de gestión (AIG), la tecnología de la información (TI), las matemáticas y otras ciencias. Para funcionar en la

sociedad, cada ciudadano del siglo 21 debe comprender al menos los principios básicos de la ciencia de la computación. Un amplio compromiso y la rigurosa aplicación de los cursos de informática de K-12 crearán una amplia comprensión pública y ayudará a satisfacer las crecientes necesidades de la fuerza laboral internacional. Las escuelas primarias y secundarias tienen una oportunidad y responsabilidad única para afrontar esta necesidad.

La Ciencia de la computación es una disciplina establecida en los niveles universitarios y de postgrado. Se define mejor como "el estudio de las computadoras y de los procesos algorítmicos, incluidos sus principios, diseños de hardware y software, aplicaciones y su impacto en la sociedad." Desafortunadamente, los conceptos de las ciencias de la computación y los cursos en el currículo de K-12 no han mantenido el mismo ritmo de otras disciplinas académicas en los Estados Unidos. Como resultado, el público en general no está bien educado sobre la computación, hasta el punto que la nación se enfrenta a una escasez de científicos informáticos en todos los niveles y que probablemente continúe en el futuro previsible.

Estos estándares de las ciencias de la computación tienen como objetivo ayudar a solventar estos problemas. Proporcionan un marco en el que los departamentos estatales de educación y los distritos escolares pueden revisar sus planes de estudio para educar mejor a los jóvenes en este importante tema y así prepararlos mejor para la ciudadanía efectiva en el siglo 21.

El propósito de este documento es establecer el conocimiento de las ciencias de la computación y las habilidades que los estudiantes deben tener en todas las etapas de su aprendizaje, para permitirles avanzar en esta nueva economía global de la información. Define un conjunto de estándares de aprendizaje de las ciencias de la computación de K-12 y sugiere los pasos necesarios para su aplicación. Mediante la implementación de estos estándares, las escuelas pueden introducir los principios y metodologías de la informática a todos los estudiantes, ya sea que se dirijan a la universidad o a un centro de trabajo determinado. Los estándares establecidos en este documento, complementan los planes de estudio existentes para los grados de K-12 de ciencias de la computación ya establecidos, especialmente a los planes de estudio avanzados (AP) de ciencias de la

computación (AP, 2010) y las certificaciones profesionales IT.

Este documento delinea un conjunto básico de estándares de aprendizaje diseñados para proporcionar la base para un plan de estudios de ciencias de la computación completo y su implementación en los niveles de K-12. Para este fin, estas normas:

1. Introducir los conceptos fundamentales de las ciencias de la computación a todos los estudiantes, a partir de la escuela primaria;
2. presentar las ciencias de la computación en la escuela secundaria de manera que pueda cumplir con los créditos de graduación de las Ciencias de la computación, matemáticas o ciencias;
3. alentar a las escuelas a ofrecer cursos adicionales de ciencias de la computación a nivel secundario que permitan a los estudiantes interesados estudiar las facetas de las ciencias de la computación con mayor profundidad y prepararlos para el ingreso a la fuerza laboral o la universidad, y
4. aumentar la disponibilidad de las ciencias de la computación intensiva para todos los estudiantes, especialmente aquellos que son miembros de los grupos menos representados.

Nuestro objetivo es que estos estándares sean coherentes y comprensibles para los maestros, administradores y los responsables políticos. Por esta razón, nuestros debates en la primera etapa de este documento se centran en la actual proliferación y confusión de términos que a menudo hacen que esta disciplina sea incomprensible y mal definida para los que no son de este campo. También intentamos describir la importancia de la enseñanza de computación como parte del desarrollo intelectual de los estudiantes en todos los niveles enfatizando los vínculos entre las ciencias de la computación y la innovación en todas las disciplinas.

Todos los borradores de este informe han sido recibidos a través de la retroalimentación de muchas organizaciones y personas. Esperamos que este proyecto final reciba una difusión generalizada y el escrutinio constante de todos los que tienen intereses o experiencia en la educación de K-12. Con ese fin, hemos publicado estos estándares en el sitio Web de CSTA (<http://csta.acm.org>), así como en formato impreso.

Estos estándares son fundamentales para garantizar que los estudiantes alcancen el nivel necesario de conocimientos, habilidades y experiencia para prosperar en el mundo moderno. Mientras reconocemos que hay muchos obstáculos para adoptar la aplicación rigurosa de las ciencias de la computación en las aulas de K-12, no podemos, como nación, permitir que la situación actual continúe. Si no somos capaces de establecer dichos estándares o que no podemos implementarlos de manera efectiva, nuestros estudiantes se encontrarán sin preparación para trabajar en el sofisticado mundo tecnológico en el que deben competir.

2. Las Ciencias de la Computación como una Disciplina Núcleo

Nuestras vidas dependen de los sistemas informáticos y la gente que los mantiene dándonos confianza en la carretera y en el aire, ayudan a los médicos a diagnosticar y tratar los problemas de salud, y juegan un papel crítico en el desarrollo de muchos avances científicos. El comprender los fundamentos de las ciencias de la computación permite a los estudiantes a ser consumidores educados de tecnología y a ser creadores innovadores capaces de diseñar sistemas de computación para mejorar la calidad de vida de todas las personas.

Los niños de todas las edades aman la computación. Cuando se les da la oportunidad, los jóvenes estudiantes disfrutan de la maestría y la magia que ofrece la programación. Los estudiantes mayores se sienten atraídos por la combinación de arte, narrativa, diseño, programación y de la alegría obtenida de la creación de sus propios mundos virtuales. El combinar la computación con otras ciencias también provee oportunidades enriquecedoras en el aprendizaje. Por ejemplo, los estudiantes interesados en la música, pueden aprender acerca de la música y audio digital. Este campo integra la electrónica, varios niveles de matemáticas, teoría de la música, programación de computadoras y un buen oído para los sonidos hermosos, armoniosos, o simplemente interesantes.

La ciencia de la computación ha hecho posibles cambios profundos en la innovación y la imaginación, ya que facilita nuestros esfuerzos para solucionar problemas apremiantes (por ejemplo, la prevención o cura de las enfermedades, eliminar el hambre mundial). Expande la comprensión de nosotros mismos como sistemas

biológicos y nuestra relación con el mundo que nos rodea. Estos avances requieren de profesionales preparados que puedan utilizar el poder de la computación para resolver problemas complejos.

Ya no es necesario esperar hasta que los estudiantes estén en la universidad para introducir estos conceptos. Los estudiantes de hoy vivirán una vida muy influenciada por la computación y muchos trabajaran en áreas influenciadas directamente por la computación. Deben comenzar a solucionar problemas algorítmicos y los métodos de computación y sus herramientas de K-12.

2.1 Las Ciencias de la Computación son intelectualmente importantes

La invención de la computación en el siglo 20 fue un evento de "una vez en el milenio", comparado en importancia al desarrollo de la escritura o la prensa para imprimir. Las computadoras se diferencian fundamentalmente de otras invenciones tecnológicas en que aumentan directamente el pensamiento humano, en vez de las funciones musculares y sensoriales. Las computadoras tienen un enorme impacto en la manera en que vivimos, pensamos y actuamos. Es difícil sobreestimar su importancia en el futuro. Es más, algunos piensan que la verdadera revolución de las computadoras no va a ocurrir hasta que todos comprendan la tecnología lo suficiente para utilizarla de maneras innovadoras.

¿Porque es importante estudiar las ciencias de la Computación? Vivimos en un mundo digitalizado, computarizado, programable y para que tenga sentido necesitamos de las ciencias de la computación. Un ingeniero utilizando una computadora para diseñar un puente debe entender cómo se calcularon las estimaciones capacidad máxima y que tan confiables son. Un ciudadano educado utilizando una máquina para votar o para ofertar en una subasta debe tener un conocimiento básico de los algoritmos subyacentes relacionados con estas comodidades, así como los asuntos de seguridad y privacidad que surgen cuando la información es transmitida y guardada de manera digital.

Los Estudiantes de las ciencias de la computación aprenden razonamiento lógico, pensamiento algorítmico, diseño y estructura y resolución de problemas- todos los conceptos y habilidades que son valiosos más allá del aula de computación. Los estudiantes obtienen el conocimiento de los recursos necesarios para

implementar, analizar y desplegar una solución y cómo resolver los problemas del mundo real. Estas habilidades son aplicables en muchos contextos, desde las ciencias hasta la ingeniería, de las humanidades hasta las empresas y han permitido profundizar en estas y otras áreas. Las simulaciones por computadora son esenciales para comprender y descubrir las reglas fundamentales que rigen una amplia variedad de sistemas desde cómo las hormigas se reúnen alimentos a cómo se comportan los mercados de valores. La ciencia de la computación es una de las principales disciplinas que ayuda a comprender cómo funciona la mente humana, uno de los grandes desafíos intelectuales de todos los tiempos. Aunque muchas innovaciones computarizadas se encuentran delante de nosotros y la ciencia de la computación es una herramienta esencial para alcanzar nuestro vasto potencial.

2.2 Las Ciencias de la Computación nos llevan a múltiples carreras profesionales

La gran mayoría de las carreras en el siglo 21 van a requerir la comprensión de las ciencias de la computación. Muchos de los empleos que los estudiantes de hoy tendrán en 10 a 20 años no se han inventado todavía. Los profesionales de todas las disciplinas, desde artistas y animadores, a comunicadores y profesionales de la salud, trabajadores de fábricas, propietarios de pequeñas empresas y el personal de ventas al por menor, necesitan comprender la computación para ser productivo y competitivo en sus áreas. Thomas Friedman, en su libro más vendido, *The World is Flat* (2006), sostiene que la economía necesita más "Versatilistas," gente que tenga experiencia tanto en el dominio y en la tecnología. Las ciencias de la computación son el pegamento que permite a los versatilistas cruzar el puente de dominio específico y la innovación tecnológica.

Existe un vínculo inequívoco entre el éxito, la innovación y la ciencia de la computación. Películas como *Los increíbles* y *El Señor de los Anillos* ejemplifican el uso creativo de las nuevas técnicas de computación. Pero es difícil imaginar que algún campo no se ha visto afectado por la computación. Los Profesionales de las ciencias de la computación están resolviendo los desafíos de las ciencias, los negocios, el arte y las humanidades y la creando nuevas oportunidades laborales en todas estas áreas.

El estudio de las ciencias de la computación puede preparar a un estudiante para entrar en muchas carreras, tanto dentro como fuera de la computación. Los Profesionales de las ciencias de la computación nunca habían tenido más demanda como hoy en día. Los científicos de las ciencias de la computación están trabajando con expertos en otras áreas en el diseño y construcción de sistemas de computarizados que apoyan el funcionamiento de la sociedad moderna y que nos permite hacerle frente a los desafíos críticos que enfrenta nuestro mundo. Estos desafíos incluyen la energía global, la salud y el hambre mundial. Adicionalmente, las habilidades de computación son preferidas, si no necesarias, para trabajar en cualquier profesión.

2.3 Las Ciencias de la Computación nos enseñan a resolver problemas

Los Científicos de las ciencias de la computación trabajan en estrecha colaboración con los hombres de negocios, científicos, artistas y otros expertos para entender los problemas y definirlo de manera explícita para que se pueda representar en una computadora. Este proceso cooperativo requiere de personas con diferentes conocimientos y perspectivas para trabajar juntos para aclarar los problemas tomando en cuenta las prioridades y las limitaciones de los demás.

Las Ciencias de la Computación enseñan a los estudiantes a pensar en el proceso para resolver los problemas. En las ciencias de la computación, el primer paso para resolver un problema es nombrarlo con claridad y sin ambigüedades. Un científico de la computación que ayuda a diseñar un nuevo sistema para la consulta médica, por ejemplo, tiene que tener en cuenta el flujo continuo de trabajo, la privacidad del paciente, las necesidades de capacitación para el nuevo personal, la tecnología actual y futura, y por supuesto, el presupuesto. Una vez que el problema está bien definido, se debe crear una solución. Se debe seleccionar o construir el hardware y los dispositivos periféricos. Los Programas de computación se deben diseñar, escribir y probar. Los Sistemas y paquetes de software existentes pueden ser modificados e integrados en el sistema final. En todas las fases, el científico de la computación piensa en el uso reflexivo del tiempo en la computadora y los recursos compartidos. La construcción de un sistema es un proceso creativo que requiere también del pensamiento científico. Con cada corrección de un bug o el agregar una nueva

característica, se da la hipótesis de que el problema ha sido resuelto. Los experimentos se diseñan, se reúnen los datos, se analizan los resultados, y si la hipótesis es falsa, el ciclo se repite.

Un científico de la computación se preocupa por la robustez, facilidad de uso, facilidad de mantenimiento, y sobre todo la exactitud de las soluciones computarizadas para los problemas empresariales, científicos y de ingeniería. Estos problemas requieren de un análisis intenso y creatividad. ¿Cómo responderá el sistema si se va la luz, o si dos enfermeras intentan acceder al registro de un paciente al mismo tiempo, o si se cambia el sistema de la compañía de seguros, o si alguien introduce datos inesperados al sistema? La cooperación es nuevamente la clave. Los usuarios y los clientes tienen que pensar acerca de cómo se utilizar el sistema en la vida diaria y anticipar su uso futuro. Los científicos de las ciencias de la computación basan su formación y experiencia para afrontar los problemas y crear las mejores soluciones posibles.

2.4 Las Ciencias de la Computación Apoya y provee enlaces a otras ciencias

Para resolver los grandes problemas científicos del siglo 21, como lidiar con nuevas enfermedades y el cambio climático, necesitamos personas con diversas habilidades, capacidades y perspectivas. La secuencia del genoma humano en el año 2001 fue un hito de la biología molecular, que no habría sido posible sin un científico de las ciencias de la computación. Después que fragmentos cortos de ADN del genoma fueron secuenciados en laboratorios de biología, se utilizaron computadoras para encontrar la manera de unir los fragmentos. Este conocimiento está cimentando el camino para mejores métodos computacionales para detectar y curar las enfermedades como el cáncer, porque ahora estamos mejor capacitados para simular y por tanto, entender las mutaciones genéticas implicadas.

El cerebro humano es complejo y sorprendente. Sabemos, por ejemplo, que un niño puede reconocer sin esfuerzo una cara familiar desde puntos de vista diferentes y sin embargo, tenemos una mala comprensión de los mecanismos computacionales que el cerebro utiliza para resolver estos problemas. Inferir significado de las imágenes es una tarea computacional y los científicos de la computación y los neurocientíficos están trabajando juntos para encontrar la manera de

construir computadoras que pueden procesar imágenes y al final, cómo podemos entender mejor la inteligencia misma. El uso de modelos y la simulación, la visualización y manejo de grandes conjuntos de datos ha propiciado el surgimiento de un nuevo campo que une la ciencia, tecnología, ingeniería y la matemática computacional. Este campo integra muchos aspectos de la ciencia de la computación, tales como el diseño de algoritmos y gráficos con su aplicación en las ciencias.

En las clases de ciencias, los estudiantes utilizan un sofisticado software de simulación para formar moléculas y para que los procesos geológicos cobren vida. La Creación de programas de computadora que modelen el comportamiento permite a los científicos generar resultados y probar teorías que son imposibles de probar en el mundo físico. Los avances en la predicción del tiempo, por ejemplo, dependen en gran medida de los modelos y simulación por computadora. Los métodos computacionales también han transformado campos como la estadística y la química. Los científicos que pueden entender y contribuir a la innovación tecnológica tienen una gran ventaja. Una buena capacitación para los futuros científicos debe incluir, por tanto una base sólida en ciencias de la computación.

2.5 Las Ciencias de la Computación Involucran a todos los Estudiantes

La ciencia de la computación se aplica virtualmente a casi todos los aspectos de la vida, de modo que pueda ser vinculado fácilmente a los innumerables intereses del estudiante. Por ejemplo, los estudiantes que están fascinados con tecnologías específicas como los teléfonos celulares pueden tener una pasión innata para el diseño visual, el entretenimiento digital o ayudar a la sociedad. Los profesores de ciencias de la computación de K-12 pueden promover en los estudiantes sus intereses, pasiones y el sentido de compromiso con el mundo que les rodea, ofreciendo oportunidades para resolver problemas computacionales relevantes a sus propias experiencias.

La excelencia en la educación de las ciencias de la computación se basa en las prácticas equitativas que maximizan el potencial de aprendizaje de todos los estudiantes. Las oportunidades de aprendizaje de las ciencias de la computación deben conformarse de manera que conecten el canon del contenido de ciencias de la computación, proporcionada en los estándares

curriculares, con las experiencias vividas por diversos estudiantes. Las prácticas equitativas en la educación en ciencias de la computación que conectan a los estudiantes con el plan de estudios incluyen:

- Todos los estudiantes deben tener acceso riguroso y culturalmente significativo a las ciencias de la computación y mantener grandes expectativas para interactuar con el plan de estudios.
- Las diversas experiencias, creencias y formas de conocer las ciencias de la computación deben ser reconocidos, incorporados y celebrados en el salón de clases.
- La integración de diferentes interpretaciones, estrategias y soluciones seguras para la computación mejoran los debates en clase y profundizan la comprensión.
- Los recursos necesarios para la enseñanza y el aprendizaje de las ciencias de la computación deben asignarse de manera equitativa entre los grupos de estudiantes, las aulas y las escuelas.
- Los grupos de aprendizaje en el aula deben fomentar un ambiente en el que todos los estudiantes sean escuchados, respetados y vistos como contribuyentes valiosos al proceso de aprendizaje.
- Las reflexiones docentes que se llevan a cabo actualmente, sobre los sistemas de creencias, suposiciones y prejuicios apoyan el desarrollo de prácticas de enseñanza equitativas.

Pedagógicamente, la programación de computadoras tiene la misma relación con el estudio de las ciencias de la computación como tocar un instrumento al estudio de la música o la pintura a estudiar arte. En cada caso, incluso poca experiencia ayuda en gran medida a apreciar y comprender toda la vida, aunque el estudiante no continúe programando, tocando, o pintando una vez que sea adulto. Aunque convertirse en un programador experto, violinista o pintor al óleo exige mucho tiempo y talento, todavía queremos exponer a todos los estudiantes a las alegrías de ser creativo. El objetivo de la enseñanza de la computación debe ser lograr que el mayor número posible de estudiantes participen con entusiasmo en cada asignación. Podemos proveer a los estudiantes las herramientas necesarias para diseñar y escribir programas para controlar sus teléfonos celulares

o robots, crear simulaciones de física y biología o componer música. Los estudiantes querrán aprender a utilizar los condicionales, circuitos, parámetros y otros conceptos fundamentales sólo para lograr realizar cosas emocionantes.

En un campo que avanza rápidamente, como ser la ciencia de la computación, todos tenemos el desafío de mantenernos al día con nuestros compañeros y los estudiantes. La tecnología cambia rápidamente y los estudiantes muchas veces tienen más facilidades que los profesores conocer la última tecnología. Ningún maestro debe ser aprensivo de aprender de sus estudiantes. El verdadero aprendizaje involucra que todos en el aula vivan con una sensación de asombro y expectativa.

Sabemos que enseñar ciencias de la computación implica desafíos únicos y que nadie tiene todas las respuestas. Los recursos compilados en el sitio Web CSTA en <http://csta.acm.org/WebRepository/WebRepository.html> ofrecen una amplia colección de recursos para profesores. Estos recursos se han encontrado útiles para motivar, involucrar e interesar a los estudiantes. No todos se podrán aplicar en todas las aulas, pero creemos que muchos contienen sugerencias útiles y variadas para inspirar a los estudiantes y profesores por igual.

3. Definiendo la Terminología

Las ciencias de la computación cambian constantemente. Las nuevas maneras de pensar y las nuevas tecnologías continúan expandiendo nuestra comprensión de lo que los científicos de la computación necesitan saber. Esto ha dado lugar a un importante debate sobre la definición de las ciencias de la computación.

Antes de hablar de los estándares curriculares de K-12, primero hay que aclarar el contexto en el que se establecen los estándares y aclarar la confusión que gira en torno a la proliferación de los términos utilizados para describir los distintos tipos de educación en computación.

Para los educadores de secundaria, tal vez la confusión más grande surge, cuando se trata de distinguir entre las tres áreas más comunes de la educación de computación que se ofrece en las escuelas. Si bien a cada una de estas áreas se ha conocido por varios nombres, para los propósitos de esta discusión las llamamos:

- Educación Tecnológica
- Información Tecnológica y
- Ciencias de la Computación.

La Tecnología Educativa se puede definir como el uso de computadoras durante todo el plan de estudios, o más específicamente, el uso de la tecnología de computación (hardware y software) para aprender sobre otras disciplinas. Por ejemplo, el profesor de ciencias puede utilizar simulaciones por computadora ya existentes para proporcionar a los estudiantes una mejor comprensión de los principios específicos de física, o un profesor de inglés puede utilizar el software de procesamiento de textos para ayudar a los estudiantes a mejorar sus habilidades de edición y revisión. Mientras que la tecnología educativa se preocupa por el uso de estas herramientas, la ciencia de la computación se ocupa de diseñar, crear, probar, modificar y verificar estas herramientas.

La Tecnología de la información (IT) es "el uso adecuado de las tecnologías por medio de las cuales las personas manipulan y comparten información en sus diversas formas." Mientras que la Tecnología de la Información consiste en aprender acerca de las computadoras, esta hace énfasis en la tecnología en sí. Los especialistas en tecnología de la información asumen la responsabilidad de seleccionar productos de hardware y software adecuados, integrando estos productos con las necesidades de organización e infraestructura y la instalación, personalización, y mantenimiento de esos recursos. Por lo tanto, los cursos de tecnología de la información se centran en:

- instalar, asegurar y administrar las redes de computadoras;
- instalar, mantener y personalizar el software;
- administrar y asegurar la información en los mundos virtuales y físicos;
- administrar los sistemas de comunicación;
- diseñar, implementar y administrar los recursos de la Web; y
- desarrollar y administrar los recursos multimedia y otros medios digitales.

IT es un campo aplicado de estudio, impulsado por los beneficios prácticos de su conocimiento, mientras que las ciencias de la computación añaden dimensiones tanto

prácticas como matemáticas. Algunas de las dimensiones prácticas de las ciencias de la computación se comparten con IT, como el trabajo con textos, gráficos, sonido y video. Mientras el aprendizaje IT se centra en cómo utilizar y aplicar estas herramientas, las ciencias de la computación tiene que ver con el aprendizaje de cómo estas herramientas se han diseñado y por qué funcionan. Las ciencias de la computación e IT tienen mucho en común, pero ninguno de los dos puede ser sustituido por el otro. Por ejemplo, la complejidad de los algoritmos es una idea fundamental en las ciencias de la computación, pero probablemente no aparece en un plan de estudios de IT.

Las Ciencias de la Computación, por el contrario, se extienden por una amplia gama de esfuerzos de computación, desde los fundamentos teóricos de la robótica, la visión artificial, los sistemas inteligentes y la bioinformática. El trabajo de los científicos de la computación se concentra en tres áreas:

- Diseño e implementación de software,
- Desarrollo de maneras efectivas para resolver los problemas de computación, y profundamente, prepararse para la Universidad o la fuerza laboral, y
- Elaborando nuevas formas para utilizar las computadoras.

Para los propósitos de este documento, nos apoyamos en gran medida de la definición de ciencias de la computación proporcionada en el Modelo curricular original ACM / CSTA para Ciencias de la Computación de K-12, ya que creemos que esta definición tiene la relevancia más directa a la enseñanza de las ciencias de la computación en la escuela secundaria.

“Ciencias de la Computación (CS) es el estudio de las computadoras y de los procesos algorítmicos, incluyendo sus principios, sus diseños de hardware y software, sus aplicaciones y su impacto en la sociedad.”

Una comprensión básica de las ciencias de la computación es ahora el principal ingrediente para preparar a los graduandos de secundaria para la vida en el Siglo 21 y los objetivos de cualquier curso intensivo de ciencias de la computación deben ser:

- introducir los conceptos fundamentales de las ciencias de la computación a todos los estudiantes, desde los niveles primarios,

- presentar las ciencias de la computación al nivel secundario de manera que sea accesible y digna de un crédito académico en el plan de estudios (por ej., matemáticas o ciencias),
- ofrecer cursos de ciencias de la computación adicionales en el nivel secundario, que permitan a los
- aumentar el conocimiento de las ciencias de la computación en todos los estudiantes, especialmente en aquellos que son miembros de los grupos históricamente minoritarios.

Otros dos términos que aparecen a menudo en los debates sobre educación en computación son Alfabetización de las Tecnologías de la Información y Fluidez en Tecnologías de la Información. Un estudio de la Academia Nacional, publicado en 1999, define la fluidez IT como algo más amplio que la alfabetización IT. Mientras que la alfabetización de IT es la capacidad de utilizar la tecnología moderna en su propio campo, la noción de la fluidez IT añade la capacidad de aprender y

4. Organización de los Resultados del Aprendizaje: Niveles y Ramas

Proponemos un modelo de tres niveles para las ciencias de la computación de K-12 que aborda las necesidades del presente y futuro basándose en las lecciones del pasado. Se enfoca los conceptos fundamentales con los siguientes objetivos generales:

1. El plan de estudios debe preparar al estudiante para comprender la naturaleza de las ciencias de la computación y su lugar en el mundo moderno.
2. Los estudiantes deben comprender que las ciencias de la computación entrelazan conceptos y destrezas.
3. Los estudiantes deben ser capaces de utilizar sus habilidades de ciencias de la computación (especialmente el pensamiento computacional) en sus actividades de resolución de problemas y en otras materias.
4. Los estándares de las ciencias de la computación deben complementar el plan de estudios de IT y de AP en las escuelas donde se imparte actualmente.

utilizar las nuevas tecnologías independientemente, a medida que evolucionan durante la vida profesional de la persona. Por otra parte, la fluidez también incluye el uso activo del pensamiento computacional (incluyendo la programación) para resolver problemas, mientras que la alfabetización IT no. El pensamiento computacional es un método para resolver problemas de manera que pueda ser implementado con una computadora. Esto implica el uso de conceptos como abstracción, la recursividad y la iteración para procesar y analizar los datos y crear artefactos reales y virtuales.

La fluidez IT se propuso como el estándar mínimo que todos los estudiantes universitarios deben lograr para el momento de su graduación. La mayoría de universidades han puesto en práctica estos estándares o estándares similares y están esperando que sus graduandos los alcancen. En este documento, apoyamos firmemente que este estándar mínimo debería aplicarse también en los niveles de K-12.

Si estos estándares se implementan ampliamente y que se cumplen estos objetivos, los graduados de la escuela secundaria estarán preparados para ser usuarios expertos y críticos de las computadoras, así como diseñadores y constructores de aplicaciones de computación que afectarán cada aspecto de la vida en el Siglo 21.

4.1 Niveles

Los estándares de CSTA para las ciencias de la computación de K-12, se basan en un modelo en el que cada uno de los tres niveles representa un conjunto específico de grados y cursos. El Nivel 1 proporciona los estándares de

Aprendizaje para los estudiantes en los grados de K-6, el Nivel 2 proporciona los estándares de aprendizaje para los estudiantes en los grados de 6-9 y el Nivel 3 proporciona los estándares de aprendizaje para los estudiantes en cada uno de los tres campos diferenciados en los grados 9-12. (Observamos que los límites especificados para cada nivel varían de escuela a escuela.) La estructura general de este modelo se muestra en la Figura 1.

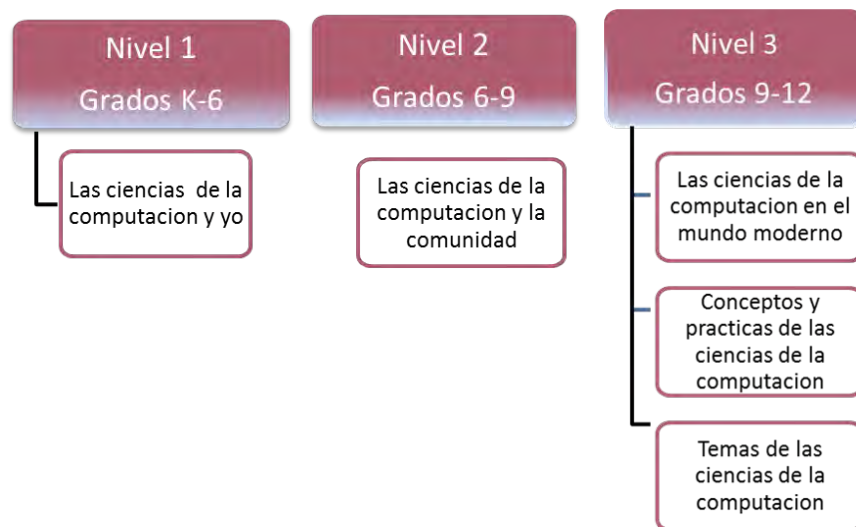


Figura 1.
Estructura organizativa para
Los estándares de las ciencias de la computación

Nivel 1 Ciencias de la Computación y Yo (se recomienda para los grados K-6): Se introducen los conceptos fundamentales de las ciencias de la computación a los alumnos de primaria, integrando las competencias básicas en la tecnología con las ideas simples sobre el pensamiento computacional. Las experiencias de aprendizaje creadas a partir de estos estándares deberían ser inspiradoras y atractivas, ayudando a los alumnos a ver las ciencias de la computación como una parte importante de su mundo. Estos deben ser diseñados enfocándose en el aprendizaje activo, la creatividad y la exploración y a menudo pueden estar incrustados dentro de otras áreas curriculares como las ciencias sociales, artes del lenguaje, matemáticas y ciencia.

Nivel 2 Ciencias de la Computación y de la comunidad (se recomienda para los grados 6-9): Estudiantes de la escuela media/ secundaria utilizando el pensamiento computacional como herramienta para resolver problemas. Ellos comienzan a apreciar la omnipresencia de la computación y las formas en que las ciencias de la computación facilitan la comunicación y la colaboración. Los estudiantes comienzan a experimentar el pensamiento computacional como un medio para abordar temas relevantes, no sólo para ellos, sino para el mundo que les rodea. Las experiencias de aprendizaje creadas a partir de estos estándares deben ser relevantes para los estudiantes y deben promover su percepción de sí mismos como solucionadores proactivos de problemas. Deben ser diseñadas con un enfoque en el

aprendizaje activo y la exploración y pueden enseñarse en cursos explícitos de ciencias de la computación o encontrarse incrustados en otras áreas curriculares como las ciencias sociales, artes de lenguaje, matemáticas y ciencia.

Nivel 3 Aplicando conceptos y creando soluciones para el mundo real (recomendado para los grados 9–12): El Nivel 3 se divide en tres campos diferenciados, cada uno de los cuales se centra en las diferentes facetas de las ciencias de la computación como una disciplina. A lo largo de estos cursos, los estudiantes pueden dominar conceptos de ciencias de la computación más avanzados y aplicarlos para desarrollar artefactos virtuales y del mundo real. Las experiencias de aprendizaje creadas a partir de estos estándares deben centrarse en la exploración de los problemas del mundo real y la aplicación del pensamiento computacional para desarrollar soluciones. Estos deben ser diseñados enfocándose en el aprendizaje colaborativo, gestión de proyectos y una comunicación eficaz. EL Nivel 3 incluye los siguientes cursos:

Nivel 3A: Ciencias de la Computación en el mundo moderno (se recomienda para los grados 9 o 10): Este curso está recomendado para todos los estudiantes. Su objetivo es consolidar en los estudiantes la comprensión de los principios y prácticas de las ciencias de la computación para que puedan tomar decisiones informadas y utilizar las herramientas y técnicas en cualquier carrera que decidan cursar. También deben ser

conscientes de la amplitud de las ciencias de la computación y su influencia en casi todos los aspectos de la vida moderna. Por último, deben entender el impacto social y ético de sus opciones al utilizar la tecnología de las computadoras en su trabajo y su vida personal y las decisiones que ya se han sido tomados por los que se desarrollan las tecnologías.

Nivel 3B: *Conceptos y Prácticas de las Ciencias de la computación* (se recomienda para los grados 10 u 11): Este curso es un estudio más profundo de las ciencias de la computación y su relación con otras disciplinas y contiene una gran cantidad de resolución de problemas algorítmicos y actividades relacionadas. Una forma de realizar este curso es a través del curso Principios de las ciencias de la computación (www.apcsprinciples.org). Los estudiantes deben completar este curso con una clara comprensión de la aplicación del pensamiento computacional a los problemas del mundo real. También deberían haber aprendido a trabajar en equipo para resolver un problema y utilizar herramientas modernas de colaboración durante ese trabajo.

Nivel 3C: *Temas de Ciencias de la Computación* (se recomienda para los grados 11 o 12): Se trata de una asignatura optativa que proporciona un estudio a profundidad de un área particular de computación. Esto puede ser, por ejemplo, puede ser el curso AP ciencias de la computación A (AP, 2010) que ofrece un estudio a profundidad de programación Java. Alternativamente, se

puede ofrecer un curso basado en proyectos centrados en un solo aspecto de la computación o un curso que lleve a la certificación profesional en computación.

4.2 Ramificaciones

Casi desde su inicio, la ciencia de la computación se ha visto obstaculizada por la percepción de que se centra exclusivamente en la programación. Este error ha sido particularmente perjudicial en los grados K-12, donde a menudo ha llevado a cursos cuyo alcance ha sido limitado y que han sido percibidos negativamente por los estudiantes. También alimenta a otras percepciones desafortunadas de las ciencias de la computación como que es una actividad solitaria, desconectada del resto del mundo y de poca relevancia para los intereses y preocupaciones de los estudiantes.

Abordamos estos problemas distinguiendo cinco cadenas complementarias y esenciales a lo largo de los tres niveles en estos estándares. Estas ramas son: el pensamiento computacional, la colaboración, práctica de computación; computadoras y dispositivos de comunicación y los impactos a la comunidad, globales y éticos. Estas ramas no sólo demuestran la riqueza de las ciencias de la computación sino que también ayudan a

organizar el material para los estudiantes, para que puedan percibir las ciencias de la computación de manera más atractiva y relevante, más que una actividad solitaria.

La Figura 2 muestra gráficamente estas ramas.



Figura 2.
Ramas en los estándares de las Ciencias de la Computación }

Las siguientes subsecciones discuten cómo estas cinco ramas pueden ayudar a los estudiantes a enriquecer su conocimiento y dominio de las ciencias de la computación durante sus años de formación. Discusiones más detalladas aparecen más adelante en este informe.

4.2.1 Pensamiento Computacional

Creemos que el pensamiento computacional (CT) se puede utilizar en todas las disciplinas para resolver problemas, sistemas de diseño, crear nuevos conocimientos y mejorar la comprensión del poder y las limitaciones de la computación en la edad moderna. El estudio del pensamiento computacional permite a todos los estudiantes de conceptualizar mejor, analizar y resolver problemas complejos mediante la selección y aplicación de estrategias y herramientas adecuadas, tanto virtuales como reales.

La educación de K-12 es un entorno complejo, altamente politizado, donde múltiples prioridades competitivas, ideologías, pedagogías y las ontologías compiten por la atención. También es objeto de una amplia diversidad de expectativas, un escrutinio intenso y recursos decrecientes. Cualquier esfuerzo para lograr un cambio

sistémico en este entorno requiere un profundo conocimiento de estas realidades. Un debate apasionado sobre la naturaleza de las ciencias de la computación o el pensamiento computacional puede proporcionar el estímulo intelectual para aquellos que trabajan en el campo de la computación. Sin embargo, incorporar el pensamiento computacional en los grados K-12 requiere un enfoque práctico, basado en una definición operativa.

El desarrollo de un enfoque de pensamiento computacional adecuado para estudiantes de K-12 es especialmente difícil en vista que hasta ahora no hay un amplio consenso sobre la definición del *pensamiento computacional*. A los efectos de este documento, nos basamos en la definición desarrollada durante una serie de talleres organizado por la Sociedad Internacional para la Tecnología en la Educación (ISTE) y la Asociación de Maestros de Ciencias de la Computación (CSTA) e informado por Barr y Stephenson (2011):

"La CT es un método para resolver problemas de manera que puedan ser implementados con una computadora. Los estudiantes no son solo usuarios de la herramienta sino constructores de ellas. Ellos utilizan un conjunto de

conceptos, tales como la abstracción, la recursividad y la iteración, para procesar y analizar los datos y crear artefactos reales y virtuales. CT es una metodología de resolución de problemas que se puede automatizar, transferir y aplicar a través de todos las clases. El poder del pensamiento computacional es que se aplica a cualquier otro tipo de razonamiento. Permite realizar todo tipo de cosas: física cuántica, biología avanzada, sistemas computarizados humanos, el desarrollo de herramientas de computación útiles”.

El Pensamiento computacional es una metodología de resolución de problemas que pueden entrelazar las ciencias de la computación con todas las disciplinas, proporcionando métodos distintivos de análisis y desarrollo de soluciones a los problemas que se pueden resolver computacionalmente. Con su enfoque en la abstracción, la automatización y el análisis, el pensamiento computacional es un elemento central de la disciplina más amplia de las ciencias de la computación y por eso se entreteje a través de estos estándares de las ciencias de la computación en todos los niveles de aprendizaje de K-12.

4.2.2 Colaboración

La ciencia de la computación es una disciplina intrínsecamente colaborativa. Rara vez se hacen progresos importantes en las ciencias de la computación por una persona trabajando sola. Normalmente, los proyectos de computación implican grandes equipos de profesionales de la computación que trabajan juntos para diseñar, codificar, probar, depurar, describir y mantener el software a través del tiempo. Las nuevas metodologías de programación como la programación en pares hacen hincapié en la importancia del trabajo en equipo. Además, los equipos de desarrollo que trabajan con expertos específicos en cada disciplina garantizan las soluciones computacionales apropiadas, eficaces y eficientes. El desarrollo de habilidades de colaboración es por tanto una parte importante de los estándares nacionales de las ciencias de la computación de K-12.

En la escuela primaria, los estudiantes pueden comenzar a trabajar en cooperación con otros estudiantes y profesores utilizando la tecnología. Ellos aprenden a reunir información y comunicarse con otros utilizando una variedad de dispositivos de comunicación tradicionales y móviles. También aprenden a utilizar los recursos en línea y participar en actividades de

resolución de problemas en colaboración. Estas actividades de colaboración continúan en la escuela media, donde los estudiantes aplican multimedia y herramientas de productividad para los ejercicios de aprendizaje en grupo. En la escuela secundaria, los estudiantes a mejoran sus habilidades de colaboración, participando en equipos para resolver problemas de software que son relevantes para su vida diaria. Las habilidades aprendidas en este nivel pueden incluir trabajo en equipo, crítica constructiva, planificación y gestión de proyectos y comunicación en equipo, todas se consideran como habilidades necesarias en el siglo 21 (ver Asociación para Habilidades del Siglo 21 en p21.org).

4.2.3 Practica de Computación y Programación

El uso de herramientas de computación es parte esencial de la educación de las ciencias de la computación en todos los niveles. Mientras esto se conoce tradicionalmente como "Tecnología de la Información," es imposible separar la IT de las otras cuatro ramas de las ciencias de la computación. Por lo tanto, la práctica de computación en los niveles de K-12 debe incluir la capacidad de crear y organizar páginas web, explorar el uso de la programación en la solución de problemas, selección del archivo y los formatos de base de datos apropiados para un problema computacional y el uso apropiado de Programación de Aplicaciones Interfaces (APIs), herramientas de software y bibliotecas para ayudar a resolver problemas algorítmicos y computacionales.

En el momento en que llegan a la escuela secundaria y seleccionan carreras o trayectorias educativas, los estudiantes deben estar bien informados sobre sus opciones para tomar decisiones inteligentes. Por lo tanto, los estudiantes de K-12 deben conocer la variedad de carreras que existen en el área de computación o a las que esta hace una contribución significativa. Ya que muchas veces se da la percepción errónea de que la computación es solo programación, es especialmente importante que los alumnos comprendan la amplia gama de conocimientos que las ciencias de la computación proporcionan a través de todos los campos y disciplinas.

4.2.4 Computadoras y Dispositivos de Comunicación

Los estudiantes de todos los niveles de K-12 deben comprender los elementos de los modernos dispositivos de computación y comunicación y sus redes. También

deben entender cómo la Internet facilita la comunicación global y como ser buen ciudadano en Internet. Los estudiantes también deben utilizar la terminología adecuada y precisa al comunicarse acerca de la tecnología

A nivel de la escuela primaria, se introduce a los estudiantes muchos dispositivos y medios de comunicación que les pueden ayudar con sus actividades de aprendizaje, tanto dentro de las ciencias de la computación como en otras disciplinas. Los estudiantes de educación media comienzan a discriminar entre los diferentes dispositivos y sus usos. También deben ser capaces de describir los componentes básicos de las computadoras y sus redes. Por ejemplo, deben entender la organización de las páginas web, direcciones URL y los motores de búsqueda. Los estudiantes de secundaria deben entender los dispositivos computacionales más detalladamente, aprendiendo a abstraer ideas acerca de componentes específicos (por ejemplo, de entrada, salida, procesadores y bases de datos) y su rol en el espectro de la computación. Los estudiantes también deben entender por qué un compilador traduce el software a una forma ejecutable por la máquina.

4.2.5 Impactos Globales, Éticos y de la Comunidad

El uso ético de las computadoras y las redes es un aspecto fundamental de las ciencias de la computación en todos los niveles y debe ser visto como un elemento esencial del aprendizaje y la práctica. Tan pronto como los estudiantes comienzan a utilizar la Internet, deben aprender las normas para su uso ético. Los principios de la privacidad personal, seguridad de redes, licencias de software y los derechos de autor se deben enseñar a un nivel adecuado para preparar a los estudiantes para convertirse en ciudadanos responsables en el mundo moderno. Los estudiantes deben ser capaces de tomar decisiones informadas y éticas entre los diversos tipos de software tales como código abierto y propietario y comprender la importancia de adherirse a los acuerdos de uso y las licencias. Los estudiantes también deben ser capaces de evaluar la fiabilidad y exactitud de la información que reciben a través de Internet.

Las computadoras y las redes son un fenómeno multicultural que afectan a la sociedad en todos los niveles. Es esencial que los estudiantes de K-12 comprendan el impacto de las computadoras en la comunicación internacional. Deben aprender la

diferencia entre los comportamientos apropiados e inapropiados de redes sociales. También deben apreciar el papel de la tecnología de adaptación en la vida de las personas con diversas discapacidades.

La Computación, como todas las tecnologías, tiene un profundo impacto en toda la cultura donde se encuentra. La distribución de los recursos de computación en una economía global, plantea problemas de equidad, acceso y poder. Los valores sociales y económicos influyen en el diseño y desarrollo de las innovaciones computacionales. Los estudiantes deben estar preparados para evaluar los impactos, tanto positivos como negativos de las computadoras en la sociedad y a identificar el grado en que los problemas de acceso (quien tiene acceso, quien no y quién toma las decisiones sobre el acceso) impactan nuestras vidas.

5. Estándares Integrales de las Ciencias de la Computación para los niveles de K-12

A partir de las interpretaciones y contextos descritos en las secciones anteriores, esta sección define nuevos estándares para la enseñanza de las ciencias de la computación de K-12, presentándolos en formato de objetivos de aprendizaje que identifican los conceptos específicos de las ciencias de la computación y las habilidades que los estudiantes deben alcanzar en cada uno de los tres niveles (grados K-6, 6-9 y 9-12).

5.1 Nivel 1: Las Ciencias de la Computación y Yo (L1)

Estos estándares introducen los conceptos primarios de las ciencias de la computación integrando las competencias básicas de tecnología con los conceptos básicos sobre el pensamiento computacional. Las experiencias de aprendizaje creadas a partir de estos estándares deberían ser inspiradoras y atractivas, ayudando a los alumnos a ver la computación como parte importante de su mundo. Estos deben ser diseñados enfocándose en el aprendizaje activo, la creatividad y la exploración, típicamente pueden encontrarse dentro de otras áreas curriculares como las ciencias sociales, artes del lenguaje, matemáticas y ciencia.

Es importante reconocer el impacto significativo que la exposición temprana a las cinco ramas, descritas en el apartado anterior, puede tener a medida que los

estudiantes progresan hacia programas de ciencias de la computación más avanzados. Los estudiantes de K-6 necesitan aprender que las herramientas computacionales pueden ser utilizadas para ayudar a resolver problemas, comunicarse con los demás, acceder y organizar la información por sí mismos o en colaboración con otros. Tienen que comenzar a examinar los problemas relativos a validar y valorar las diferentes fuentes de información. También deben aprender a ser ciudadanos responsables en el mundo digital cambiante. La utilización ética y segura de las computadoras y las redes debe enseñarse hasta a los estudiantes más jóvenes.

Estamos de acuerdo con los profesores que creen que los alumnos de esta edad deben comenzar a pensar algorítmicamente como una estrategia para resolver problemas. Por lo tanto, tiene sentido desarrollar más estrategias de enseñanza que estimulen a los estudiantes a participar en el proceso de visualizar o actuar un algoritmo. Los experimentos pioneros de Seymour Papert en la década de 1970 corroboran esta creencia y su trabajo seminal provoca una Lluvia de ideas y planes de estudios relacionados que proporcionan muchos ejemplos de cómo los estudiantes de primaria pueden involucrarse en el pensamiento algorítmico. Este compromiso puede llevarse a cabo con o sin el uso de dispositivos de computación como en los siguientes ejemplos:

- Encontrar el camino en un laberinto (gráficos Tortuga y robótica),
- un perro recobrando una bola,
- hornear galletas,
- ir de la casa a la escuela,
- construir un Castillo de arena, y
- ordenar alfabéticamente una lista de palabras.

Cualquier actividad diseñada para introducir a los estudiantes de los niveles de K-6 a las cinco ramas pueden incluir individuos trabajando solos o en colaboración con sus amigos. El concepto de "equipo" es uno que se puede introducir en cualquier grado. Los estudiantes que trabajan juntos pueden encontrar múltiples soluciones a los problemas utilizando los recursos disponibles a través de búsquedas en línea y después crear presentaciones multimedia para demostrar la solución elegida. A medida que desarrollan estas

habilidades pueden comenzar a reconocer la variedad herramientas y programas que pueden utilizar para comunicarse entre sí, sus maestros, sus padres e incluso los estudiantes de lugares lejanos. Ellos comenzarán a pensar cómo las computadoras y los programas trabajan juntos para hacer que las cosas sucedan como lo hacen.

Como estudiantes de K-6 están expuestos a las varias facetas de la comunicación y las redes sociales a través de la tecnología, es esencial que aprendan la forma segura y ética de utilizar estas herramientas y las posibles repercusiones de su uso inadecuado. Pueden explorar las diferentes maneras en que los dispositivos de computación y la tecnología afectan sus vidas y la sociedad que les rodea.

Los siguientes son los estándares que todos los estudiantes en los niveles de K-6 deben cumplir en las cinco ramas:

Pensamiento Computacional: (CT)

Grados K-3 (N1:3.CT)

El estudiante será capaz:

1. Utilizar recursos tecnológicos (ej. rompecabezas, programas de razonamiento lógico) para resolver problemas apropiados a cada edad.
2. Utilizar herramientas de escritura, cámaras digitales y dibujos para ilustrar ideas e historias paso a paso.
3. Comprender a organizar (clasificar) la información de manera útil, como organizar los estudiantes por su cumpleaños, sin utilizar la computadora.
4. Reconocer que el software es creado para controlar las operaciones de las computadoras.
5. Demostrar cómo se pueden utilizar los 0s y 1s para representar la información

Grados 3-6 (N1:6.CT)

El estudiante será capaz de:

1. Comprender y utilizar los pasos básicos para resolver problemas algorítmicos (ej., planteamiento y exploración del problema, examinar las muestras, diseño, implementación y pruebas).

2. Desarrollar el entendimiento de un algoritmo simple (ej., búsqueda, secuencia de los eventos u organización) realizando ejercicios sin utilizar la computadora.
3. Demostrar como una cadena de bits se puede utilizar para representar la información alfanumérica
4. Describir como una simulación se puede utilizar para resolver un problema.
5. Haga una lista de sub-problemas a considerar al abordar un problema mayor
6. Comprender las conexiones entre las ciencias de la computación y otras áreas.

Colaboración (CL)

Grados K–3 (N1:3.CL)

El estudiante será capaz de:

1. Recopilar información y comunicarse electrónicamente con los demás con el apoyo de maestros, miembros de la familia o compañeros estudiantes.
2. Trabajar en cooperación y colaboración con sus compañeros, maestros y otras personas que utilizan la tecnología

Grados 3–6 (N1:6.CL)

El estudiante será capaz de:

1. Uso de herramientas tecnológicas de productividad (por ejemplo, el procesamiento de textos, hoja de cálculo, software para presentaciones) para escritura individual y colaborativa, comunicación y actividades de publicación.
2. Utilizar recursos de Internet (por ejemplo, correo electrónico, conversaciones en línea, entornos web colaborativos) para participar en actividades colaborativas para resolver problemas con el propósito de desarrollar soluciones o productos.
3. Identificar formas en que el trabajo en equipo y la colaboración pueden apoyar la resolución de problemas y la innovación.

Practica de Computación y Programación (CPP)

Grados K–3 (N1:3.CPP)

El estudiante será capaz de:

1. Utilizar los recursos tecnológicos para realizar investigaciones apropiadas para cada edad.
2. Utilizar los recursos multimedia de desarrollo apropiados (por ejemplo, libros interactivos y software educativo) para apoyar el aprendizaje durante la duración del plan de estudios.
3. Crear productos multimedia de desarrollo apropiados con el apoyo de maestros, miembros de la familia o compañeros.
4. Construir un grupo de afirmaciones para realizar una tarea simple (ej. Instrucciones Tortuga).
5. Identificar los empleos que utilizan la computación y la tecnología.
6. Recopilar y organizar la información utilizando herramientas para mapeo de conceptos.

Grados 3–6 (N1:6.CPP)

El estudiante será capaz de:

1. Utilizar recursos tecnológicos (ej. calculadoras, pruebas para recopilación de datos, dispositivos móviles, videos, software educativo y herramientas de la red) para resolver problemas y para autoaprendizaje.
2. Uso de herramientas productivas y periféricos de uso general para apoyar la productividad general, remediar deficiencias en las habilidades y facilitar el aprendizaje.
3. Uso de herramientas tecnológicas (ej. Autoría de texto y multimedia, presentaciones, herramientas web, cámaras digitales y escáner) para escritura individual y colaborativa, comunicación y actividades de publicación.
4. Reunir y manipular datos utilizando una variedad de herramientas digitales.
5. Construct a program as a set of step-by-step instructions to be acted out (e.g., make a peanut butter and jelly sandwich activity). Construir un programa como un conjunto de instrucciones paso a paso para ser ejecutado (por ejemplo, hacer una actividad de realizar un emparedado de mantequilla de maní y jalea).
6. Implementar soluciones a problemas utilizando un lenguaje de programación visual basado en bloques.

7. Utilizar dispositivos de computación para acceder a información remota, comunicarse con otros en apoyo al aprendizaje directo e independiente y buscar intereses personales.

8. Navegar entre páginas web utilizando los hipervínculos y realizar búsquedas simples utilizando motores de búsqueda.

9. Identificar la amplia variedad de empleos de que requieren conocimientos o el uso de computación.

10. Recopilar y manipular datos utilizando una variedad de herramientas digitales.

Computadoras y Dispositivos de Comunicación (CD)

Grados K–3 (N1:3.CD)

El estudiante será capaz de:

1. Use standard input and output devices to successfully operate computers and related technologies. Utilización de dispositivos estándar de entrada y salida para operar con éxito las computadoras y las tecnologías relacionadas.

Grados 3–6 (N1:6.CD)

El estudiante será capaz de:

1. Demostrar un nivel adecuado de competencia con los teclados y otros dispositivos de entrada y salida.

2. Comprender la presencia permanente de las computadoras y la computación en la vida diaria (por ejemplo, correo de voz, descarga de videos y archivos de audio, hornos de microondas, termostatos, Internet inalámbrica, los dispositivos móviles de computación, sistemas de GPS).

3. Aplicar estrategias para identificar problemas simples de hardware y software que pueden ocurrir durante el uso.

4. Identificar que la información llega a la computadora de muchas fuentes en una red.

5. Identificar los factores que distinguen a los seres humanos de las máquinas.

6. Reconocer que las computadoras modelan un comportamiento inteligente (como en la robótica, reconocimiento de voz y lenguaje y la animación por computadora).

Impactos Globales, Éticos y de la Comunidad (CI)

Grados K–3 (N1:3.CI)

El estudiante será capaz de:

1. Practicar la ciudadanía digital responsable (comportamientos legales y éticos) en el uso de sistemas de tecnología y software.

2. Identificar los comportamientos sociales y éticos positivos y negativos para la utilización de la tecnología

Grados 3–6 (N1:6.CI)

El estudiante será capaz de:

1. Discutir los temas básicos relacionados con el uso responsable de la tecnología e información y las consecuencias de su uso inapropiado.

2. Identificar el impacto de la tecnología (por ejemplo, redes sociales, intimidación cibernética, computación y comunicación móvil, tecnologías web, seguridad cibernética y virtualización) en la vida personal y la sociedad.

3. Evaluar la exactitud, relevancia, pertinencia, integralidad y las influencias que se producen en las fuentes de información electrónica.

4. Comprender los problemas éticos relacionadas con las computadoras y las redes (por ejemplo, igualdad de acceso, seguridad, privacidad, derechos de autor y propiedad intelectual).

5.2 Nivel 2: Ciencias de la Computación y la Comunidad (N2)

Las expectativas de aprendizaje cubiertas por estos estándares apoyan a los estudiantes de la escuela media / secundaria en el uso del pensamiento computacional como herramienta para resolver problemas. Ellos comienzan a apreciar la ubicuidad de la computación y la manera en que las ciencias de la computación facilitan la comunicación y la colaboración. Los estudiantes comienzan a experimentar el pensamiento computacional como un medio para abordar los asuntos

relevantes, no sólo para ellos, sino para el mundo que les rodea. Las experiencias de aprendizaje creadas a partir de estos estándares deben ser relevantes para los estudiantes y deben promover su percepción de sí mismos como solucionadores proactivos de problemas dentro de su comunidad. Estos deben ser diseñados enfocándose en el aprendizaje activo y la exploración que puede enseñarse como un curso de ciencias de la computación o como unidades dentro de otras áreas del plan de estudios, como las ciencias sociales, artes del lenguaje, matemáticas y ciencia.

Los estándares curriculares del Nivel 2 suponen que los estudiantes ya han conocido los conceptos de pensamiento computacional de representación de datos, algoritmos y resolución de problemas y que han tenido experiencia en el uso de herramientas y recursos tecnológicos para el aprendizaje, creación de artefactos digitales y colaboración. Los estudiantes también deben haber aprendido sobre las carreras que utilizan la computación y la tecnología, dispositivos y equipos estándar de entrada / salida, la terminología básica de la computadora, los principios de actuación responsable y el uso ético de las computadoras al utilizarlos de forma independiente y trabajando en equipo.

Los estudiantes menores de las escuelas secundarias se están desarrollando socialmente y emocionalmente. Ellos interactúan en esferas sociales más grandes que los estudiantes de primaria y están en transición de un enfoque en uno mismo a uno grupal. Sus esferas sociales están creciendo más allá del núcleo familiar para incluir compañeros, amigos, maestros, entrenadores y otros miembros de la comunidad. Reconociendo estos cambios, el enfoque del Nivel 2 está en el uso de las computadoras y la computación como individuos y miembros de la comunidad. De esta manera, los estudiantes comienzan a experimentar el pensamiento computacional como un medio de comunicación con los demás y como un medio para abordar asuntos relevantes en la comunidad.

Los objetivos del plan de estudios del nivel 2 son involucrar a los estudiantes en el uso de pensamiento computacional como herramienta para resolver problemas, enseñarles a usar los conceptos y métodos de programación mientras crean artefactos digitales y retener su interés en la computación como un campo relevante y emocionante. Las oportunidades de

aprendizaje deben ser presentadas de forma que sean activas, conectadas y relevantes para ellos y deben promover la percepción de sí mismos como solucionadores empoderados y proactivos de problemas, creadores e innovadores capaces de cambiar el mundo. Las experiencias de aprendizaje de colaboración en este nivel deben preparar a los estudiantes para trabajar en equipo y construir asociaciones de apoyo.

A medida que los estudiantes comienzan a dominar los conceptos fundamentales de las ciencias de la computación y las prácticas, es vitalmente importante que aprendan que estos conceptos y prácticas les facultan para crear innovaciones, herramientas y aplicaciones. Los estudiantes también deben saber que con el conocimiento y el acceso viene la responsabilidad, por lo que los temas de uso ético y responsable de la computación y la información son también elementos esenciales de este plan de estudios.

Estos son los estándares que todos los estudiantes de los grados 6-9 deben cumplir en las cinco ramas:

Pensamiento Computacional: (CT)

El estudiante será capaz de:

1. Utilizar los pasos básicos en la para resolver problemas algorítmicos para diseñar soluciones (por ejemplo, el planteamiento del problema y la exploración, el examen del muestreo, diseño, implementación de una solución, prueba y evaluación).
2. Describir el proceso de paralelización en lo relativo a resolver problemas.
3. Definir un algoritmo como una secuencia de instrucciones que pueden ser procesadas por una computadora.
4. Evaluate ways that different algorithms may be used to solve the same problem. Evaluar maneras diferentes algoritmos que pueden ser utilizados para resolver el mismo problema.
5. Representar los algoritmos de búsqueda y organización.
6. Describir y analizar una secuencia de instrucciones que se lleva a cabo (por ejemplo, describir el comportamiento de un personaje en un videojuego impulsado por reglas y algoritmos).

7. Representar los datos de varias maneras, incluyendo textos, sonidos, imágenes y números.
8. Utilizar representaciones visuales de estados problemas, estructuras y datos (por ejemplo, gráficos, tablas, diagramas de red, diagramas de flujo).
9. Interactuar con los modelos específicos de contenido y simulaciones (por ejemplo, ecosistemas, epidemias, la dinámica molecular) para apoyar el aprendizaje y la investigación.
10. Evaluar qué tipo de problemas se pueden resolver utilizando modelado y simulación.
11. Analizar el grado en que un modelo por computadora representa con precisión el mundo real.
12. Utilizar la abstracción para descomponer un problema en sub problemas.
13. Comprender la noción de jerarquía y la abstracción en la computación incluyendo lenguajes de alto nivel, traducción, el conjunto de instrucciones y circuitos lógicos.
14. Examinar las conexiones entre los elementos de las matemáticas y las ciencias de la computación, incluyendo los números binarios, lógica, conjuntos y funciones.
15. Proveer ejemplos de aplicaciones interdisciplinarias del pensamiento computacional.

Colaboración (CL)

El estudiante será capaz de:

1. Aplicar las herramientas de productividad / multimedia y periféricos para la colaboración en grupo y apoyar el aprendizaje a través del plan de estudios.
2. Diseñar colaborativamente, desarrollar, publicar y presentar productos (por ejemplo, vídeos, podcasts, sitios web), utilizando recursos tecnológicos que demuestran y comunican conceptos curriculares.
3. Colaborar con compañeros, expertos y otras personas utilizando prácticas de colaboración como programación en parejas, trabajo en equipo para proyectos y la participación en grupos de actividades de aprendizaje activo.

4. Exponer las disposiciones necesarias para la colaboración: proporcionar retroalimentación útil, integrar la retroalimentación, la comprensión y aceptación de perspectivas múltiples y la socialización.

Practica de Computación y Programación (CPP)

El estudiante será capaz de:

1. Seleccionar herramientas y recursos tecnológicos apropiados para llevar a cabo una variedad de tareas y resolver problemas.
2. Utilizar una variedad de herramientas multimedia y periféricos para apoyar la productividad personal y el aprendizaje durante el plan de estudio.
3. Diseñar, desarrollar, publicar y presentar productos (por ejemplo, páginas web, aplicaciones móviles, animaciones), utilizando recursos tecnológicos que demuestran y comunican conceptos curriculares.
4. Comprender los algoritmos y su aplicación práctica.
5. Implementar soluciones a los problemas utilizando un lenguaje de programación, incluyendo: comportamiento en circuitos, problemas enunciados condicionales, lógica, expresiones, variables y funciones.
6. Demostrar buenas prácticas en la seguridad de la información personal, utilizar contraseñas, codificación y transacciones seguras.
7. Identificar las carreras interdisciplinarias que han mejorado por las ciencias de la computación.
8. Demostrar disposiciones susceptibles para resolver problemas de composición abierta y programación (por ejemplo, la comodidad con la complejidad, la persistencia, lluvia de ideas, adaptabilidad, paciencia, propensión a jugar, creatividad y la aceptación del reto).
9. Recopilar y analizar los datos que se envían desde múltiples programas de computación.

Computadoras y Dispositivos de Comunicación (CD)

El estudiante será capaz de:

1. Reconocer que las computadoras son dispositivos que ejecutan programas.

2. Identificar una variedad de dispositivos electrónicos que contienen procesadores computarizados.
3. Demostrar y comprender la relación entre el hardware y el software.
4. Utilizar terminología exacta y apropiadamente desarrollada al comunicarse acerca de la tecnología.
5. Aplicar estrategias para identificar y resolver problemas rutinarios de hardware que pueden ocurrir durante el uso diario de la computadora.
6. Describir los principales componentes y funciones de los sistemas y redes de computación.
7. Describir lo que distingue a los humanos de las máquinas centrándose en la inteligencia humana frente a la inteligencia artificial y las formas en que nos podemos comunicar.
8. Describir las formas en que las computadoras utilizan modelos de comportamiento inteligente (por ejemplo, el movimiento del robot, el habla y la comprensión del lenguaje y visión por computadora).

Impactos Globales, Éticos y de la Comunidad (CI)

El estudiante será capaz de:

1. Exhibir comportamientos legales y éticos al utilizar la información y la tecnología y discutir las consecuencias del mal uso.
2. Demostrar conocimiento de los cambios en las tecnologías de información a través del tiempo y los efectos que esos cambios tienen en la educación, el trabajo y la sociedad.
3. Analizar los impactos positivos y negativos de la computación en la cultura humana.
4. Evaluar la precisión, relevancia, pertinencia, integralidad, y parcialidad de las fuentes de información electrónica sobre los problemas del mundo real.
5. Describir los problemas éticos relacionados con las computadoras y las redes (por ejemplo, seguridad, privacidad, propiedad e intercambio de información).
6. Discutir cómo la distribución desigual de los recursos de computación en una economía global plantea cuestiones de equidad, acceso y poder.

5.3 Nivel 3: Aplicar los Conceptos y crear soluciones del mundo real (N3)

El Nivel 3 se divide en tres campos diferenciados, cada uno de los cuales se centra en un aspecto diferente de las ciencias de la computación como disciplina. A lo largo de estos cursos, los estudiantes aprenderán los conceptos avanzados de las ciencias de la computación y a aplicar estos conceptos para desarrollar artefactos del mundo virtual y real. Las experiencias de aprendizaje creadas a partir de estos estándares deben enfocarse en la exploración de los problemas del mundo real, la aplicación del pensamiento computacional para el desarrollo de soluciones de problemas y las interconexiones entre las ciencias de la computación y otras materias académicas. Estas experiencias también deberían incluir un enfoque en el aprendizaje colaborativo y la comunicación efectiva.

El Nivel 3 incluye los siguientes cursos:

3A: Las Ciencias de la Computación en el mundo moderno (MW)

3B: Principios de las Ciencias de la Computación (CP)

3C: Temas en Ciencias de la Computación (TO)

Normalmente, el curso 3A será un requisito previo para cualquiera de los otros dos. Las siguientes secciones describen estos tres cursos más detalladamente. Estos cursos han sido diseñados asumiendo que son de un año de duración. Las escuelas no los pueden ofrecer como cursos de un año, puede ser necesario ajustar la cantidad de estándares que pueden ser cubiertos.

5.3. A *Ciencias de la Computación en el Mundo Moderno (MW)*

Ciencias de la Computación en el mundo moderno es un curso diseñado para exponer a los estudiantes a la naturaleza interdisciplinaria de las ciencias de la computación en la sociedad dinámica y globalmente conectada de hoy. Los estudiantes tendrán la oportunidad de explorar los usos de las ciencias de la computación como herramienta en la creación de soluciones eficaces a los problemas complejos contemporáneos. El carácter práctico de la asignatura pretende proporcionar a los estudiantes la oportunidad de explorar la comprensión conceptual en un ambiente de aprendizaje práctico. Este curso es recomendado para todos los estudiantes, ya que proporciona una visión

general de las ciencias de la computación y sus aplicaciones en diversas disciplinas, profesiones y actividades personales.

En este curso, los estudiantes aprenderán a utilizar el pensamiento computacional para desarrollar soluciones algorítmicas a problemas del mundo real. Ellos comenzarán a comprender los diferentes niveles de complejidad al resolver problemas y determinar cuándo los proyectos en equipo pueden generar soluciones a los problemas más efectivos que los esfuerzos individuales. Los estudiantes aprenderán y utilizar un lenguaje(s) de programación y herramientas relacionadas, así como las herramientas de colaboración adecuadas, dispositivos de computación y los entornos de red. Por último, demostrarán que comprenden las implicaciones sociales y éticas de su trabajo y mostrarán un comportamiento de comunicación apropiado al trabajar en equipo.

Las Ciencias de la Computación en el mundo moderno es un curso diseñado para todos los estudiantes de noveno y décimo grado. Este curso se basa en las habilidades esenciales que todos los estudiantes de las escuelas secundarias deben tener después de graduarse. También proporciona las habilidades necesarias para estudios más avanzados en los niveles 3.B y 3.C Se recomienda que este curso sea requisito para todos los estudiantes.

Pensamiento Computacional (CT)

El estudiante será capaz de:

1. Uso de las funciones y parámetros, clases y métodos predeterminados para dividir un problema complejo en partes más simples.
2. Describir un proceso de desarrollo de software que se utiliza para resolver los problemas de software (por ejemplo, diseño, codificación, prueba, verificación)
3. Explicar cómo la secuencia, selección, iteración y recursión son bloques de construcción de algoritmos.
4. Comparar las técnicas para analizar las recopilaciones masivas de datos.
5. Describir la relación entre las representaciones binarias y hexadecimales.
6. Analizar la representación y las compensaciones entre las diversas formas de información digital.

7. Describir cómo se almacenan los diferentes tipos de datos en un sistema computarizado.

8. Utilizar los modelos y la simulación para representar y entender los fenómenos naturales.

9. Discutir el valor de la abstracción para manejar la complejidad de un problema.

10. Describir el concepto de procesamiento paralelo como estrategia para resolver los grandes problemas.

11. Describir cómo la computación comparte características con el arte y la música mediante la traducción de la intención humana en un artefacto.

Colaboración (CL)

El estudiante será capaz de:

1. Trabajar en equipo para diseñar y desarrollar un artefacto de software. Trabajar en equipo para diseñar y desarrollar un artefacto de software.
2. Utilizar las herramientas de colaboración para comunicarse con los miembros del equipo del proyecto (por ejemplo, foros de discusión, wikis, blogs, control de versiones, etc.)
3. Describir cómo la computación aumenta las formas tradicionales de expresión y permite nuevas como la expresión, la comunicación y la colaboración.
4. Identificar cómo la colaboración influye en el diseño y desarrollo de productos de software.

Practica de Computación y Programación (CPP)

El estudiante será capaz de:

1. Crear y organizar las páginas web mediante el uso de una variedad de herramientas para diseño de programación web.
2. Utilizar los dispositivos móviles / emuladores para diseñar, desarrollar e implementar aplicaciones para computación móvil.
3. Utilizar diversos métodos de depuración y prueba para garantizar la exactitud del programa (por ejemplo, casos de prueba, prueba de la unidad, caja blanca, caja negra, las pruebas de integración)

4. Aplicar análisis, diseño e implementación de técnicas para resolver problemas (por ejemplo, utilizar uno o varios modelos del ciclo de vida del software).
5. Utilizar la aplicación de programa interfaces (API) y las bibliotecas para facilitar la programación de soluciones.
6. Seleccionar los formatos adecuados de archivo para diversos tipos y usos de los datos.
7. Describir la variedad de lenguajes de programación disponibles para resolver problemas y desarrollar sistemas.
8. Explicar el proceso de ejecución de programas.
9. Explicar los principios de seguridad examinando la codificación, la criptografía, y las técnicas de autenticación.
10. Explorar una variedad de carreras para las que la computación es central.
11. Describir las técnicas para localizar y reunir conjuntos pequeños y grandes de datos.
12. Describir cómo se utilizan las funciones matemáticas y estadísticas, los conjuntos y la lógica en la computación.

Computadoras y Dispositivos de Comunicación (CD)

El estudiante será capaz de:

1. Describir las características únicas de las computadoras integradas en los dispositivos móviles y los vehículos (por ejemplo, teléfonos celulares, automóviles, aviones).
2. Desarrollar criterios para la compra o actualización de hardware del sistema de computación.
3. Describir los principales componentes de la organización de los computadores (por ejemplo, entrada, salida, procesamiento y almacenamiento).
4. Comparar las diversas formas de entrada y salida.
5. Explicar los múltiples niveles de hardware y software que apoyan la ejecución de un programa (por ejemplo, los compiladores, intérpretes, sistemas operativos, redes).

6. Aplicar estrategias para identificar y resolver problemas rutinarios de hardware y software que se producen en la vida diaria.
7. Comparar y contrastar las estrategias de redes cliente-servidor y persona-persona.
8. Explicar los componentes básicos de las redes de las computadoras (por ejemplo, servidores, protección de archivos, enrutamiento, colas de impresión y colas, recursos compartidos y tolerancia a fallos).
9. Describir cómo el Internet facilita la comunicación global.
10. Describir las principales aplicaciones de la inteligencia artificial y la robótica.

Impactos Globales, Éticos y de la Comunidad (CI)

El estudiante será capaz de:

1. Comparar los comportamientos apropiados e inapropiados de las redes sociales
2. Discutir el impacto de la tecnología de la computación en las empresas y el comercio (por ejemplo, el seguimiento automatizado de las mercancías, las transacciones financieras automatizadas, el comercio electrónico, el cloud computing).
3. Describir el papel que la tecnología adaptativa puede tener en las vidas de las personas con necesidades especiales.
4. Comparar los impactos positivos y negativos de la tecnología en la cultura (por ejemplo, las redes sociales, entrega de noticias y otros medios de comunicación públicos y la comunicación intercultural).
5. Describir las estrategias para determinar la fiabilidad de la información que se encuentra en Internet.
6. Diferenciar entre el acceso a la información y los derechos de distribución de la información.
7. Describir cómo los diferentes tipos de licencias de software se pueden utilizar para compartir y proteger la propiedad intelectual.
8. Discutir las implicaciones sociales y económicas asociadas al hacking y la piratería de software.

9. Describir las diferentes formas en las que se crea y se comparte el software y sus ventajas y desventajas (software comercial, software de dominio público y de desarrollo de código abierto).

10. Describir los problemas de seguridad y privacidad relacionados con las redes de las computadoras.

11. Explicar el impacto de la brecha digital en el acceso a la información crítica.

5.3. B *Conceptos y Practicas de las Ciencias de la Computación (CP)*

Conceptos y Prácticas de las Ciencias de la Computación es un curso de seguimiento a las Ciencias de la Computación en el Mundo Moderno. Esta diseñado para aprovechar los intereses de aquellos estudiantes que desean mejorar aún más sus estudios en los campos de la computación. En este curso, los estudiantes comenzaran a desarrollar las habilidades de computación del nivel superior y aplicarlos a una variedad de temas y disciplinas. Los estudiantes aprenderán cómo las ciencias de la computación impactan en la sociedad los y promueven el cambio. A través del análisis de los problemas mundiales, los estudiantes explorarán cómo las ciencias de la computación pueden ayudar a resolver problemas del mundo real mediante la innovación, la colaboración y la creatividad. Este curso también proporcionará a los estudiantes la oportunidad de explorar las ciencias de la computación como una carrera potencial a nivel universitario.

En su contenido y pedagogía este curso tiene como objetivo atraer una amplia audiencia. En este curso los estudiantes comenzarán a comprender las ideas centrales de las ciencias de la computación, enfocándose en los conceptos y prácticas del pensamiento computacional y crítico y la participación en actividades que muestran cómo la informática está ayudando a cambiar el mundo. Este curso riguroso también involucra a los estudiantes en los aspectos creativos de las ciencias de la computación.

Conceptos y Prácticas de Ciencias de la Computación está diseñado como un curso electivo dirigido a estudiantes de los grados de 10 a 12. Los estudiantes inscritos en este curso deben haber completado previamente *Ciencias de la computación en el Mundo Moderno*. Debido a su contenido matemático, los estudiantes también deben haber completado por lo

menos Álgebra I. No obstante, este curso debe ser accesible a todos los estudiantes. Los estudiantes con un interés especial en un área específica de las ciencias de la computación (por ejemplo, redes, programación, diseño de juegos, etc.) pueden continuar su exploración a través de los cursos descritos en el Nivel 3.C: *Temas en Ciencias de la Computación*.

Pensamiento Computacional (CT)

El estudiante será capaz de:

1. Classify problems as tractable, intractable, or computationally unsolvable. Clasificar los problemas en tratable, intratable o imposible de resolver computacionalmente.
2. Explicar el valor de algoritmos heurísticos para soluciones aproximadas a problemas intratables.
3. Examinar críticamente los algoritmos clásicos e implementar un algoritmo original.
4. Evaluar los algoritmos por su eficiencia, exactitud y claridad.
5. Utilizar el análisis de datos para mejorar la comprensión de los sistemas complejos naturales y humanos.
6. Comparar y contrastar las estructuras de datos simples y sus usos (por ejemplo, matrices y listas).
7. Discutir la interpretación de secuencias binarias en una variedad de formas (por ejemplo, instrucciones, números, texto, sonido, imagen).
8. Utilizar modelos y simulaciones para ayudar a formular, refinar y probar las hipótesis científicas.
9. Analizar los datos e identificar patrones a través de los modelos y la simulación.
10. Descomponer un problema definiendo nuevas funciones y clases.
11. Demostrar la concurrencia mediante la separación de los procesos en hilos y dividiendo los datos en ramas paralelas.

Colaboración (CL)

El estudiante será capaz de:

1. Utilizar las herramientas de colaboración, sistemas de control de versiones y entornos de desarrollo integral (IDE) mientras se trabaja en un proyecto de colaboración de software.
2. Demostrar el proceso del ciclo de vida del software participando en un equipo de proyecto de software.
3. Evaluar los programas escritos por otros para mejorar la legibilidad y facilidad de uso.

Practica de Computación y Programación (CPP)

El estudiante será capaz de:

1. Herramientas avanzadas para crear artefactos digitales (por ejemplo, diseño web, animación, video, multimedia).
2. Utilizar las herramientas de abstracción para descomponer un problema de computación a gran escala (por ejemplo, abstracción de procedimientos, diseño orientado a objetos, diseño funcional).
3. Clasificar los lenguajes de programación basados en su nivel y dominio de aplicación.
4. Explorar los principios de diseño de sistema en escala, eficiencia y seguridad.
5. Implementar principios de seguridad mediante la implementación de estrategias de cifrado y autenticación.
6. Anticipar las futuras carreras y las tecnologías que existirán.
7. Utilizar el análisis de datos para mejorar la comprensión de los sistemas complejos naturales y humanos.
8. Implementar diversas técnicas de recolección de datos para diferentes tipos de problemas.

Computadoras y Dispositivos de Comunicación (CD)

El estudiante será capaz de:

1. Discutir el impacto de las modificaciones en la funcionalidad de los programas de aplicación.
2. Identificar y describir el hardware (por ejemplo, capas físicas, portales lógicas, chips, componentes).

3. Identificar y seleccionar el formato más adecuado en función de las compensaciones (por ejemplo, precisión, velocidad, facilidad de manipulación).
4. Describir los problemas que impactan la funcionalidad de la red (por ejemplo, latencia, ancho de banda, servidores de seguridad, capacidad de servidor).
5. Explicar la noción de un comportamiento inteligente a través de modelos por computadora y la robótica.

Impactos Globales, Éticos y de la Comunidad (CI)

El estudiante será capaz de:

1. Demostrar el uso ético de los medios de comunicación y dispositivos de comunicación modernos.
2. Analizar los efectos beneficiosos y perjudiciales de las innovaciones de computación.
3. Resumir cómo los mercados financieros, las transacciones y las predicciones han sido transformados por la automatización.
4. Resumir cómo la computación ha revolucionado la forma de construir organizaciones e infraestructuras reales y virtuales.
5. Identificar las leyes y regulaciones que afectan el desarrollo y uso de software.
6. Analizar el impacto de la regulación gubernamental sobre la privacidad y la seguridad.
7. Diferenciar entre código abierto, gratuito y licencias de propietarias de software y su aplicación a los diferentes tipos de software.
8. Relacionar los asuntos de equidad, acceso y poder en la distribución de los recursos computarizados en una sociedad global.

5.3. C *Temas en Ciencias de la Computación (TO)*

En este nivel, los estudiantes interesados y calificados deben ser capaces de seleccionar uno de varios cursos electivos para lograr comprender a profundidad o lograr habilidades especiales en áreas específicas de ciencias de la computación. Todas estas asignaturas optativas pueden requerir el curso Nivel 3A, como requisito previo, mientras que algunos pueden requerir el curso Nivel 3B también. Más importante aún, estos cursos

proporcionan a los estudiantes la oportunidad de explorar temas de interés personal con mayor profundidad y así prepararse para el lugar de trabajo o para continuar sus estudios en el nivel post-secundario.

Estas optativas incluyen, pero no se limitan necesariamente a:

- Colocación Avanzada (CP) Ciencias de la Computación A,
- Un curso basado en proyectos en los que los estudiantes cubren un tema a profundidad,
- Un curso suministrado por el proveedor, lo que puede estar relacionado con la certificación profesional.

Estas alternativas se discuten detalladamente a continuación.

5.3. C.1 *CA Ciencias de la Computación A*

El plan de estudios de los cursos de nivel avanzado de ciencias de la computación está bien establecido (CA, 2010) y se ofrece en muchas escuelas secundarias para estudiantes que planean continuar su educación en una universidad de dos o cuatro años, posiblemente en las ciencias de la computación, negocios o un campo relacionado. El CA de Ciencias de la Computación es un curso que enfatiza la resolución de problemas y el desarrollo de algoritmos y estructuras de datos elementales introduce. Los estudiantes que completen este curso y obtengan una buena puntuación en el examen pueden calificar para un semestre de crédito universitario. Los estudiantes que toman el curso CA de Ciencias de la Computación A deben haber completado los niveles 1, 2 y 3A. Es decir, tienen que estar familiarizados con los conceptos algorítmicos / computacionales introducidos en esos niveles. El curso de Nivel 3B proporciona una excelente base en los principios de computación y también puede ser útil para los estudiantes que tengan la intención de tomar El curso CA de Ciencias de la Computación A.

5.3. C.2 *Cursos basados en proyectos*

Un curso basado en proyectos estará disponible para todos los estudiantes que han completado los cursos de los Niveles 1 y 2. La mayoría de los cursos basados en proyectos también requieren la finalización del curso Nivel 3A. Algunas variantes de este curso también

requieren completar el curso Nivel 3B. Un curso basado en proyectos puede ser de un semestre o de un año.

Los proyectos de este tipo reflejarán naturalmente diversos intereses de los estudiantes y los conocimientos específicos de la facultad. Los proyectos específicos que se eligen año tras año también van a evolucionar para reflejar las características cambiantes de las ciencias de la computación y la tecnología de la información. Idealmente, cada proyecto debe basarse en los conceptos básicos de las ciencias de la computación y a ayudar a los estudiantes a desarrollar habilidades profesionales en la aplicación de la tecnología. Las escuelas también deben considerar ofrecer cursos basados en proyectos en conjunto con las universidades locales para asegurar estar actualizados y aprovechar la experiencia externa.

Mientras que algunos de los cursos basados en proyectos pueden basarse más en habilidades, todavía deben estar vinculados a las actividades del software "detrás de escena" y otros principios generales de las ciencias de la computación. Hacer este tipo de conexiones permite a los estudiantes resolver problemas cuando el software no funciona según lo previsto.

Aquí hay algunos cursos basados en proyectos que podrían cumplir los requisitos de un curso de Nivel 3C.

Ejemplo: Autoedición: Este curso introduce la planificación, diseño de página y el uso de plantillas para crear folletos, documentos y boletines informativos. El procesamiento de palabras y la fluidez de la edición gráfica (Nivel 2) ayudarán a asegurar el éxito del estudiante. Los métodos de distribución de estos documentos, tanto escritos como en formato electrónico deben ser incluidos. Para ello será necesaria la comprensión de los conceptos de Internet y la conectividad de red (nivel 3A).

Ejemplo: Comunicaciones Técnicas: La capacidad de comunicarse y compartir ideas debe ser un requisito básico para todos los graduados de la escuela secundaria. Este tipo de proyecto se centra en la documentación del usuario final y la investigación y presentación de la información técnica a personas sin conocimientos técnicos, en forma oral, escrita y multimedia. Se requiere dominio de procesadores de texto y programas de presentación y la comprensión de las ciencias de la computación y la tecnología (nivel 3A).

Ejemplo: Multimedia: El uso de multimedia ha aumentado de manera constante a nivel del usuario, alimentada por un hardware más eficiente y la disponibilidad de las cámaras digitales y equipos de audio digital. Sin embargo, a menudo se abusa de la multimedia al incorporarla a los programas, páginas web y presentaciones. Este proyecto proporcionará instrucción en el uso de audio digital y vídeo y el software de edición correspondiente. Un aspecto importante será el despliegue de multimedia de manera responsable. Se requieren habilidades básicas de software (Nivel 2) y una comprensión de los conceptos de multimedia (Nivel 3A).

Ejemplo: Gráficos: Esta clase explora mapas de bits y gráficos vectoriales. La discusión incluye los beneficios y limitantes de cada tipo de software y la experiencia práctica con ambos. CAD, CAM y software de diseño 3-D deben ser exploradas, así como el software de mapas de bits para la creación y edición de gráficos. Se requiere disponibilidad de una cámara digital y escáner. El Despliegue Responsable de los gráficos incluyendo los temas de estilo y legales que hay que investigar. La discusión de los gráficos basados en vectores se facilitara por la finalización del Nivel 3A Limites de computadoras y diseño para usabilidad.

Ejemplo: Programación de juego: Este curso ayuda a los estudiantes a entender la creatividad necesaria para programar eficazmente y refuerza el ciclo de desarrollo de software. Los estudiantes planean, diseñan, codifican y prueban los juegos de computadora. Se requieren conocimientos de programación básicos y una comprensión de los medios de comunicación (nivel 3B).

Ejemplo: Modelado por Computadora: Este curso explora el modelado por computadora de sistemas complejos. Utilizando técnicas basadas en agentes, asuntos importantes a nivel local como la propagación de una enfermedad, los ecosistemas y los patrones de tráfico pueden ser modelados e investigados y se pueden diseñar y probar de forma virtual los esfuerzos por mejorar los impactos negativos. En este curso los estudiantes llegarán a entender cómo las interacciones entre los elementos individuales (por ejemplo, personas, animales o vehículos) y las personas y su entorno pueden dar lugar a patrones emergentes y a menudo imprevisibles. Los proyectos de los estudiantes incluyen la abstracción de un problema del mundo real o escenario, implementando un modelo por computadora

especificando los agentes, interacciones y el ambiente en el modelo y utilizando la automatización para realizar ejecuciones múltiples de la simulación como un banco de pruebas experimental. El análisis propio del modelo y los datos que este produce determinan si el mundo simulado se refiere al mundo real.

Ejemplo: Desarrollo Web: En varios lugares del plan de estudios los alumnos están expuestos a los conceptos de Internet y HTML. Este curso incluye hojas de estilo en cascada (CSS) y presenta una visión más profunda de los problemas de diseño y desarrollo que deben tomarse en cuenta para una implementación internacional multiplataforma. La estandarización del desarrollo de páginas web utilizando las recomendaciones del Consorcio WWW es una cuestión de enfoque. El Desarrollo de páginas web incluirá HTML y CSS de codificación usando un editor de texto y la utilización de secuencias de comandos simples para mejorar las páginas web.

Ejemplo: Programación Web: Los estudiantes que han completado con éxito los niveles 3A y 3B, pero no desean tomar un curso CA, pueden disfrutar aplicando sus conocimientos de programación para la WWW. Para tener éxito, será necesario un conocimiento sólido de los conceptos de internet, temas de desarrollo web y los conceptos básicos de programación. Los temas de este curso pueden incluir lado del cliente y lenguajes de scripting del lado del servidor. Los estudiantes tendrán que escribir guiones y desplegarlos dentro de las páginas web o en el servidor web.

Ejemplo: Tecnologías emergentes: Este proyecto puede incluir temas distintos y se espera que su contenido cambie de forma regular. Se Necesitara desarrollar un Currículo y materiales para este tema a partir de los recursos actuales en la web, tal vez en conjunto con los colegios y universidades locales y con el aporte del sector profesional de la comunidad empresarial.

Ejemplo: Desarrollo de Software Gratis y de Recursos abiertos (FOSS): Los estudiantes que han completado con éxito Nivel 3A pueden inscribirse en un curso en el que puedan contribuir a un proyecto en curso de software FOSS. Aquí, pueden leer códigos escritos por otros, dar sugerencias para nuevas características, identificar errores, escribir documentación del usuario y aprender a utilizar las tecnologías modernas de

colaboración. Los estudiantes participarán activamente en foros de discusión del proyecto. Ejemplos de proyectos de software libre que son accesibles a los estudiantes se identifican en <http://hfoss.org>.

Algunos otros temas (junto con sus requisitos previos) incluyen:

- La computadora y la animación (Nivel 3A)
- Tecnologías de Red (Nivel 3A)
- Simulaciones de Programación (ej. Un experimento de química controlado por computadora) (Niveles 3A y 3B)
- Diseño orientado a objetos y codificación (Nivel 3B)
- Uso efectivo de las aplicaciones por computadora (Niveles 1 y 2)

5.3. C.3 *Cursos para la Certificación de la Industria*

Dicho curso está dirigido principalmente a los estudiantes que piensan ingresar a la fuerza laboral, **Ejemplo: A+ Técnico Certificado:** La certificación + CompTIA A es el estándar de la industria para los técnicos de soporte de computadoras. La certificación de un proveedor neutral internacional demuestra competencia en materias tales como la instalación, mantenimiento preventivo, redes, seguridad y solución de problemas (<http://www.comptia.org/certificaciones/inscrito/A.aspx>). Dos exámenes diferentes: CompTIA A + Esenciales y CompTIA A + Aplicación Práctica. El uso de las habilidades de pensamiento crítico para resolver problemas es necesario para solucionar y resolver problemas. Estas habilidades refuerzan y amplían los conceptos presentados en los niveles 1, 2 y 3A.

Ejemplo: Seguridad Rápida+: El campo de la seguridad de las computadoras es una de las disciplinas de mayor crecimiento en las Tecnologías de la Información. La Seguridad CompTIA + es una certificación internacional de proveedor neutral que demuestra competencias en seguridad de red, amenazas y vulnerabilidades, control de acceso y gestión de identidades, criptografía, y más ([http://certification.comptia.org/Obtener la certificación / certificaciones / seguridad. Aspx](http://certification.comptia.org/Obtener%20la%20certificaci%C3%B3n/certificaciones/seguridad.aspx)). Estas habilidades refuerzan y amplían los conceptos presentados en los niveles 1, 2 y 3A.

continuar su educación en una escuela técnica post-secundaria o entrar al programa de dos años de Asociado universitario del programa de Ciencias Aplicadas. Los estudiantes que toman este curso deben haber completado los Niveles 1 y 2 y por lo general los cursos del nivel 3A.

La Certificación de la Industria ofrece un estándar que es útil para los posibles empleadores en la evaluación de un candidato que no tiene ninguna experiencia laboral previa. Las Certificaciones de la industria pueden ser de vendedor neutral o de vendedor patrocinado. Los planes de estudio de proveedores patrocinado deben ser evaluados cuidadosamente. Aunque es rica en contenido, algunos de estos cursos están estructurados para enfatizar productos patentados, en lugar de conceptos generales. Los estudiantes que completan los cursos de certificación deben ser alentados a tomar el examen correspondiente como prueba de los conocimientos adquiridos. Éstos son algunos ejemplos de los programas de certificación de proveedor neutral.

Ejemplo: Certificado de Webmaster de Internet (CIW): El Plan de estudios básico de CIW se centra en las normas fundamentales de la web, incluyendo diseño, desarrollo y seguridad web. Las Certificaciones CIW verifican que las personas certificadas cuentan con las habilidades para tener éxito en un mundo impulsado por la tecnología (http://ciwcertified.com/About_CIW/index.php). EL currículo CIW es avalado en diversas zonas del país (http://ciwcertified.com/About_CIW/Why_CIW/highschools.php). El examen de los Fundamentos Web CIW requiere la competencia en el negocio de Internet, diseño de páginas web y los fundamentos de redes. Muchos de estos conceptos se introducen en los niveles 1, 2 y 3A.

Encuentre información más detallada sobre estos y otros programas de certificación, tanto específicos del proveedor y proveedor neutral, buscando en la Web.

6. Desafíos de Implementación

Entendemos que muchos obstáculos se encuentran en el camino para llegar a un modelo ideal de educación de Ciencias de la Computación de K-12 para todos los estudiantes. ¿Cómo encontrar espacio en un plan de estudios tan lleno? ¿Cómo se contrataran profesores calificados, capacitados y acreditados? ¿En el mundo de los estándares de evaluación centrada de las escuelas,

deberían las ciencias de la computación apoyar los estándares existentes o si se diseñan otros nuevos para las Ciencias de la Computación? Estas y otras preguntas y desafíos son importantes, pero también son los beneficios de las ciencias de la computación para los estudiantes y la sociedad se convierte en una parte importante de una educación de alta calidad como las demás disciplinas fundamentales.

La enseñanza efectiva de cualquier tema depende de la existencia de estándares firmes de aprendizaje para los estudiantes, los estándares explícitos de certificación de maestros, programas apropiados de formación de docentes y materiales curriculares eficaces y un núcleo de profesores dispuestos, capaces y capacitados para impartir el currículo. La educación de Ciencias de la computación de K-12 se enfrenta a desafíos únicos a lo largo de todas estas líneas.

El reto de mejorar la enseñanza de computación es importante y requerirá la atención e intervención de diversas instituciones. Las organizaciones profesionales en ciencias de la computación pueden hacer una contribución importante. CSTA, por ejemplo, es una organización profesional que apoya y promueve la enseñanza de las ciencias de la computación y otras disciplinas de computación. CSTA proporciona un gran número de programas que incluyen el desarrollo y difusión de los recursos de aprendizaje, desarrollo profesional, y la promoción de políticas a nivel estatal y federal para mejorar la enseñanza de computación. Otras organizaciones, como ACM, IEEE Sociedad de Computación, las instituciones de educación superior y organizaciones nacionales y locales de maestros también pueden trabajar para resolver estos problemas en la educación de ciencias de la computación de K-12. La industria también está profundamente afectada por la escasez de información y de trabajadores con habilidades para apoyar y construir las herramientas de la tecnología del futuro. Por tanto, es su mejor interés contribuir significativamente a mejorar el acceso a la calidad de la educación en ciencias de la computación a nivel de K-12.

Para las escuelas que implementen ampliamente estas normas, se necesita trabajar en tres áreas importantes: formación docente, estándares de contenido a nivel estatal y desarrollo de materiales curriculares. Además, las personas en posiciones de liderazgo deben reconocer la importancia de la educación en ciencias de la

computación para el futuro de nuestra sociedad. Los Estados y las organizaciones de acreditación deben hacer de este un factor en su proceso global de acreditación de las escuelas. Algunos estados han comenzado a establecer estándares de contenido de ciencias de la computación para definir modelos para la certificación de maestros, proporcionar capacitación en servicio en ciencias de la computación y experimentar con el desarrollo de nuevos materiales curriculares. Sin embargo, ahora se requiere un esfuerzo y un compromiso mayor.

Recientemente, se han incrementado los esfuerzos para desarrollar estándares nacionales y estatales de contenido para las ciencias de la computación. Los estándares curriculares sirven para definir las competencias y conocimientos de la disciplina para ser adquiridos por cada estudiante. Los estándares de contenido para la enseñanza de computación dentro de los estados se deben elaborar y aprobar de manera similar a lo que ocurre en disciplinas como la ciencia, las matemáticas y redacción. El contenido curricular alineado con estos estándares puede ser desarrollado para el aula.

En el diseño de los estándares estatales, es importante garantizar la distinción entre la enseñanza de habilidades de IT y la enseñanza de las ciencias de la computación en sí. Es decir, las ciencias de la computación debe ser visto como un tema distinto y la tecnología debe ser visto como una herramienta que atraviesa todas las materias. Los estándares tecnológicos existentes, cuando estén presentes, no deben ser sustituidos por los estándares de las ciencias de la computación. (Para un análisis exhaustivo de las formas en que los Estados no han logrado incorporar los estándares de las ciencias de la computación en los estándares estatales de esta manera, ver Funcionando en vacío: el fracaso para enseñar las Ciencias de la Computación de K-12 en la era digital en <http://csta.acm.org/Comunicaciones/sub/documents.html>).

7. Llamado a la Acción

Las ciencias de la computación es una disciplina convencional que ya no puede ser ignorada por las escuelas públicas en el siglo 21. Los estándares de aprendizaje que se detallan en este documento proporcionan una base para que los Estados, las escuelas de educación y los distritos escolares individuales puedan empezar a poner en práctica un plan de estudios

de ciencias de la computación coherente que esté disponible para todos los estudiantes

Hay mucho trabajo por hacer para traducir estos estándares en material didáctico y de laboratorio que sea adecuados que permitan un esfuerzo de desarrollo curricular masivo para tener éxito.

pedagógicamente sólido y culturalmente significativo para todos los estudiantes. Esperamos que los departamentos estatales y federales de educación, corporaciones, fundaciones y otros grupos de interés externos apoyen este trabajo a través de incentivos

Notas

8. Actividades

En estas actividades, se expone la viabilidad de este modelo curricular, proporcionando actividades de ejemplo para los cursos y módulos que se imparten actualmente en varias escuelas de todo el mundo. La estructura de estas actividades se explica a continuación:

ACTIVIDAD: NOMBRE DE LA ACTIVIDAD

- Tiempo:** Número de horas clase necesarias para completar la actividad
- Descripción:** Breve descripción del tema y los objetivos de la actividad.
- Nivel:** 1, 2, o 3, como se definen en este informe (ver Sección 4.1)
- Temas:** Los temas en este nivel cubiertos por esta actividad (ver lista de temas en las secciones 5.1, 5.2 y 5.3 de este informe)
- Conocimiento Previo:** Lo que los estudiantes deben saber antes de comenzar esta actividad
- Notas de Planificación:** Sugerencias para los maestros para preparar esta actividad
- Estrategias de Enseñanza/ Aprendizaje:** Organización de la presentación en clase y de las tareas de los estudiantes
- Evaluación y Valoración:** Las evaluaciones formativas y sumarias del trabajo en clase y laboratorio
- Necesidades:** materiales de apoyo adicionales (por ejemplo, laboratorios escalonados, programas ejemplo, problemas difíciles)
- Recursos:** Enlaces a la fuente de esta actividad, así como otras actividades relacionadas
-

A.1 Ejemplo de Actividades para el Nivel 1

ACTIVIDAD: OBRA DE MISTERIO

- Duración:** 8 horas
- Descripción:** Los estudiantes aprenden y actúan una obra corta de misterio. Otros estudiantes utilizan la lógica para resolver el misterio
- Nivel:** 1
- Temas:** Comprender las ideas fundamentales de la lógica y su uso para resolver problemas del mundo real
- Conocimiento previo:** Elementos de lógica (declaraciones, verdad y la falsedad), lectura y expresión oral
- Notas de Planificación:**
- Los estudiantes tendrán tiempo para aprender sus partes y ensayar la obra con antelación
- Estrategias de Enseñanza/ Aprendizaje:**
- La obra dura aproximadamente 20 minutos.
 - El escenario es un salón de clases, por lo que no se necesitan accesorios especiales o escenografía.
- Evaluación y Valoración:**
- Los estudiantes discuten el misterio y el razonamiento que usaron para resolverlo.
- Necesidades:**
- Los estudiantes reciben copias de la obra, que tiene escenas de 3 y 4 páginas.

ACTIVIDAD: ACORAZADOS

- Duración:** 4 horas
- Descripción:** Las computadoras a menudo tienen que buscar información en grandes colecciones de datos. Necesitan desarrollar formas rápidas y eficientes de hacer esto. Esta actividad demuestra tres métodos de búsqueda -- búsqueda lineal, búsqueda binaria y refundición-- el uso de tarjetas numeradas y el juego de acorazados como vehículos.
- Nivel:** 1 (Grados 3–5)
- Temas:** Comprensión básica de algoritmos de búsqueda
- Conocimiento Previo:** Matemáticas, mayor, menor y la igualdad de las relaciones, geometría (coordenadas)
- Notas de Planificación:** Búsqueda de información de manera eficiente-- búsqueda lineal, búsqueda binaria, refundición
- Estrategias de Enseñanza/ Aprendizaje:**
- 15 niños tienen tarjetas con diferentes números en ellas, ordenadas al azar y ocultas de uno de los niños, que trata de adivinar quien tiene un número misterioso. El juego se repite después de que los 15 números se arreglen en orden.
 - Los niños se agrupan en parejas y a cada pareja se le da dos tarjetas del juego acorazado. El juego se juega con una técnica de hash fáciles para localizar la columna de un barco en la tarjeta.
- Evaluación y Valoración:**
- Las discusiones deben explorar los puntajes logrados por los niños en cada juego.
 - Las discusiones también deben explorar las ventajas y desventajas de cada estrategia de búsqueda.
- Necesidades:**
- Cada niño necesitará una tarjeta del juego acorazados (copiado de maestros)
- Recursos:**
- Ver <http://csunplugged.org/> para aprender sobre esta actividad.

ACTIVIDAD: CONTRA RELOJ

Duración: 4 horas

Descripción: Hay un límite en cuanto a la velocidad con que las computadoras pueden resolver los problemas. Una forma de acelerarlas es utilizar varias computadoras para resolver diferentes partes de un problema. Esta actividad utiliza la clasificación de redes para hacer varias comparaciones al mismo tiempo.

Nivel: 1 (Grados K-2)

Temas: Entender la forma de organizar (clasificar) la información en un orden útil

Conocimiento Previo: Matemáticas de 2do grado; mayor que y menor que

Notas de Planificación:

Algunas tareas se pueden hacer más rápido utilizando menos pasos, mientras que otras se pueden hacer más rápido usando computación paralela. El ordenar las redes es un buen ejemplo de esto.

Estrategias de Enseñanza/Aprendizaje:

- Seis estudiantes tienen un número cada uno y se ordenan del lado izquierdo de la cancha. Se mueven hacia adelante al siguiente círculo de la red y esperan a que alguien llegue.
- El círculo es un punto de decisión de donde el estudiante con el número más pequeño va a la izquierda y el número más grande va a la derecha.
- Al final, los números están ordenados.

Evaluación y Valoración:

- Los estudiantes clasifican correctamente los números usando la red proporcionada.
- También discuten el uso de otras redes para la ordenar.

Necesidades:

Esta es una actividad de grupo al aire libre. Tiza, se necesitan dos juegos de seis tarjetas con números en ellos y un cronómetro. Una maestra se utiliza para dibujar una red de clasificación en la acera.

Recursos:

Ver <http://csunplugged.org/> para aprender más sobre esta actividad.

ACTIVIDAD: COLOREAR POR NUMEROS

Duración: 3 horas

Descripción: La computadora archiva los dibujos, fotografías, texto y otras imágenes usando sólo números. Esta actividad demuestra cómo se hace.

Nivel: 1 (Grados K-2)

Temas: Utilizar 0s y 1s para representar la información

Conocimiento Previo: Geometría de 2do Grado (explorando las formas, conteo, graficas)

Notas de Planificación:

- Las Preguntas para la discusión de motivación incluyen, "¿Qué hace un fax?"
- "¿En qué situaciones almacenaría imágenes la computadora?"
- "¿Cómo almacenan imágenes las computadoras cuando sólo pueden utilizar los números?"

Estrategias de Enseñanza/Aprendizaje:

- Una cuadrícula rectangular de 5x6 se utiliza como base para representar diferentes imágenes (tales como letras) coloreando algunos de los cuadros (píxeles).
- La codificación de la imagen se lleva a cabo mediante el escaneo de las secuencias de 1s (cuadros sombreados) y 0s (cuadros no sombreados) en cada fila de la cuadrícula y registrando la longitud de cada secuencia.

Evaluación y Valoración:

Hojas de Actividades

Necesidades:

No se requieren computadoras, los estudiantes utilizan dos hojas de actividades, llamadas "niño fax" y "Haga su propia imagen"

Recursos:

Ver <http://csunplugged.org/> para aprender más sobre esta actividad.

ACTIVIDAD: EL GATO DANZANTE

Duración:	60 minutos
Descripción:	Los estudiantes construyen un programa de la nada para hacer bailar una sprite
Nivel:	1
Temas:	Reconocer que el software es creado para controlar las operaciones de una computadora de modo secuencial.
Conocimiento Previo:	Los estudiantes deben tener un conocimiento básico de como interactuar con una computadora y sus dispositivos de entrada.

Notas de Planificación:

En preparación a esta actividad los maestros deberán:

- Decidir cómo a él / ella le gustaría que el gato sprite "dance". Un programa de gato danzarín de muestra se explica en la Guía de introducción: <http://scratched.media.mit.edu/resources/getting-started-guide>.
- Decidir cuáles son los conceptos que él / ella desea hacer hincapié en su trabajo para modelar la actividad. Por ejemplo, si los estudiantes son bastante nuevos en la programación y el Scratch, el profesor puede enseñar como arrastrar y hacer clic en los bloques para hacer que el gato sprite realice acciones como mover 10 pasos.

Estrategias de Enseñanza/Aprendizaje:

Esta actividad involucre los siguientes pasos:

- El maestro modela la actividad del Gato Danzante para los estudiantes. Por ejemplo, el maestro puede:
 - Presentar cómo se ve el scratch cuando se abre por primera vez, con la Paleta de Bloques de la izquierda, el área de scripts y el escenario con el gato sprite.
 - Arrastrar un bloque para mover de la paleta de bloques al Área de Scripts.
 - Haga clic en el bloque de movimiento en el área de Scripts. Señale cómo el gato sprite se mueve 10 pasos en el escenario

- Arrastre un bloque game drum de la categoría de sonido y conéctelo al bloque Mover para crear una pila de bloques o un script. Señale la luz blanca que aparece al conectar los dos bloques. Cuando un bloque está siendo arrastrado en el Área de Scripts, la luz blanca indica donde ese bloque puede hacer una conexión válida
 - Haga clic en cualquier lugar de la pila para ejecutar el script. Señale cómo el gato sprite se mueve 10 pasos y luego toca un sonido de batería. Scratch ejecuta un script desde la parte superior de una pila.
 - Construya el resto de la secuencia de comandos de baile con los bloques Move y Play Drum. Para imitar el movimiento de vaivén de la danza, introduzca -10 para el segundo bloque Mover. Después de la adición de cada nuevo bloque, haga clic en la pila y señale las nuevas acciones que el gato sprite realiza en el escenario.
 - Arrastre un bloque eterno de la Categoría de control y conéctelo a la parte superior de la pila, encerrando todos los bloques.
 - Haga clic en la pila de nuevo y señale cómo la secuencia de acciones se repetirá una y otra vez. Arrastre el bloque Bandera Verde Cuando CTR de la Categoría de control y conéctelo a la parte superior de la pila.
 - Haga clic en la bandera verde encima del escenario. Explique como el hacer clic en la bandera verde ejecuta el script.
- Los estudiantes construyen sus propios proyectos de Gato Danzante. El maestro debe animarlos a experimentar, a buscar nuevas formas de expresar el baile o de ampliar su programa como la creación de su propia sprite.
 - Si hay tiempo, los estudiantes deben ver los trabajos de los demás. Los estudiantes también pueden hacer un ejercicio de mostrar y contar, explicando sus proyectos a sus compañeros.
 - Después que los estudiantes completen sus proyectos, el maestro facilita una discusión en torno a algunas de las formas en que los estudiantes programaron su sprite para bailar.

Evaluación y Valoración:

- Si hay tiempo para que los estudiantes caminen alrededor y realicen un ejercicio de mostrar y contar, los estudiantes pueden describir sus proyectos y su proceso en la creación de sus scripts. También pueden evaluar los scripts de sus compañeros.
- En el debate de clausura, los estudiantes pueden articular lo que han aprendido en la creación de su programa Scratch.

Necesidades:

Mientras que esta actividad puede llevarse a cabo sin ningún tipo de materiales complementarios, el profesor puede tener la Guía de introducción disponible para los estudiantes como apoyo en la actividad.

Recursos:

- Guía de introducción: <http://scratched.media.mit.edu/resources/getting-started-guide>
- ScratchEd: <http://scratched.media.mit.edu/>

ACTIVIDAD: PUESTO DE HELADOS

Duración:

4 horas

Descripción:

Un gráfico que se utiliza para representar el mapa de una ciudad. Una compañía de helados quiere construir puestos de helados en diferentes intersecciones, para que sea fácil que la gente llegue a ellos, pero no deben construirse demasiados puestos.

Nivel:

1 (Grados 6–9)

Temas:

Entender el gráfico como una herramienta para representar los estados con problemas y sus soluciones.

Conocimiento Previo:

Lectura Elemental de Mapas

Notas de Planificación:

- Reparta copias de un mapa de una ciudad, donde las líneas representan las calles y los círculos representan las intersecciones.

Estrategias de Enseñanza/Aprendizaje:

- Los niños tienen que determinar el menor número de puestos a construir para que ninguna persona tenga que caminar más de una intersección para comprar helado.

Evaluación y Valoración:

- Los niños discuten diferentes estrategias para la colocación de los puestos y evalúan las soluciones de cada uno.

Necesidades:

Se deben entregar copias de los diferentes mapas de la ciudad.

ACTIVIDAD: EL JUEGO DE LA NARANJA

Duración: 4 horas

Descripción: Esta actividad utiliza un juego simple con naranjas, para ilustrar la administración del tráfico de Internet (routing) y los puntos muertos.

Nivel: 1 (Grados 3–6)

Temas: Algoritmos simples para el routing de la red

Conocimiento Previo: Matemáticas; lógica y razonamiento

Notas de Planificación: Esta es una actividad grupal, que requiere cinco o más niños sentados en un círculo y con diferentes letras en sus camisetas. Hay dos naranjas para la letra de cada niño, excepto uno, para la que sólo hay una naranja.

Estrategias de Enseñanza/Aprendizaje:

- Cada niño recibe una naranja en cada mano (a excepción del niño que solo tiene una naranja) al azar.
- Las naranjas se pasan entre los niños hasta que todo el mundo tiene dos naranjas con su propia letra.
- Only an empty hand can receive an orange, and only from an adjacent hand.

Evaluación y Valoración:

- Los niños deben aprender que quedarse con su propia naranja tan pronto como la reciba podrá impedirle al grupo lograr su objetivo.

Necesidades:

- Una bolsa de naranjas

Recursos:

Ver <http://csunplugged.org/> para aprender más acerca de esta actividad.

ACTIVIDAD: PUEDE VOLVERLO A DECIR

Duración: 4 horas

Descripción: El texto puede comprimirse aprovechando los patrones en las palabras y la vinculación de los patrones de repetición entre sí sin reescribirlos. Por ejemplo, "golpeteo" se puede codificar mediante la sustitución de la última instancia de "teo" por un enlace a la primera instancia

Nivel: 1 (Grados 3–6)

Temas: Desarrollar una comprensión simple de un algoritmo

Conocimiento Previo: Reconocer patrones en las palabras, copiar un texto escrito; familiaridad básica con las computadoras

Notas de Planificación: Las Imágenes y textos que contienen millones de piezas de información se transmiten a través de la Internet todos los días. Para ahorrar tiempo y espacio, se comprimen en formato GIF o ZIP antes de su envío.

Estrategias de Enseñanza/Aprendizaje:

- Los estudiantes codifican y descodifican con éxito el texto, utilizando hojas de cálculo.
- También discuten los tipos de textos e imágenes que se comprimen mejor /peor utilizando este algoritmo.

Evaluación y Valoración:

- Las hojas de trabajo terminadas de los estudiantes son evaluadas.

Necesidades:

Cuatro hojas de trabajo diferentes se utilizan para facilitar esta actividad; una transparencia se utiliza para presentar el algoritmo de compresión

Recursos:

Ver <http://csunplugged.org/> para aprender más sobre esta actividad

A.2 Ejemplos de Actividades para el nivel 2

ACTIVIDAD: CONEXIONES ADENTRO Y AFUERA

Duración: 3 horas y 40 minutos

Descripción: Los estudiantes ven el video El viaje al interior de la computadora (de Intel Corporation (<http://educate.intel.com/en/thejourneyinside/>)) y examinan los componentes internos individuales de los recursos de la computadora. Utilizando los recursos que tienen disponibles los estudiantes descubren la importancia de la cada componente y su impacto en las operaciones de la computadora. La actividad termina con una serie de problemas que los estudiantes deben resolver utilizando este nuevo conocimiento. Finalmente, los estudiantes utilizan esta información para sugerir una colocación alternativa de las computadoras en el entorno escolar lo que tiene un impacto positivo en la comunidad escolar y demuestra el uso inteligente de los recursos.

Nivel: 2

Temas: Los estudiantes obtendrán un entendimiento conceptual de los principios de organización de la computadora y los componentes principales (de entrada, de salida, memoria, almacenamiento, procesamiento, software, sistemas operativos).

Los estudiantes obtendrán una comprensión conceptual de los componentes básicos de las redes de las computadoras (servidores, protección de archivos, bobinas y listas de espera, los protocolos de enrutamiento de la conexión, la comunicación, recursos compartidos y la tolerancia a fallos).

Conocimiento Previo: Las diferencias entre el hardware y el software:

- Capacidad para registrar los resultados de la observación,
- Familiaridad con el Sistema operativo que están utilizando y la red temporal, y
- Familiaridad con los componentes internos y sus usos.

Notas de Planificación:

- Solicitar permiso para que los estudiantes visiten ciertas áreas de la escuela durante el horario de clase como un viaje de estudio dentro de la escuela.
- Piense de visitar un laboratorio midi de música, laboratorio de comunicación, oficina y cualquier recurso especializado relativo a su entorno local.
- Consulte con el administrador del sitio si no está seguro del tipo (s) de red disponible en la escuela.
- Preparar una lista de verificación de términos para uso de los estudiantes durante el video.
- Haga arreglos para que el administrador del sitio de la computadora de la escuela o la oficina o un técnico en computación hable a la clase sobre las redes y sistemas operativos
- Tener un mapa escolar disponible para que los estudiantes hagan una gira y una transparencia del mapa para repasar.
- Compruebe si los materiales de El viaje al interior de la computadora están disponibles de Intel (Intel Corporacion El viaje interior. Parte del Kit del Viaje dentro de la Educación.)

Estrategias de Enseñanza/Aprendizaje:

- Mostrar el video del viaje dentro de la computadora, Unidad 4 de Microprocesadores, después la unidad 6 en creación de redes, con el propósito de revisar los componentes del equipo y extender el conocimiento del estudiante sobre las redes y los sistemas operativos.
- Recoger la hoja de términos y hacer que los estudiantes completen las definiciones de las palabras que no conocen (los profesores pueden introducir al alumno el diccionario en línea en www.dictionary.com).
- Compartir información sobre las redes con los estudiantes.
- Indicar el tipo (s) de redes que actualmente se utilizan en el entorno escolar.
- Compartir información sobre los sistemas operativos con los estudiantes.
- Dar una prueba corta sobre redes y sistemas operativos.
- Proporcionar a cada estudiante con un mapa de la escuela y explicarla ruta del recorrido y las rutinas especiales requeridas para las áreas seguras.
- Dar a los estudiantes una clave sencilla para marcar el mapa (por ejemplo, C = computadora que destaca por sí sola, L = lab, SL = laboratorio especializado, S = cuarto del servidor, P = recurso de impresora).
- Regreso a las aulas y revisión del mapa en una transparencia, con el aporte de los estudiantes.
- Fomentar una discusión de cómo las mejoras que se han hecho en las redes y los sistemas operativos hacen una diferencia en una comunidad computarizada como una escuela.
- Pídeles que reflexionen sobre por qué creen que los recursos de la computadora se han colocado en la escuela de la manera en que están.
- Pida a los estudiantes que preparen un escrito breve sobre los cambios que les gustaría ver en el entorno de computación de la escuela.
- Pida a los estudiantes que incluyan los efectos positivos que sus sugerencias tienen sobre el ambiente escolar e incorporar sus conocimientos sobre redes.
- Facilitar al estudiante pareja/ plaza y parte de las sugerencias.

Evaluación y Valoración:

- Una evaluación formativa en el uso de la hoja de repaso de términos y
- Una evaluación de la prueba en las redes y sistemas operativos.

Necesidades:

- Aplique un examen oral, si fuese apropiado
- Proveer a los estudiantes con discapacidades físicas asistencia, si es necesario.
- Ayudar a los estudiantes con necesidades especiales con la hoja de términos.

ACTIVIDAD: HERRAMIENTAS DE DISEÑO CULTURALMENTE SITUADAS—TEJIDO

Duración: 4 horas

Descripción: En esta lección, los estudiantes aprenden cómo las computadoras pueden ser utilizadas como una herramienta para la visualización de datos, modelado y diseño y el arte en el contexto culturalmente situado de las herramientas de diseño. Se explorarán las conexiones entre el diseño de las herramientas y las matemáticas.

Nivel: 2

Temas:

Interactuar con los modelos específicos de contenido y simulaciones (por ejemplo, los ecosistemas, las epidemias, la dinámica molecular) para apoyar el aprendizaje y la investigación. Haga la conexión entre los elementos de las matemáticas y ciencias de la computación, incluyendo los números binarios, lógica, conjuntos y funciones. En colaboración diseñar, desarrollar, publicar y presentar productos (por ejemplo, vídeo, podcasts, wikis, etc.) utilizando los recursos tecnológicos que demuestran y comunican conceptos curriculares.

Conocimiento Previo:

Comprensión del sistema de coordenadas cartesianas.

Notas de Planificación:

- Asegúrese de que el sitio web de las Herramientas de Diseño Culturalmente Situadas (csdt.rpi.edu) esté disponible para todos los estudiantes.
- El tutorial del telar de cuentas está en línea. Usted puede adaptar la herramienta del Pacific Northwest Basket Weaver y la de Navajo Rug Weaver del tutorial del telar de cuentas y proporcionar copias impresas.
- Crear una rúbrica para proyecto y proporcionarla a los estudiantes con antelación
- Repasar el Sistema de Coordenadas cartesianas

Estrategias de Enseñanza/Aprendizaje:

- Publicar las posibles herramientas de diseño:
 - Telar de Cuentas Virtual
 - Tejedor de canastas Pacific Northwest
 - Tejedor de Alfombras Navajo
- Mostrar la primera página de cada herramienta con el fin de dar a los estudiantes una idea de lo que hace cada una. (<http://csdt.rpi.edu>) Los alumnos se dividen en grupos para trabajar en la herramienta de su elección. El tamaño del grupo dependerá del tamaño de la clase. Puede que haya más de un grupo por cada herramienta.
- Cada miembro del grupo debe ir a través de toda la sección cultural de forma individual.
 - Responder a cualquier pregunta que se plantea en su cuaderno.
 - Busque y anote las conexiones matemáticas.
- Todos los miembros del grupo discuten la sección.
 - Resolver las respuestas a las preguntas y conexiones matemáticas.
- Cada miembro del grupo completa el tutorial.
 - Los estudiantes deben hacer el tutorial a su propio ritmo, pero discutir con otros miembros cuando surgen preguntas.
 - Animar a los estudiantes a registrar en su diario la información que quieran recordar.
- Los grupos crean los diseños utilizando el software de la herramienta de diseño.
 - Cada persona debe elegir una de las imágenes objetivo para la práctica y discutir cualquier tema con los otros miembros del grupo.
 - Los grupos deciden si desean crear un diseño como grupo o tener múltiples diseños para su presentación.
 - Los grupos trabajan en el diseño (s) estos deben ser creaciones propias, más que una imitación de los diseños precargados.
- Editar los diseños con Photoshop Express.
 - Pida a los estudiantes que vean el tutorial en línea y abran una cuenta.
 - Editar el diseño
- Prepare las presentaciones para que incluyan:
 - Cultura
 - Conexiones matemáticas
 - Demo de software
 - Presentación de diseños
- Los grupos hacen sus presentaciones.

Técnicas de Evaluación y Valoración:

Los estudiantes son evaluados en sus proyectos, incluyendo diseño, descripción de la cultura, conexiones matemáticas y demostración. Los estudiantes también deben ser evaluados por su trabajo en equipo.

Necesidades:

- Utilizar numerosos recursos visuales y demostraciones para ayudar a los estudiantes según sea necesario.
- Proporcionar dispositivos de adaptación apropiados o implementación de necesidades para estudiantes identificados.
- Construir grupos de estudiantes para que haya asistencia en caso necesario

Recursos:

- Herramientas de diseño culturalmente situadas: <http://csdt.rpi.edu/> (sitio y adaptaciones de tutoriales cortesía de Ron Eglash)
 - Tutorial del Telar de cuentas virtual

- Tutorial del Tejedor de Canastas Pacific Northwest
 - Tutorial del tejedor de Alfombras Navajo
 - Proyecto muestra de la rúbrica de las Herramientas de diseño culturalmente situadas
- Photoshop Express: <http://photoshoponline.com/>

ACTIVIDAD: COMPARAR EL SOFTWARE DE PRODUCTIVIDAD DE CODIGO ABIERTO CON EL SOFTWARE COMERCIAL

Duración: 4 horas

Descripción: Los estudiantes desarrollan una comprensión del software de productividad de código abierto mediante la comparación con sus contrapartes comerciales. Varias características del software de productividad serán exploradas. Los estudiantes tienen la oportunidad de comparar y analizar los resultados de sus hallazgos.

Nivel: 2

Temas: La comparación del software de productividad de código abierto con sus contrapartes comerciales.

Conocimiento previo: Conocimiento práctico del software de productividad comercial.

Notas de Planificación:

- Revisar el propósito y las características de los diversos tipos de software de productividad: procesadores de texto, hojas de cálculo, presentaciones
- Instale software de productividad comercial, como Microsoft Office.
- Instale software de productividad de código abierto, como Open Office.
- Visit openoffice.org or appropriate URL for your open source software and review different kinds of open source productivity software available. Visitar openoffice.org o el URL apropiado para su software de código abierto y revisar los diferentes tipos de software de productividad de código abierto disponibles.

Estrategias de Enseñanza/Aprendizaje:

- Seleccione un conjunto de tareas a realizar (por ejemplo: crear un folleto en el software de procesamiento de textos, crear una tabla y un gráfico en una hoja de cálculo, crear una presentación de diapositivas en el software de presentaciones).
- Divida la clase en equipos de dos.
- En cada pareja, haga que un estudiante utilice los productos comerciales y el otro utilice los productos de código abierto para completar una tarea asignada.
- Una vez completado, haga que los estudiantes comparen los resultados e identifiquen las similitudes y diferencias en el uso del software y en el resultado final.
- Cada pareja de estudiantes realiza un informe indicando sus hallazgos. El informe cubrirá las siguientes categorías: tarea (s) completada, tipos de software utilizados, características utilizadas en el software, similitudes / diferencias. El informe debería concluir con un enunciado sobre cuál es el software que el equipo prefiere y por qué.
- Una vez que los informes se someten y /o se presentan a la clase, se puede dar una discusión abierta sobre los beneficios e inconvenientes de la utilización de productos de código abierto frente a los productos comerciales. La discusión puede incluir pero no se limita a los costos iniciales, costos de licencias y los gastos de apoyo de ambos tipos de software.

Técnicas de Evaluación y Valoración:

- Informe sobre las similitudes y diferencias entre el software comercial y de código abierto.
- Participación en la discusión en clase.

Necesidades:

- Las instrucciones pasó a paso sobre cómo llevar a cabo una tarea determinada en un procesador de textos, hoja de cálculo y software de presentaciones.
- Manual del usuario o Ayuda sobre el uso de software de productividad.

ACTIVIDAD: SISTEMAS NUMERICOS

Duración: 4 horas

Descripción: Los estudiantes desarrollan una comprensión de la relación entre el sistema de numeración binaria y la lógica de la computadora. Además, los estudiantes aprenden cómo convertir los números Base 10 números en binarios y viceversa. Se explora la representación en caracteres de los códigos binarios.

Los estudiantes tienen la oportunidad de experimentar por escrito su propio mensaje y decodificación.

Nivel: 2

Temas: La conexión entre los elementos de matemáticas y ciencias de la computación, incluyendo los números binarios, lógica, conjuntos y funciones.

Conocimiento Previo: Comprensión del sistema numérico decimal y posición de valor

Notas de Planificación:

- Revisar cómo el software de programación maneja las representaciones de caracteres.
- Tener ocho centavos por cada pareja de estudiantes y un folleto y / o una sobrecarga de información de bit
- Revisar las conversiones binarias y base 10.
- Preparar mensajes codificados para que los estudiantes los descifren.
- Tener copias del código ASCII disponibles (estándar y extendido).

Estrategias de Enseñanza/Aprendizaje:

- Muestre el segmento 3 del vídeo de viaje al interior (8 min. 25 seg. en Corporación Intel. El viaje interior. Parte del Kit de Viaje Dentro de la Educación) o cualquier otro vídeo que muestre cómo las computadoras vuelven las imágenes y colores en códigos. Los estudiantes comprenden cómo la información se comunica a través del uso de códigos.
- Entregue a cada pareja de estudiantes ocho peniques y trabaje en las preguntas sobre información bit. Pregunte a los alumnos qué patrón pueden ver formándose en la columna de la derecha (números de dobles).
- Los estudiantes tienen el reto de contar lo más alto posible con una mano y se les dice que la respuesta es mayor que 10. Mientras que los estudiantes reflexionan sobre el reto, los maestros demuestran, con ayuda de un circuito en serie simple, los estados lógicos binarios de UNO y CERO (VERDADERO y FALSO, ALTO y BAJO) igualándolos a una condición de serie de circuitos de lámpara ENCENDIDO y APAGADO.
- Los números binarios se introducen iniciando el conteo con los dedos de una mano—sin levantar los dedos significa cero, el pulgar hacia arriba es un uno, el dedo índice hacia arriba es dos, el dedo medio para arriba es un cuatro, el dedo anular es ocho y el dedo meñique representa dieciséis. Los estudiantes demuestran que pueden contar hasta 31 con una sola mano.
- Esto prepara el escenario para demostrar cómo convertir los números de base 10 a base 2 (binario). Trabaje varios ejemplos con los estudiantes.
- Dar a los estudiantes una prueba sobre conversión binaria para evaluar su comprensión del concepto.
- Hoja impresa de la información de conversión ASCII. Dado que las computadoras no pueden pensar como nosotros, necesitan un código para traducir nuestro lenguaje a datos que puedan procesar y luego convertirlos a un lenguaje reconocible.
- Los estudiantes completan ejercicios de conversión.

Técnicas de Evaluación y Valoración:

- La evaluación formativa de la prueba al final del ejercicio de conversión binario motiva a los estudiantes a progresar y mostrar los cambios necesarios para el éxito de la aplicación de conversión.
- Evaluación sumaria de los ejercicios de conversión.

Necesidades:

- Utilice ayudas visuales extensas y demostraciones para ayudar a los estudiantes a medida sea necesario.
- Proporcionar una copia ampliada de la metodología de conversión en el aula, así como la carta de caracteres ASCII.
- Usar una variedad de estilos de enseñanza para adaptarse a los estilos de aprendizaje.
- Provide appropriate adaptive devices or implementation accommodations for identified students. Proporcionar dispositivos de adaptación apropiados o implementar necesidades para los estudiantes identificados.

ACTIVIDAD: PASALO

Duración: 90 minutos

Descripción: En esta actividad, los estudiantes colaborativamente y gradualmente construyen proyectos utilizando Scratch pasando los proyectos de estudiante a estudiante.

Nivel: 2

Temas: Colaborar con compañeros, expertos y otras personas con prácticas colaborativas tales como la programación en parejas, trabajar en proyectos en equipo y participar en actividades grupales de aprendizaje activo (por ejemplo, CS desenchufado y Klas).

Diseñar, desarrollar, publicar y presentar productos (por ejemplo, páginas web, aplicaciones móviles, animación) utilizando recursos tecnológicos que demuestran y comunican conceptos curriculares.

Conocimiento Previo:

Los estudiantes deben tener al menos el conocimiento básico de Scratch y su mecanismo para encajar bloques juntos para especificar las instrucciones del programa.

Notas de Planificación:

Al prepararse para esta actividad, los maestros deben:

- Seleccione un tema para el proyecto de pásalo. Algunos de los temas que hemos tratado antes y han sido populares incluyen historias de pásalo y fiestas de baile de pásalo. El contenido de los proyectos de pásalo también se puede conectar a las diversas áreas curriculares.
- Decidir qué características de Scratch y que conceptos computacionales serán demostrados en la introducción de apertura.

Estrategias de Enseñanza/Aprendizaje:

Esta actividad involucra los siguientes pasos:

- El maestro modela la actividad comenzando con el proyecto de Pásalo. Por ejemplo, el maestro puede demostrar cómo iniciar un proyecto de fiesta de baile, añadiendo música, un fondo y un fiestero. El profesor explica el proceso de Pásalo y como ejemplo, hace (o solicita) dos o tres sugerencias para elementos que puedan añadirse al proyecto de modelo posteriormente (por ejemplo, los asistentes adicionales a la fiesta, más trajes).
- Los estudiantes (ya sea individualmente o en parejas) utilizan una computadora para iniciar sus proyectos de pásalo. Tienen 15 minutos para trabajar en sus proyectos.
- Después de 15 minutos, los estudiantes rotan a un nuevo equipo y continúan construyendo el proyecto que encuentran en el nuevo equipo. Tienen 15 minutos para trabajar en sus proyectos.
- Después de 15 minutos, los estudiantes rotan una última vez y trabajan en un tercer proyecto. Tienen 15 minutos para trabajar en sus proyectos.
- Al final de los 15 minutos finales, los estudiantes regresan a su equipo original para ver cómo el proyecto que comenzaron ha evolucionado.
- En función del número de proyectos, los estudiantes pueden caminar alrededor para ver todo (o algún subconjunto) de los otros proyectos de pásalo.
- El profesor facilita una discusión acerca de los conceptos y características que los estudiantes aprendieron mientras trabajaban en los proyectos y miraron el código de los demás.

Evaluación y Valoración:

- La discusión de cierre: La discusión puede ser utilizada para hacer explícitos los diferentes conceptos y características que los estudiantes aprendieron y los retos que experimentaron.
 - El proyecto de Scratch, Pásalo: Los proyectos pueden ser evaluados por procedimiento (por ejemplo, la naturaleza de la actividad colaborativa, como los estudiantes se apoyaron el aprendizaje de los demás) y producto (por ejemplo, la función y los conceptos incluidos en el proyecto, las formas en que el tema fue creativamente apropiado).
- Esta evaluación puede incluir la autoevaluación, evaluación por pares y la evaluación-instructor.

Necesidades:

Aunque esta actividad se puede completar sin materiales complementarios, los conceptos y las características demostradas en el modelo inicial del maestro se puede apoyar con folletos que sugieran que cosas probar. Por ejemplo, los siguientes folletos podrían estar accesibles cerca de los estudiantes mientras ellos trabajan en sus creaciones:

- Tarjetas de Scratch :Un conjunto de 12 fragmentos <http://scratched.media.mit.edu/resources/scratch-cards>
- El negocio del mono: Una sugerencia de una página de bloques introductorios para experimentar con. <http://scratched.media.mit.edu/resources/monkey-business>

Recursos:

- Video, documentar la actividad de la "historia de Pásalo", como se hace con un grupo de chicas adolescentes. <http://scratched.media.mit.edu/stories/csed-week-day-4-wise-dance-party>
- ScratchEd comunidad en línea. <http://scratched.media.mit.edu/>

ACTIVIDAD: CONFIGURANDO UNA COMPUTADORA**Duración:**

2 horas y 30 minutos

Descripción:

Los estudiantes configuraron una computadora, incluyendo la instalación del software disponible y un sistema operativo. Los estudiantes conectan, configuran y prueban todos los periféricos. Por último, los estudiantes solucionan cualquier problema que surja. Todos los estudiantes configuraron una PC.

Nivel: 2

Temas: Los estudiantes obtendrán un entendimiento conceptual de los principios de organización de la computadora y los componentes principales (es decir, de entrada, de salida, memoria, almacenamiento, procesamiento, software, sistemas operativos).

Conocimiento Previo: Componentes de un sistema de computación, la terminología correcta

Notas de Planificación:

- Preparar las muestras disponibles de micro controladores y PC de varios tipos.
- Determinar el uso más eficaz del hardware existente dentro del tiempo recomendado asignado (por ejemplo, de dos a tres estudiantes por computadora).
- Abra un disco duro viejo descartado para fines de demostración.
- Si los recursos son limitados, un solo sistema puede ser configurado en varias ocasiones para acomodar a todos los estudiantes.
- Esta actividad se realiza con máquinas individuales para no interrumpir un entorno de red.
- El maestro debe revisar los procedimientos en los anexos adjuntos. Esta actividad asume que el disco duro del sistema de la computadora se ha configurado anterior a la instalación del sistema operativo.
- La instalación real del sistema se puede realizar como una clase de "caminar a través." El maestro puede modificar el proceso para que grupos individuales realicen la tarea de configuración.
- El maestro debe revisar las particiones del disco, el formato, las operaciones de escaneo del disco y la información disponible en los archivos de ayuda del sistema operativo (ver Recursos).
- Inventario del CD-ROM del sistema operativo y la llave del software.
- Asegúrese de que todo el software está disponible para la instalación completa, incluyendo sistemas operativos, controladores de dispositivos y software de aplicación.

Estrategias de Enseñanza/Aprendizaje:

- Los maestros y los estudiantes revisan la seguridad con la electricidad estática y la importancia de mantener los contactos limpios ya que se aplican a los componentes. Revise las consideraciones de seguridad al instalar un equipo de escritorio (enchufes conectados a tierra, el uso de barras de energía, cables que cuelgan, eliminar el peligro de la electricidad estática y desconectar la fuente de alimentación antes de abrir la PC, etc.)
- El maestro explica cómo funcionan los discos duros para que los estudiantes puedan comprender las funciones de utilidad que se requieren para completar la actividad.
- Los estudiantes usan el equipo necesario para completar la tarea, incluyendo el monitor, CPU, teclado, ratón y una impresora, si está disponible. El maestro explica algunas consideraciones especiales que necesitan saber (por ejemplo, colocación de equipos para los enchufes en la habitación). Los estudiantes utilizan esta información para crear una lista de control para la actividad.
- Dependiendo de los recursos disponibles, se dividen los alumnos en el número apropiado de grupos. Los estudiantes conectan todas las partes de su sistema computarizado. Circule para ayudar a solucionar problemas y utilice técnicas de interrogatorio para ayudar a resolver problemas
- Una vez que todos los componentes están conectados, los estudiantes cargan el software del sistema operativo. Los estudiantes completan su lista personal de verificación.
- Todos los grupos deben poner a prueba su software para asegurarse de que su sistema está funcionando y que todos los periféricos conectados funcionan correctamente.

Técnicas de Evaluación y Valoración:

- Una evaluación formativa a través de la discusión de los estudiantes y la observación, anima a los estudiantes a evaluar su forma de pensar para finalizar con éxito de la tarea.
- Assess student-created checklists. Provide students with written/oral feedback, to assist their success in upcoming related activities. Evaluar las listas de verificación creados por los estudiantes. Proporcionar a los estudiantes retroalimentación por escrito /oral, para ayudar en su éxito en próximas actividades.

Necesidades:

- Provea instrucciones paso a paso.
- Provea un glosario de términos.
- Provea visuales de los diferentes tipos de computadoras.

A.3 Ejemplo de Actividades para el Nivel 3

ACTIVIDAD: CARRERAS EN INGENIERIA EN SISTEMAS

Duración: 3 horas y 45 minutos

Descripción: Se invita a un orador invitado para compartir información acerca de su / su carrera con los estudiantes. Los estudiantes amplían su conocimiento en la industria de la computación. Los estudiantes miran los grados y las certificaciones disponibles y las oportunidades que tienen a nivel de la escuela secundaria y más allá para guiarlos hacia las carreras en la industria de la computación. 3A, 3B (se puede utilizar para el nivel 2 con algunas modificaciones)

Nivel:

Tema: Los estudiantes obtendrán una comprensión conceptual de la identificación de las diferentes carreras en las ciencias de la computación y su conexión con los temas estudiados en este curso (por ejemplo, especialista en tecnología de la información, diseñador de página web, analista de sistemas, programador, CIO).

Conocimiento Previo: Habilidades de Procesamiento de texto

Notas de Planificación:

- Entre los oradores invitados incluya al administrador del sistema escolar, al técnico del distrito o alguien de la comunidad local.
- Recopilar información de una universidad o colegio comunitario local, incluyendo los calendarios del curso escolar y catálogos colegio / universidad.
- Reúna copias de las últimas revistas especializadas en computación.
- Arregle con anticipación para que un estudiante presente los oradores invitados y otro estudiante para agradecerles.
- Recoger los anuncios de periódico para puestos de trabajo en la industria de la computación.
- Distribuya una muestra de la hoja de trabajo de certificación

Estrategias de Enseñanza/Aprendizaje:

- Los maestros introducen las expectativas de la actividad.
- Los maestros revisan con los estudiantes (anticipadamente) las técnicas para de interrogar al orador invitado.
- Un estudiante puede presentar al orador invitado. Los estudiantes toman notas breves con el fin de hacer preguntas relevantes e interesantes.
- Discuta la información del orador invitado con los estudiantes para que después escriban sus ideas acerca de la información.
- Los estudiantes revisan las revistas especializadas para ver los avances en la industria de la computación. Cada estudiante elige un artículo de una revista para resumir o revisar utilizando el procesador de textos.
- Finalmente, los alumnos miran las oportunidades para las diferentes denominaciones de computadoras que van desde MCSE (Microsoft Certified Engineer) hasta la ingeniería en sistemas en el ámbito universitario. Los estudiantes utilizan los anuncios de periódico para explorar cuáles son las habilidades y denominaciones solicitadas por los posibles empleadores.
- Los Estudiantes recuperan el archivo gráfico de la certificación (por medios electrónicos o por medio de la hoja informativa) y utilizando las denominaciones que descubrieron en el anuncio del ejercicio anterior, completan el cuadro.
- Los estudiantes crean un plan sobre la manera de seguir una carrera en computación, a partir de la finalización de este curso y guardan la información en su portafolio (objetivo a largo plazo).

Evaluación y Valoración:

- Proporcione retroalimentación acerca de la culminación y comprensión de las tareas dadas.
- Evalúe el resumen del artículo utilizando una rúbrica que usted ha creado.

Necesidades:

- Permitir cronogramas flexibles para la fecha de entrega del informe.
- Utilice vídeos centrados en carreras, si están disponible.
- Invite al personal de recursos del Servicios Estudiantiles al aula.
- Grabe en video la presentación del orador invitado para permitir a los estudiantes la oportunidad de verlo de nuevo.

ACTIVIDAD: HERRAMIENTAS DE DISEÑO CULTURALMENTE SITUADAS—CURVA DE CONROW

Duración: 3 horas

Descripción: Esta lección sirve como refuerzo del proceso y las conexiones entre las matemáticas y la computación en el contexto de una herramienta de diseño culturalmente situada para resolver problemas.

Nivel: 3A, 3B

Temas: Explicar cómo la selección, iteración y recursión de la secuencia son bloques de construcción de los algoritmos; Describir cómo la computación comparte características con el arte y la música mediante la traducción de la intención humana en un artefacto; Describir cómo la computación aumenta las formas tradicionales y permite nuevas formas de experiencia, expresión, comunicación, y colaboración.

Conocimiento Previo: Comprensión del Sistema de Coordenadas Cartesianas.

Notas de Planificación:

- Asegúrese de que csdt.rpi.edu es accesible a todos los estudiantes
- Crear una rúbrica para el proyecto y proporcionarla a los estudiantes con anticipación
- Revisar el sistema de coordenadas cartesianas

Estrategias de Enseñanza/Aprendizaje:

- Antecedentes culturales del entrelazado cornrow
 - Los estudiantes leen los antecedentes culturales y cómo entrelazar secciones (csdt.rpi.edu, curvas Cornrow).
 - Discusión grupal sobre los antecedentes culturales del entrelazado de cornrow
 - Divida a los estudiantes en grupos de 3-4 y pida a cada grupo que reflexione sobre una de las siguientes secciones:
 - Orígenes Africanos
 - Travesía del Atlántico
 - Guerra Civil a los Derechos Civiles
 - Hip Hop
 - Cada grupo comparte su respuesta con el resto de la clase.
 - Herramienta de Diseño tutorial para las curvas de Cornrow
 - Cada estudiante completa la Parte I del tutorial siguiendo todas las instrucciones y la comprobación de su trabajo con su pareja en codo.
 - Discuta cualquier problema, como clase, antes de proceder a la Parte II.
 - Completar la Parte II del tutorial de diseño.
 - Refuerce las matemáticas y la investigación estructurada.
 - Proyecto de las Curvas de Cornrow
 - Los estudiantes crean su propio diseño.
 - Describir cada paso del proceso de resolución de problemas utilizado.
 - Destacar los conceptos matemáticos usados y dónde se utilizaron.
- Reforzar la estrategia de buscar de un problema similar que ya se ha resuelto para ayudar a resolver el nuevo Problema

- Paseo por la Galería de Diseños
 - Los estudiantes comparten sus soluciones.

Evaluación y Valoración:

- Los estudiantes son evaluados en sus proyectos, incluyendo el diseño, descripción del proceso utilizado y la descripción de los conceptos matemáticos utilizados.

Necesidades:

- Utilice ayudas visuales y demostraciones extensas para ayudar a los estudiantes según sea necesario.
- Proporcionar dispositivos de adaptación apropiados o adaptaciones necesarias para los estudiantes identificados.
- Construir grupos de estudiantes para asistir en caso necesario

Recursos:

- Herramientas de diseño culturalmente situadas para las curvas de Cornrow-csdt.rpi.edu (cortesía de Ron Eglash)

ACTIVIDAD: SOLUCIONES NUEVAS PARA PROBLEMAS VIEJOS

Duración: 5 horas

Descripción: Los estudiantes examinan los problemas que pueden ser resueltos utilizando más de un algoritmo (por ejemplo, la determinación del valor factorial de un número). Utilizar la lluvia de ideas u otras técnicas de resolución de problemas en grupo, los estudiantes desarrollan algoritmos alternativos que utilizan técnicas recursivas y no recursivas. Los estudiantes identifican los componentes de un algoritmo recursivo y desarrollan criterios para reconocer cuándo se puede aplicar un algoritmo recursivo.

Nivel: 3A, 3B

Temas: Las ideas fundamentales sobre el proceso de diseño del programa y la solución de problemas, incluyendo el estilo, la abstracción, y las discusiones iniciales de la corrección y la eficiencia como parte del proceso de diseño de software.
Estructuras simples de datos y sus usos.

Conocimiento Previo: Uso de modelos para resolver problemas, la capacidad de desarrollar algoritmos apropiados para resolver los problemas y la capacidad de escribir un pseudocódigo.

Notas de Planificación:

- Revisar la naturaleza de la recursividad.
- Reunir ejemplos de problemas que pueden ser resueltos utilizando más de un método, incluyendo la recursividad y determinar qué problemas se pueden resolver utilizando un algoritmo recursivo.

Estrategias de Enseñanza/Aprendizaje:

- Divida la clase en grupos de dos o tres estudiantes.
- Revise la técnica de lluvia de ideas para resolver problemas.
- Presentar un problema que puede ser resuelto mediante un algoritmo familiar aunque complejo pero también puede ser resuelto mediante un algoritmo menos familiar pero más simple (por ejemplo, determinar el cociente y el resto de la división de dos números enteros).
- Los estudiantes, en sus grupos, desarrollan más de un algoritmo para la solución.
- El maestro facilita una discusión en clase para desarrollar criterios para evaluar los algoritmos, incluyendo la eficiencia de la solución y la complejidad de la codificación requerida. Se consideran ambas eficiencias, la del procesamiento y la interfaz de usuario.
- Los Grupos evalúan los algoritmos utilizando los criterios desarrollados y comparten sus algoritmos y evaluaciones con la clase.
- El maestro presenta el método recursivo para resolver problemas e ilustra un algoritmo recursivo para la solución a un problema diferente (por ejemplo, el cálculo del valor factorial de un número).
- Los Grupos desarrollan un algoritmo recursivo para el problema inicial y evalúan su eficacia.
- El maestro facilita una discusión en clase para establecer los criterios para determinar si un algoritmo recursivo es una solución apropiada e identifica problemas adicionales que pueden ser resueltos mediante la recursividad.
- Trabajando en grupos, los estudiantes desarrollan algoritmos recursivos y no recursivos para resolver problemas adicionales que se asignan.

Evaluación y Valoración:

- Se realiza una evaluación formativa de los trabajos asignados en clase en la forma de conferencias itinerantes, y
- Una evaluación sumaria en la que los estudiantes completan una tarea que requiere el desarrollo de un algoritmo recursivo y un algoritmo no recursivo.

Necesidades:

- Proporcionar copias impresas de ejemplos de algoritmos utilizando métodos recursivos y no recursivos, incluyendo ilustraciones gráficas y modelos para ilustrar los algoritmos.
-

ACTIVIDAD: PLANEANDO UNA SOLUCION

Duración: 6 horas

Descripción: Los estudiantes trabajan en grupos para analizar los problemas complejos (por ejemplo, Torres de Hanoi) y desarrollar algoritmos apropiados utilizando técnicas recursivas y no recursivas. Los estudiantes crean un pseudocódigo y gráficos de diseño para ayudarles en la planificación de una solución y evaluar estas representaciones de código como herramientas para resolver problemas.

Nivel: 3A, 3B

Temas: Las ideas fundamentales sobre el proceso de diseño del programa y la resolución de problemas, incluyendo estilo, abstracción y las discusiones iniciales de corrección y eficiencia como parte del proceso de diseño de software.

Conocimiento Previo: Los estudiantes pueden aplicar los pasos en el ciclo de vida del diseño de software, el uso de pseudocódigo, diagramas y gráficas para resumir el diseño del programa y desarrollar algoritmos apropiados para resolver problemas.

Notas de Planificación:

- Revise la solución de problemas de arriba hacia abajo.
- Seleccione un problema para usar en el desarrollo de un modelo de solución y prepare los modelos adecuados.

Estrategias de Enseñanza/Aprendizaje:

- La clase se divide en grupos de dos o tres estudiantes y a cada grupo se le asigna un problema.
- Los Grupos investigan el problema utilizando una variedad de técnicas para resolver problemas, para analizarlo.
- Cada grupo utiliza la lluvia de ideas u otras técnicas de resolución de problemas para desarrollar un algoritmo para la solución al problema. En la discusión en clase, los grupos presentan y comparten sus algoritmos.
- Los estudiantes comparan la eficacia y eficiencia de los algoritmos presentados, luego los grupos refinan sus algoritmos.
- Cada grupo desarrolla un diagrama de flujo, uno de estructura y / o un pseudocódigo para representar la aplicación del algoritmo. El maestro consulta con cada grupo para discutir y evaluar el diseño de la solución.

Evaluación y Valoración:

- Una evaluación formativa de los compañeros de los algoritmos presentados.
- Una evaluación formativa del diseño para la solución del problema.

Necesidades:

- Proporcionar una muestra impresa de algoritmos similar a los estudiados.
 - Utilizar modelos gráficos para ilustrar el problema.
 - Selectivamente emparejar alumnos / grupo para ayudar a la resolver los problemas.
 - Proporcionar problemas de diversa complejidad para proporcionar un desafío apropiado.
-

ACTIVIDAD: FUNCIONES AUXILIARES JUEGO DE ROLES/RECURSIVIDAD

Duración: 1 hora

Descripción: Los Estudiantes juegan el rol de diversos objetos de programas sencillos para entender el paso de parámetros y llamadas recursivas

Nivel: 3B, 3C.1

Temas: Métodos (funciones) y parámetros, recursividad

Conocimiento Previo: Compilar y ejecutar programas sencillos, escribir códigos utilizando parámetros.

Notas de Planificación:

- Preparar u obtener de recursos guiones para juegos de rol con objetos pequeños en un programa con varios anidada (Y por lo general también recursivo) llamadas.
- Reúna marcadores de colores y cartulina, según sea necesario.

Estrategias de Enseñanza/Aprendizaje:

- Revisar los conceptos de constructores, parámetros y solicitar métodos de ayuda.

- Los estudiantes leen el código para el programa que se utilizará para el juego de roles
- Seleccione a un estudiante para desempeñar el papel de la función principal. Si lo desea, dar a los estudiantes una etiqueta grande con el nombre para que la usen. Seleccione otro estudiante para que monitoree el código cuyo trabajo consiste en llevar un registro de la línea de código que se está ejecutando actualmente.
- Ayude a la clase, si actúan fuera del guion, cada vez que un objeto se construye el estudiante llamando al constructor recoge un compañero de clase para interpretar el papel del objeto, si utilizan las etiquetas de nombre, asegúrese de darle a cada objeto -jugador una etiqueta con su nombre. Es útil utilizar etiquetas de diferentes tamaños, formas o color para diferentes clases.
- Haga pausas frecuentes en el juego y pida a los miembros de la audiencia que identifiquen quién será el próximo actor.

Evaluación y Valoración:

- Se hace una evaluación formativa del juego de roles en forma de entrevistas itinerantes.
- Una evaluación sumaria se puede administrar pidiendo a los estudiantes que indiquen el número de objetos en existencia a medida que progreso la obra y preguntas similares.
- Deje que pares de estudiantes actúen un papel.
- Asigne los roles usted mismo dando papeles simples a los estudiantes que están teniendo problemas.

Recursos:

- Varios ejercicios de juegos de rol están disponibles en: <http://cs.colgate.edu/APCS/Java/RolePlays/JavaRolePlays.htm>

ACTIVIDAD: RETOS CORTOS DE PROGRAMACION EN SCRATCH

Duración: 90 minutos

Descripción: Los estudiantes construyen soluciones para retos de programación cortos que exploran la interactividad en Scratch y discuten las diversas soluciones y posibles aplicaciones de estos desafíos en sus programas.

Nivel: 3

Temas: Aplicar las técnicas para resolver problemas algorítmicos para resolver problemas computacionales.

Conocimiento Previo: Los estudiantes deben tener una cierta familiaridad con el entorno de Scratch, conectando los bloques para crear programas, creando sprites, el manejo del flujo del programa y eventos desencadenantes.

Notas de Planificación:

Antes de la actividad, los maestros deben seleccionar desafíos para seguir adelante con sus alumnos basándose en las características y conceptos de computación del Scratch, que les gustaría explorar. Por ejemplo, el maestro puede considerar los siguientes retos:

- Siempre que dos sprites colisionan, uno de ellos dice: "Disculpe".
 - Características del Scratch: Si, Los bloques están en contacto
 - Conceptos Computacionales: Condición, Eventos
- Cada vez que se hace clic en una sprite, el resto de sprites hacen un baile.
 - Características de Scratch: transmisiones, cuando se hace click en el bloque sprite
 - Conceptos Computacionales: Eventos, Paralelismo, Sincronización
- Cuando el marcador llega a 10, la escena (fondo) cambia.
 - Características del Scratch: Si, Operadores, Bloques de variables
 - Conceptos Computacionales: Condición, Operadores, Variables

Encuentre más desafíos en: <http://scratched.media.mit.edu/resources/short-scratch-programming-challenges>

Estrategias de Enseñanza/Aprendizaje:

Para cada desafío, los maestros deben:

- Presentar a los estudiantes con un desafío. Animarles a trabajar con sus compañeros para llegar a una posible solución.
- Después de 15 minutos, facilitar un debate en torno a la diversidad de soluciones que los estudiantes desarrollaron para abordar el desafío dado. Pregunte a los alumnos cómo pueden utilizar sus soluciones para crear otros proyectos de Scratch. Por ejemplo, el reto "Siempre que dos sprites colisionan, uno de ellos dice:" Disculpe "se pueden adaptar a un proyecto con una historia que involucre la interacción de múltiples sprites.

Evaluación y Valoración:

En la discusión en grupo, los alumnos pueden expresar sus soluciones y evaluar las soluciones de otras personas para hacer frente a cada desafío.

Necesidades:

Los maestros pueden imprimir los desafíos que los estudiantes tengan a la mano.

Recursos:

- Descripción y video de Scratch, Retos Actividades con un grupo de educadores en <http://scratched.media.mit.edu/resources/short-scratch-programming-challenges>
- ScratchEd website at <http://scratched.media.mit.edu/>

A.4 Ejemplo de Actividades para el Nivel 3C: Temas en Ciencias de la Computación

ACTIVIDAD: INTRODUCCION A DISEÑO ORIENTADO A OBJETOS

Duración: 3 horas

Descripción: Los estudiantes son introducidos a las etapas iniciales de la aplicación del diseño orientado a objetos (OOD), a un problema de programación y practicar la aplicación de esas medidas a un problema que luego puede ser utilizado como un proyecto de programación significativa

Nivel: 3C.1 AP Ciencias de la Computación, 3C.2 Proyectos

Temas: Ideas fundamentales sobre el proceso de diseño orientado a objetos del programa

Conocimiento y Habilidades Previas: ninguno

Notas de Planificación:

- Preparar diagramas en blanco para usar como tarjetas CRC y / o diagramas de objetos
- Seleccionar ejemplos de problemas que pueden ser resueltos mediante el uso de técnicas orientadas a objetos.

Estrategias de Enseñanza/Aprendizaje:

- Explicar las diferencias entre un objeto y un enfoque funcional orientado al el diseño de un programa de computación.
- Pida a un alumno explicar cómo se juega un juego de cartas (como el blackjack).
- Trabajar con la clase, identificar potenciales objetos involucrados en el juego.
- Trabajar con la clase, identificar las posibles operaciones que puedan ser realizadas por o para una tarjeta o un objeto posible. Los estudiantes hacen lluvia de ideas en pequeños grupos para identificar otros objetos como parte del problema.
- Ilustrar cómo las dos anotaciones se pueden utilizar para resumir el análisis.
- Discuta las posibles relaciones entre los objetos.
- Mostrar anotaciones para las relaciones.
- Describa un posible escenario en el juego y utilice las anotaciones desarrollados para representar ese escenario.
- Pida a los estudiantes en parejas que describen un segundo escenario y lo representen.
- Compartir los escenarios de la clase y considere si colectivamente representan la gama de posibilidades de escenarios especialmente extremos.
- Los estudiantes leen los posibles problemas y seleccionan uno sobre el cual hacer un OOD. Los estudiantes trabajan individualmente en las dos primeras fases (identificación de objetos y operaciones).
- Los estudiantes que trabajan en los mismos problemas comparten resultados y acuerdan en "mejor" conjunto de objetos y operaciones.
- Los estudiantes completan individualmente los dos últimos pasos.

Evaluación y Valoración:

- Evaluación formativa de los trabajos asignados en clase en forma de conferencias itinerantes.
- Evaluación sumaria aplicando OOD a un nuevo problema

Necesidades:

- Proporcionar copias de descripciones de los problemas, con los sustantivos y los verbos importantes indicados utilizando una fuente diferente o tamaño de letra. Proporcionar escenarios escritos para cada problema.

Recursos:

- Wirfs-Brock, Wilkerson, and Wiener, Designing Object-Oriented Software, Prentice Hall, 1990
- Fowler, UML Distilled: Applying the Standard Object Modeling Language, Addison-Wesley, 1997.

- Descripción general de los ejercicios ejemplo de la lección y están disponibles en: <http://max.cs.kzoo.edu/AP/OOD/OODPresentation/> and <http://max.cs.kzoo.edu/AP/OOD/OODSpecifications>

A.5 Recursos Adicionales para los Niveles 3.C.2 y 3.C.3:

Una amplia gama de recursos está disponible para apoyar el plan de estudios de Ciencias de la Computación en el nivel 3.

AP Curso de Ciencias de la Computación

El plan de estudios para todos los cursos AP CS se rige por el esquema de temas disponibles de la Junta Universitaria en <http://apcentral.collegeboard.com/apc/Controller.jpf>.

Cursos que llevan a la certificación de la Industria

Muchos de los cursos de certificación ofrecen un plan de estudios preparado que detalla el contenido y el orden de los temas. Si bien la aplicación de este tipo de cursos se puede simplificar por la información proporcionada, se necesita una evaluación cuidadosa en cuanto a contenidos propios en comparación a los conceptos generales. La información sobre el contenido de algunos cursos de certificación se puede encontrar en línea mediante la búsqueda de la certificación específica.

Cursos Basados en Proyectos

Los cursos basados en Proyectos pueden proporcionar educación específica orientada a los intereses de estudiantes específicos en Ciencias de la Computación. Mientras que estos cursos pueden ser ofrecidos por el distrito escolar local, se debería de considerar inscribir a los estudiantes en un curso universitario. Los Cursos basados en proyectos de computación a menudo se ofrecen en un colegio o universidad del departamento de ciencias de la computación, tecnología de la información o sistemas de información. Los programas computarizados de estudio a nivel universitario son bien establecidos y proporcionan una variedad de temas de actualidad. Además, un estudiante que participa en un curso de computación a nivel universitario puede ser elegible para obtener créditos universitarios. La escuela

secundaria tendrá que investigar el contenido de cada curso para determinar si es apropiado para estudiar en el Nivel 3C.

Los cursos para créditos universitarios se pueden ofrecer en el campus de la universidad, en la escuela secundaria o por medio de educación a distancia. Un curso que se ofrece en el campus de la universidad pondrá menos carga en la escuela secundaria para facilitar hardware especial, software y recursos de la facultad.

Los cursos en línea se pueden tomar fuera de horas o durante el tiempo de la escuela. Un estudiante que decide tomar un curso en línea fuera del horario escolar tendrá acceso a hardware específico, software e Internet. Cuando se ofrece un curso en línea por una universidad durante el día escolar, puede ser posible programar varios cursos en línea durante la misma sesión de clase, permitiendo una mejor utilización del tiempo de la facultad. Para apoyar esto, la escuela también debe proporcionar acceso al hardware necesario, el software e Internet.

Un curso para obtener créditos universitarios ofrecidos en la escuela secundaria permite que el estudiante y los miembros de la facultad de la escuela interactúen de manera tradicional. La escuela secundaria tendrá que asegurarse de que el plan de estudios está sancionado por la universidad y los estudiantes están matriculados oficialmente en la universidad. La universidad podría participar en la selección del miembro de la facultad que impartirá el curso.

Algunos métodos típicos por los que los estudiantes de secundaria pueden lograr créditos universitarios son los siguientes.

Tech Prep y crédito de Articulación- El estudiante toma un curso en la escuela secundaria y recibe el crédito de secundaria. El curso está también pre-aprobado para

crédito universitario en una universidad en particular y se enseña típicamente por un miembro de la facultad de la escuela secundaria. El crédito por el curso se otorga al matricularse en esa universidad. La universidad puede requerir que el estudiante apruebe un examen antes de aplicar el crédito para el expediente académico del alumno.

Crédito Doble / inscripción simultánea-El estudiante está matriculado en un curso para el que él / ella recibe simultáneamente crédito para la escuela secundaria y universitario. El estudiante debe cumplir con todos los requisitos de la universidad para el ingreso en el curso.

Exámenes reto-El estudiante puede ser capaz de demostrar la competencia y recibir crédito por examen. Este método es útil para un estudiante que completa un curso de la escuela secundaria que no es articulado o que cree que ha ganado independientemente el conocimiento de todos los temas tratados en un curso específico. El estudiante puede tener que pagar una cuota para tomar el examen.

Prueba-A Cursos Avanzados / CLEP – un estudiante planeando hacia una carrera universitaria en Ciencias de la Computación, Tecnología de la Información, Ingeniería o tal vez desee tomar el examen AP en Ciencias de la computación. Un estudiante planeando ingresar a la universidad en una carrera de negocios tal vez desee tomar el examen CLEP en Sistemas de Información y Aplicaciones computacionales. Se recomienda que el estudiante determine cómo se otorgará el crédito cuando apruebe el examen de su institución de destino.

Programas que ofrecen crédito universitario para estudiantes de secundaria

Varios programas ofrecen créditos universitarios para los estudiantes de secundaria. Aquí hay tres ejemplos:

La Universidad de Pittsburgh's College in High School es un ejemplo de las escuelas secundarias que interactúan con una universidad. El programa College High School (CHS) ha ofrecido a los estudiantes de secundaria calificados la oportunidad de obtener créditos de la Universidad de Pittsburgh durante su día regular de clases. Los estudiantes están obligados a pagar una cuota reducida para participar en el programa. Puede haber ayuda financiera disponible. El programa ofrece "14

cursos a más de 2.900 estudiantes de escuelas secundarias de más de 100 escuelas secundarias con más de 200 maestros" (Pittsburg). Al final del curso, los estudiantes reciben expediente universitario de la Universidad de Pittsburgh. Encuentre información sobre este programa en <http://www.asundergrad.pitt.edu/offices/chsp/highschools.html>.

La Universidad de Cincinnati Clermont College también ofrece a los estudiantes de secundaria la oportunidad de obtener créditos universitarios al proporcionar un Programa de Opciones de Inscripción Post-Secundaria (PSEOP). Este programa permite al estudiante matricularse en la universidad para obtener créditos universitarios solamente o para inscribirse en la escuela secundaria y obtener crédito universitario. En algunos semestres, el costo de la matrícula y los libros puede ser pagado por la escuela secundaria y / o universidad. Ver http://admissions.uc.edu/highschool/adv_placement/pseo.html para mayor información.

La Universidad del Norte de Colorado proporciona "crédito concurrente de la escuela secundaria, un programa para Residentes de Colorado que permiten a los estudiantes de onceavo y doceavo grado obtener créditos universitarios mientras están en la escuela" (Colorado). Se proporcionan tres opciones en el marco de este programa:

Opción I—Programa Fast Track: Para el estudiante que está en Doceavo grado de la Escuela Secundaria y ha cumplido con los requisitos de graduación.

Opción II—Programa de Opciones de Inscripción Post-Secundaria: Para el estudiante que está en onceavo o doceavo grado en una escuela secundaria o superior y no ha cumplido con los requisitos de la escuela secundaria.

Opción III—Programa de Aceleración de la Universidad: Para el estudiante que está en onceavo o doceavo grado de secundaria y quiere acelerar su programa de la universidad haya o no cumplido con los requisitos de graduación. (Colorado)

Recursos

1. eHow.com, How to Get College Credit Before Attending College, Downloaded 1/29/2011 http://www.ehow.com/how_4424237_getcollege-credit-before-attending.html

3. Arthur R. Greenberg, ERIC Clearinghouse on Higher Education Washington DC, BBB27915, George Washington Univ. Washington DC. School of Education and Human Development, ERIC Identifier: ED347956, Publication Date: 1992-03-00, <http://www.ericdigests.org/1992-2/high.htm>

4. Comparison of Methods to Receive College Credit for Courses Taken in High School, <http://www.tea.state.tx.us/Cate/teched/collegedcreditinhs.pdf>

5. Dawn Fuller, University of Cincinnati, High School Students: Do You Qualify for UC College Credit? Downloaded: March 7, 2003, from <http://www.uc.edu/news/NR.asp?id=300>

6. The College Board, AP Central, Downloaded 1/29/2011, <http://apcentral.collegeboard.com/apc/Controller.jp>

7. University of Cincinnati Clermont College, Post-Secondary Enrollment Options Program. Downloaded 1/29/2011 from http://admissions.uc.edu/highschool/adv_placement/pseop.html

8. University of Northern Colorado, Undergraduate Admissions, High School Concurrent. Downloaded 1/29/2011 from http://www.unco.edu/admissions/special_populations/highschoolconcurrent.asp

9. University of Pittsburgh, College in High School, Downloaded 1/29/2011 <http://www.asundergrad.pitt.edu/offices/chsp/highschools.html>

10. U.S. Department of Education, Community College and High School Partnerships by Elisabeth Barnett and Katherine Hughes, Community College Research Center. Downloaded 1/29/2011 from <http://www2.ed.gov/PDF Docs/college-completion/09-community-college-and-high-school-partnerships.pdf>

11. U.S. Department of Education, Tech Prep Education. Downloaded 1/29/2011 from <http://www2.ed.gov/programs/techprep/index.html>

8. Lapidot, T. and Hazzan, O. (2003). Methods of Teaching a Computer Science Course for Prospective Teachers. *Inroads—The SIGCSE Bulletin*, 35(4), 29–34.
9. National Council for Accreditation of Teacher Education. Program for Initial Preparation of Teachers of: Educational Computing and Technological Literacy, and Secondary Computer Science Education. <http://www.ncate.org/Standards/ProgramStandardsandReportForms/tabid/676/Default.aspx>
10. National Research Council Committee on Information Technology Literacy, Being Fluent with Information Technology, National Academy Press, Washington, DC, May 1999.

11. Papert, Seymour. *Mindstorms: Children, Computers, and Powerful Ideas* (1980). New York, NY: Basic Books.

12. Simard, C., Stephenson, C., Kosaraju, D. (2010). Addressing Core Equity Issues in K–12 Computer Science Education: Identifying Barrier and Sharing Strategies. The Anita Borg Institute and the Computer Science Teachers Association. Palo Alto, CA.

13. Task Force of the Pre-College Committee of the Education Board of the ACM. (1993). ACM Model High School Computer Science Curriculum. *Communications of the ACM*, 36(5), 87–90.

14. Tucker, A., McCowan, D., Deek, F., Stephenson, C., Jones, J., & Verno, A. (2003, 2006). A Model Curriculum for K–12 Computer Science: Report of the ACM K–12 Task Force Computer Science Curriculum Committee, New York, NY: Association for Computing Machinery.

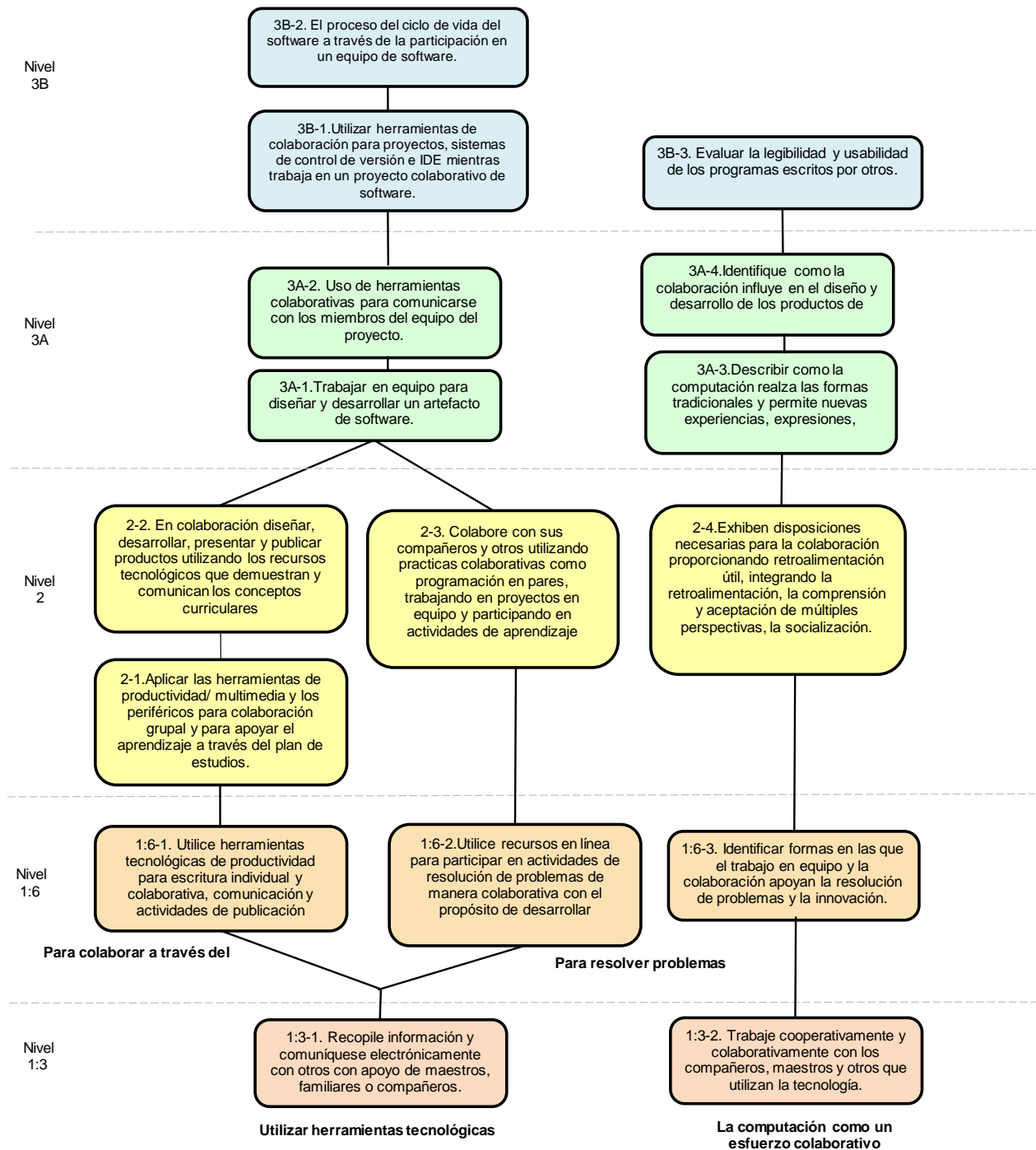
15. Wilson, C., Stephenson, C., Stehlik, M., Sudol, L. (2010). *Running on Empty: The Failure to Teach Computer Science in the Digital Age*. Association for Computing Machinery and the Computer Science Teachers Association, New York, NY.

Referencias

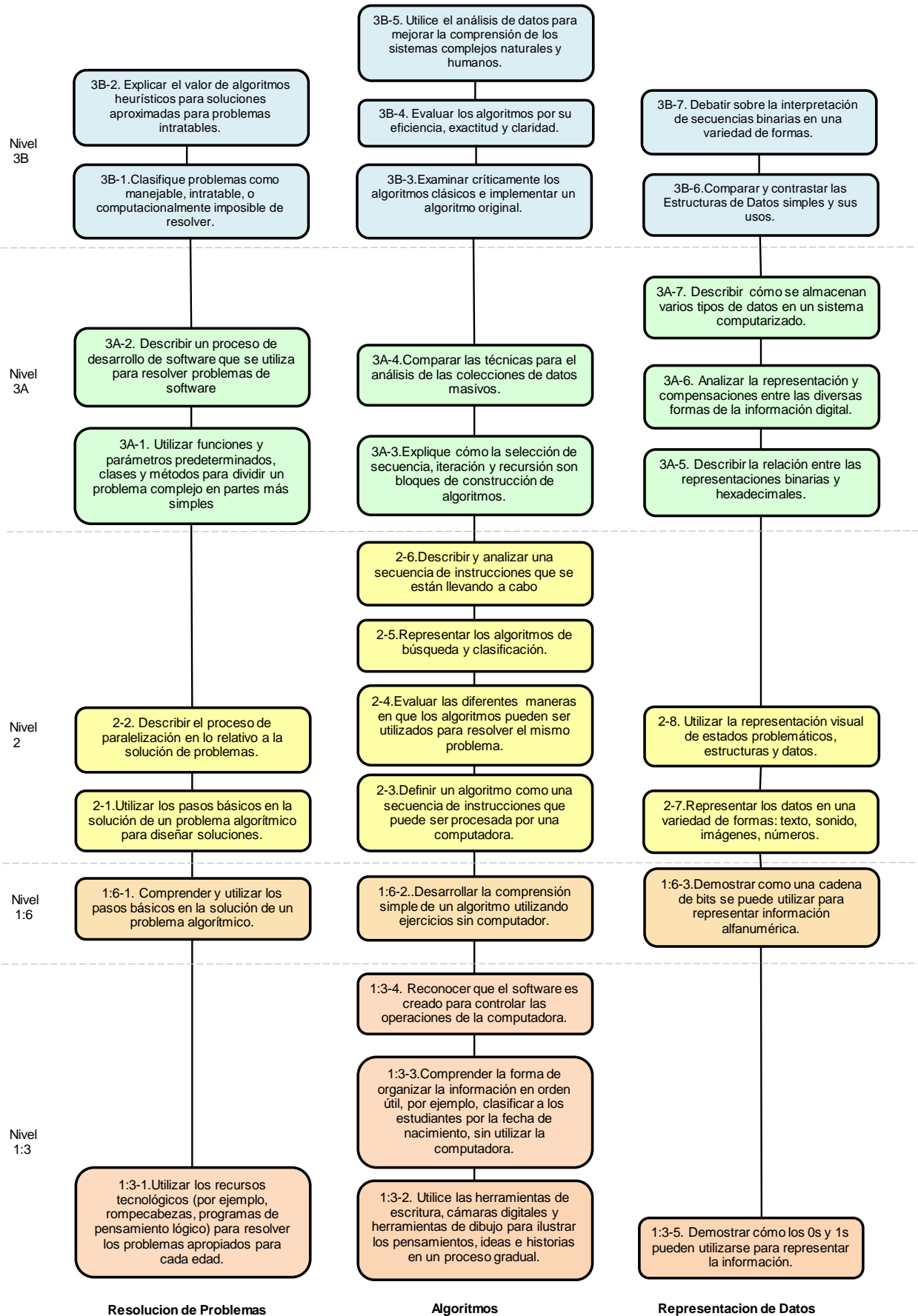
1. 2008–09 Taulbee Survey. (2009). Downloaded February 2011 from <http://www.cra.org/resources/taulbee/>
2. AP Course Description: Computer Science. (2011) Downloaded February 2011 from http://apcentral.collegeboard.com/apc/public/courses/teachers_corner/4483.html
3. Computer Science Unplugged. (2011). Downloaded February 2011 from <http://csunplugged.org/>
4. Deek, F. and H. Kimmel. (1999). Status of Computer Science Education in Secondary Schools. *Computer Science Education* 9(2), 89–113, August 1999.
5. Ericson, B., Armoni, M., Gal-Ezer, J., Seehorn, D., Stephenson, C., Trees, F. (2008) Ensuring Exemplary Teaching in an Essential Discipline: Addressing the Crisis in Computer Science Teacher Certification. New York: Association for Computing Machinery.
6. Friedman, T.L. (2007). *The World is Flat 3.0: A Brief History of the Twenty-first Century*. Picador/Farrar Straus, and Giroux. New York, NY.
7. Gal-Ezer, J. and D. Harel. (1999). Curriculum for a high school computer science curriculum. *Computer Science Education* 9(2), 114-147.

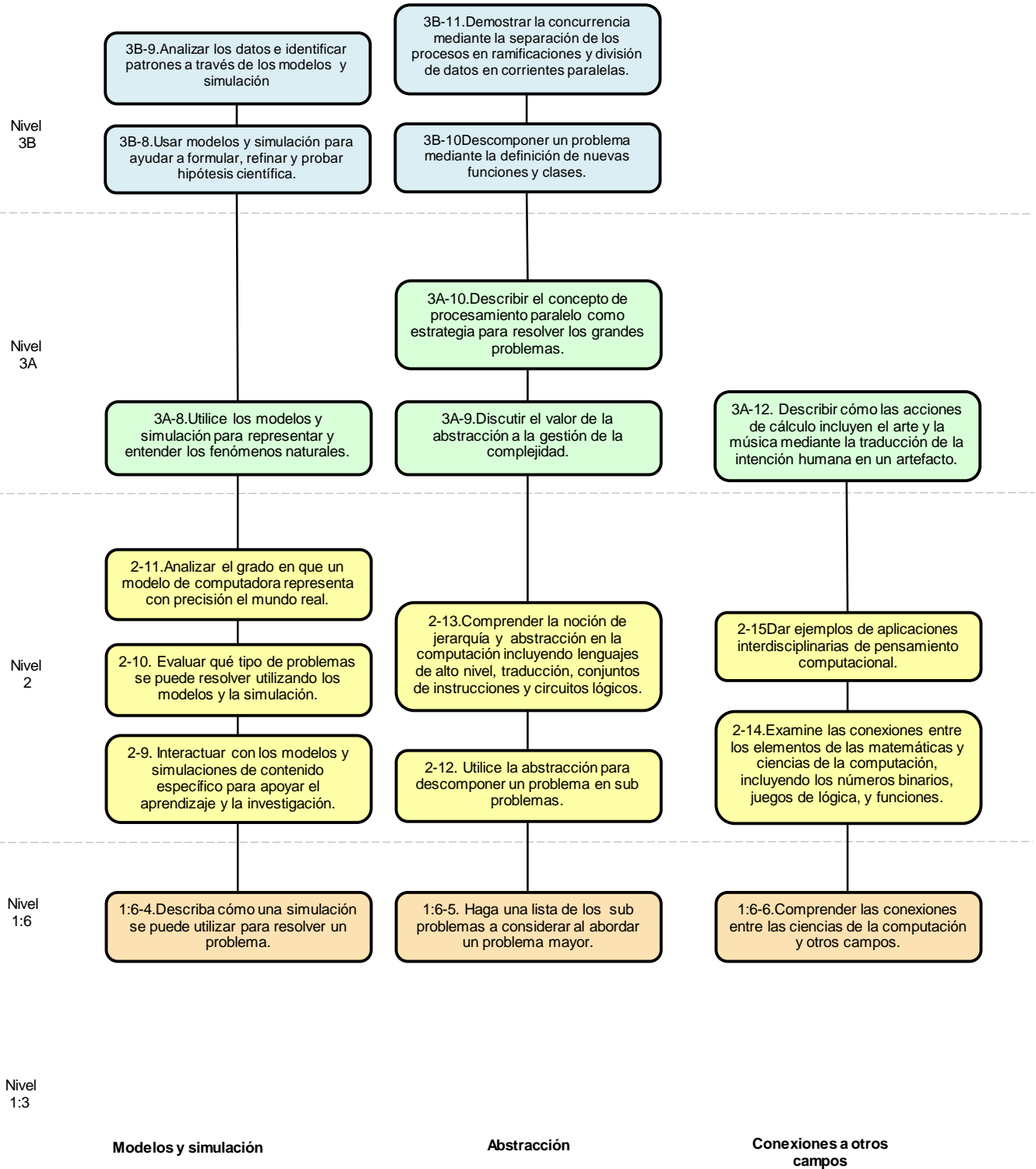
K-12 Gráficos Escalonados de los Estándares

Colaboración

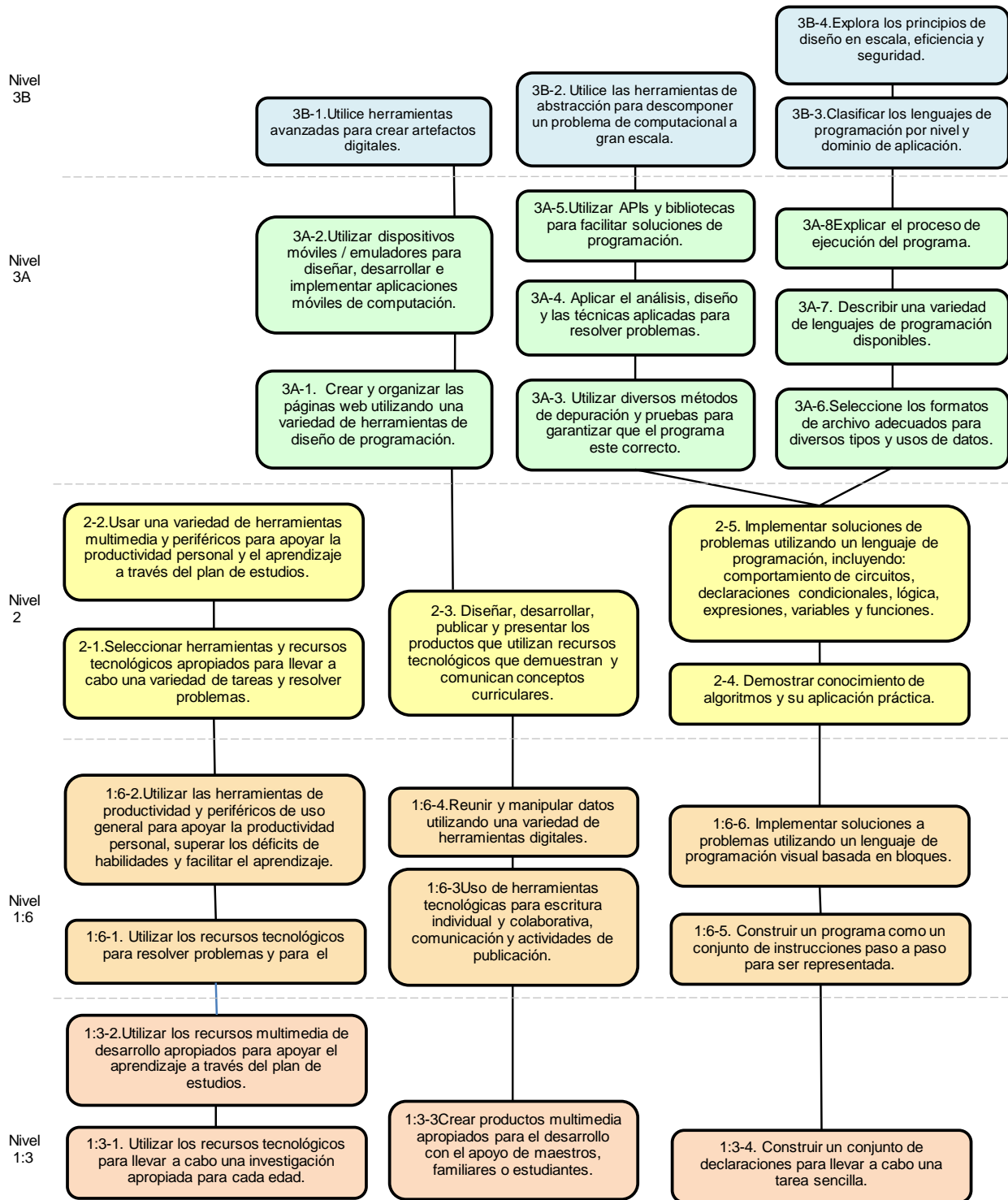


Pensamiento Computacional





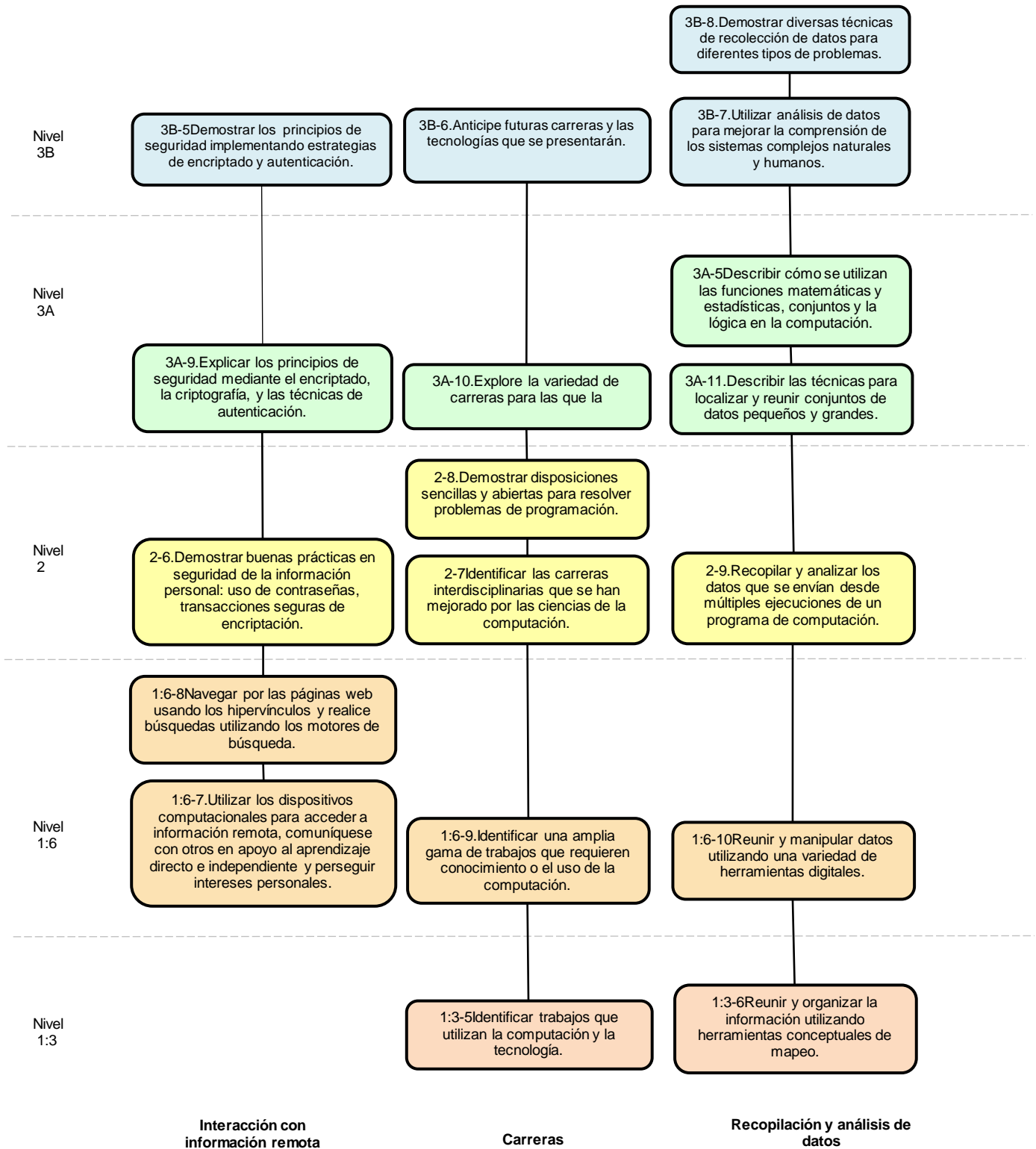
Practica de Computación y Programación



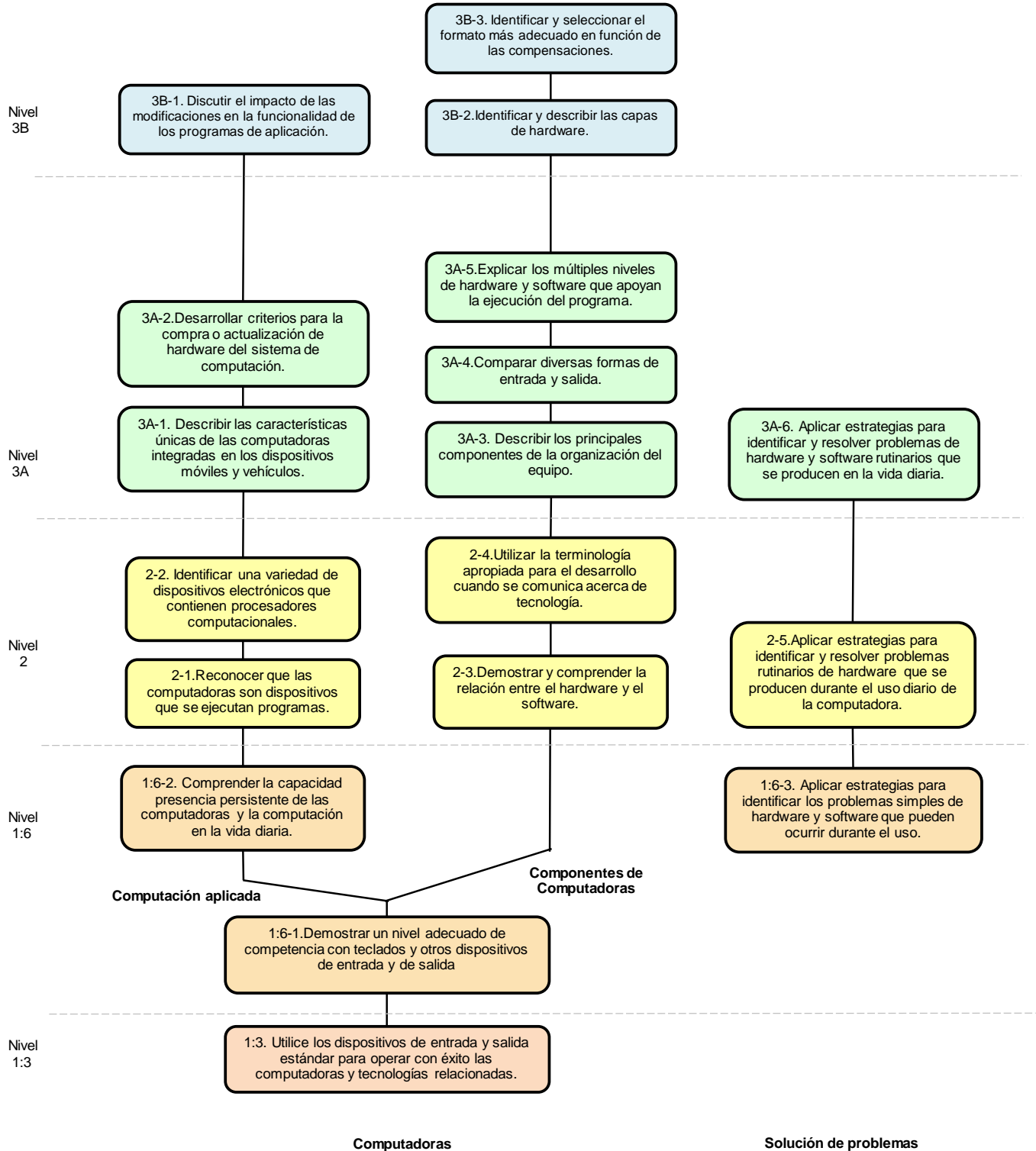
Utilización de los recursos tecnológicos para el aprendizaje

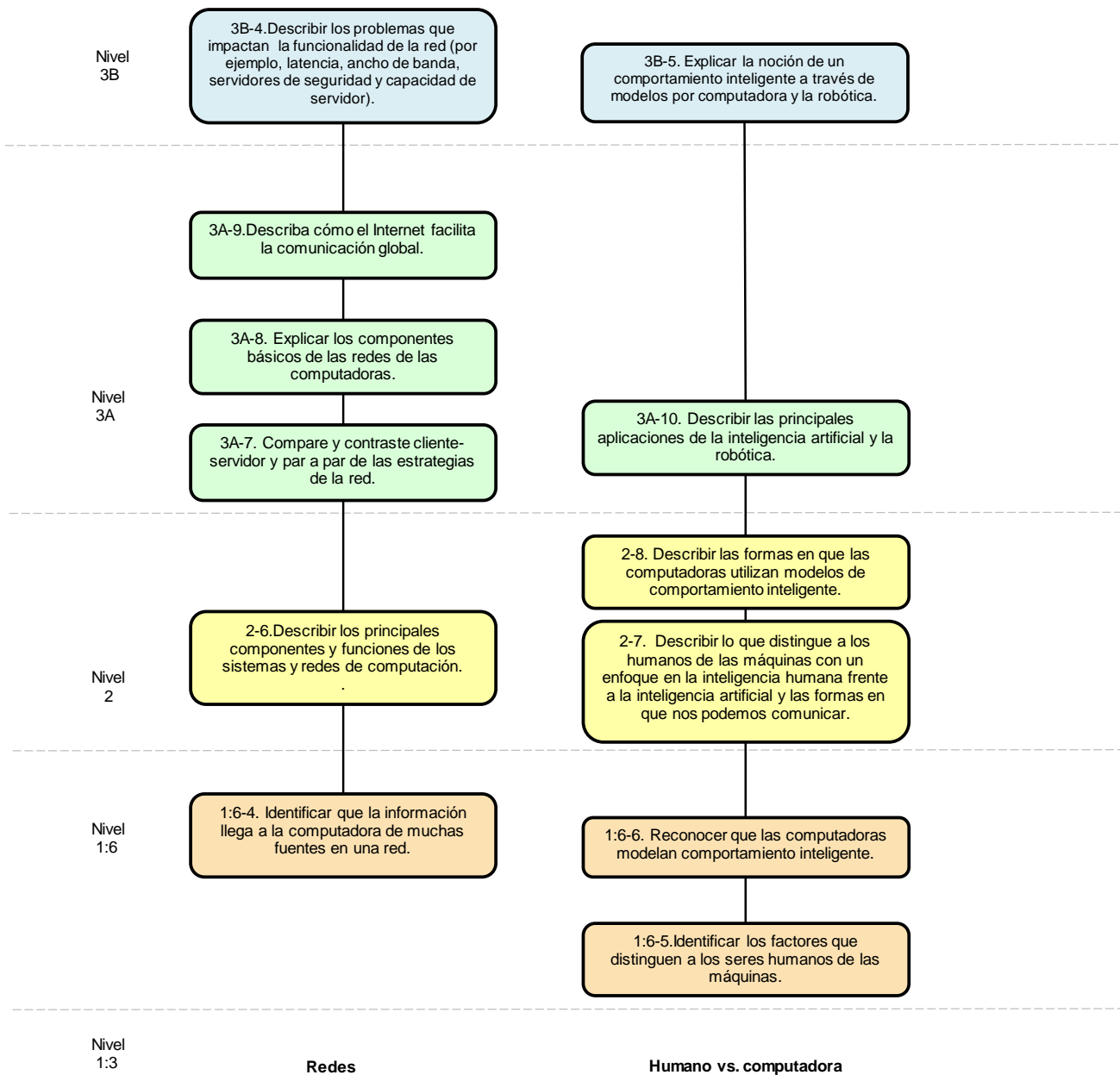
Utilización de herramientas tecnológicas para crear artefactos digitales

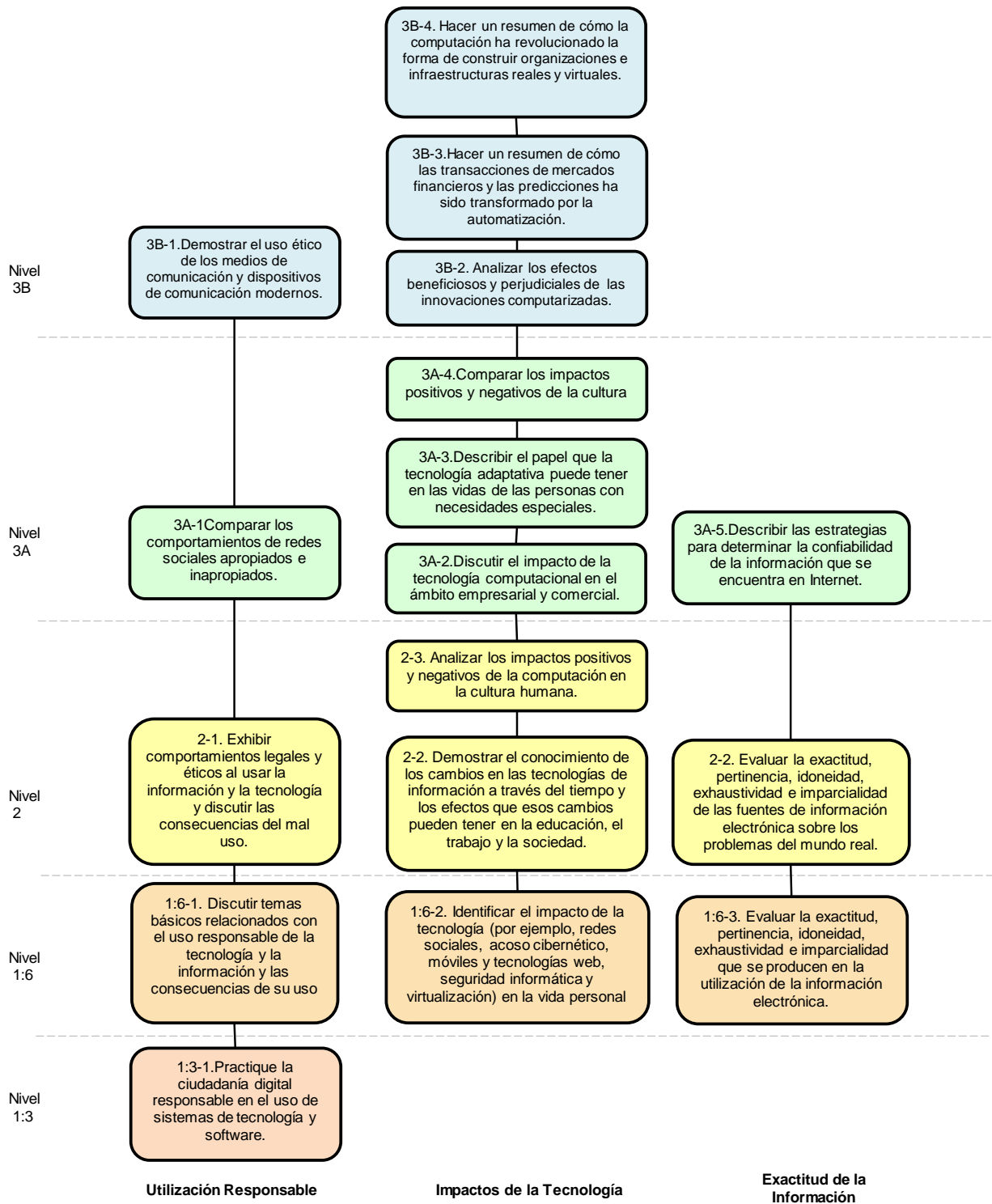
Programación

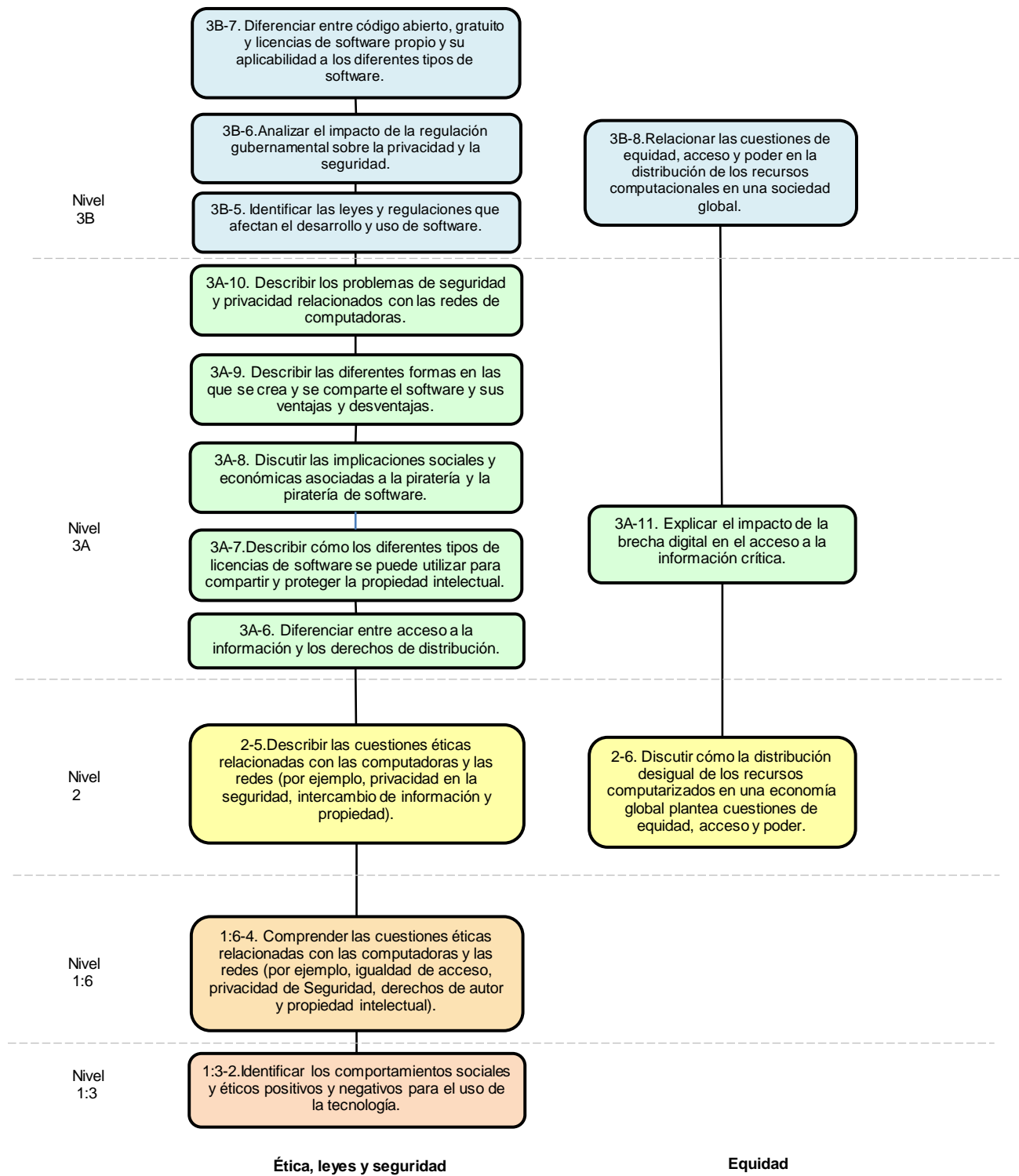


Computadoras y Dispositivos de Comunicación









¡El primer año de su membresía CSTA es GRATIS!

¿QUÉ ES LA ASOCIACION DE MAESTROS DE LAS CIENCIAS DE LA COMPUTACION (CSTA)?

La Asociación de Maestros de Ciencias de la Computación, es una sociedad de responsabilidad limitada bajo los auspicios de ACM, ha sido organizada para servir como centro de coordinación para hacer frente a varios problemas (nivel de crisis graves) en la enseñanza de computación en los niveles de K-12, incluyendo:

- La falta de apoyo administrativo, curricular, financiero, de desarrollo profesional y de apoyo al liderazgo de los maestros
- Falta de un plan de estudios estandarizado
- Falta de comprensión de la disciplina y su lugar en el plan de estudios
- Falta de oportunidades para que los maestros desarrollen sus habilidades e intereses

Estas cuestiones dan lugar a:

- Una profunda sensación de aislamiento, y
- Baja en la inscripción en los programas de Ciencias de la Computación a nivel universitario

Hay otras organizaciones que se ocupan de la utilización de la tecnología en el plan de estudios, pero sólo CSTA habla directamente y con pasión por las ciencias de la computación de la escuela secundaria.

NUESTRA MISIÓN

CSTA es una organización que apoya y promueve la enseñanza de las ciencias de la computación y otras disciplinas de computación en el nivel de K-12, proporcionando oportunidades para los maestros y estudiantes para entender mejor las disciplinas computacionales y a prepararse más exitosamente a sí mismos para enseñar y aprender.

NUESTRAS METAS

Las metas organizacionales y educativas de CSTA son:

- Ayudar a construir una fuerte comunidad de educadores de CC que comparten sus conocimientos
- Proporcionar a los maestros oportunidades para desarrollo profesional de alta calidad
- Abogar en todos los niveles para un plan de estudios integral de las ciencias de la computación
- Apoyar proyectos que comunican la emoción de CC a los estudiantes y mejoran su comprensión de las oportunidades que ofrece
- Recopilación y difusión de la investigación sobre la enseñanza de computación
- Formulación de recomendaciones de políticas para apoyar las CC en el plan de estudios de la escuela secundaria, y
- Crear conciencia de que los educadores de las ciencias de la computación son profesionales altamente calificados, con habilidades que enriquecen la experiencia educativa de sus estudiantes

NUESTRO ALCANCE

El alcance de la organización incluye:

- La escuela secundaria (todos los aspectos de la enseñanza de computación)
- La escuela primaria y secundaria (introducción a la solución de problemas y pensamiento algorítmico)
- Colegio/universidad (para establecer una mejor transición de los programas de la escuela secundaria y proporcionar un mayor nivel de apoyo a los maestros de la escuela secundaria)
- Comercio e industria (apoyando la enseñanza de ciencias de la computación y a los maestros)

¿QUIÉN DEBE REGISTRARSE?

- Todos los maestros de ciencias de la computación de Escuelas Secundarias y Escuelas Medias
 - Todos los Maestros de Aplicaciones computacionales de los niveles de K-12
 - Todas las personas interesadas (y apasionados) acerca de la Educación de los niveles de K-12
- ¡El primer año de su membresía CSTA es GRATIS!

ÚNETE VIA...

web: <http://csta.acm.org/>

Teléfono: 1.800.342.6626

Correo electrónico: cstahelp@acm.org